

**Національний університет «Чернігівський колегіум»
імені Т.Г. Шевченка**

**Г.Ю. Цибко
Ю.В. Горошко
А.О. Костюченко**

**Програмування у Python.
Практичний курс**

Чернігів, 2022

УДК 004.438 Python

ORCID <https://orcid.org/0000-0002-1861-3003>

ORCID <https://orcid.org/0000-0001-9290-7563>

ORCID <https://orcid.org/0000-0002-6178-6444>

Ц56 Цибко Г.Ю., Горошко Ю.В., Костюченко А.О.

Програмування у Python. Практичний курс: навчальний посібник. Ч.: ФОП Баликіна С.М., 2022. 180 с.

Подано рекомендації до проведення практичних занять і лабораторних робіт з основ програмування мовою Python. Тематика посібника включає основи алгоритмізації, структури даних, основи об'єктно-орієнтованого програмування. Посібник складено з урахуванням досвіду викладання програмування студентам освітніх програм „Комп'ютерні науки” та „Середня освіта (інформатика)”. Може бути використаний науково-педагогічними працівниками ЗВО та ЗСО, здобувачами середньої та вищої освіти, а також у процесі самостійного вивчення основ програмування.

Рецензенти:

Ткач Юлія Миколаївна - доктор педагогічних наук, професор, завідувач кафедри кібербезпеки та математичного моделювання Національного університету «Чернігівська політехніка»

Горчинський Сергій Володимирович - кандидат педагогічних наук, доцент кафедри технологічної освіти та інформатики Національного університету «Чернігівський колегіум» імені Т.Г.Шевченка

Рекомендовано до друку вченою радою природничо-математичного факультету Національного університету «Чернігівський колегіум» імені Т.Г.Шевченка, протокол № 6 від 31 січня 2022 р.

© Цибко Г.Ю., 2022

© Горошко Ю.В., 2022

© Костюченко А.О., 2022

ЗМІСТ

ВСТУП.....	4
1. РЕАЛІЗАЦІЯ ОСНОВНИХ АЛГОРИТМІЧНИХ СТРУКТУР У PYTHON ..6	
1.1. Основні поняття алгоритмізації та програмування. Базові структури алгоритмів. Мова блок-схем.....	6
1.2. Оператори мови. Оператор присвоювання. Ведення–виведення даних. Класифікація типів даних. Стандартні типи даних та операції над ними. Лінійні програми. Опрацювання виключних ситуацій.	7
1.3. Лабораторна робота №1. Лінійні програми.....	14
1.4. Організація розгалужень. Умовний оператор	21
1.5. Лабораторна робота №2. Розгалуження	26
1.6. Організація циклів. Оператори циклу з передумовою. Цикл з параметром. Вкладені цикли.	31
1.7. Лабораторна робота №3. Цикли	39
2. ОСНОВНІ СТРУКТУРИ ДАНИХ У PYTHON.	
ФУНКЦІЇ КОРИСТУВАЧА.....	47
2.1. Структури даних. Списки	47
2.2. Лабораторна робота №4. Списки	70
2.3. Функції користувача.....	77
2.4. Лабораторна робота №5. Функції.....	88
2. 5. Рядкові величини	94
2.6. Лабораторна робота №6. Рядкові величини	102
2.7. Множини та словники	106
2.8. Лабораторна робота №7. Множини та словники	117
2.9. Файли	124
2.10. Лабораторна робота №8. Текстові файли.....	133
3. ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ У PYTHON	140
3.1. Об'єктний тип даних. Побудова ієрархічної залежності між об'єктами.....	140
3.2. Лабораторна робота № 9. Наслідування	147
3.3. Основи об'єктно-орієнтованого програмування. Поліморфізм	152
3.4. Лабораторна робота №10. Поліморфізм	159
3.5. Агрегація	163
3.6. Лабораторна робота № 11. Агрегація.....	167
3.7. Основи об'єктно-орієнтованого програмування. Захист даних	172
3.8. Лабораторна робота № 12. Захист даних	175
Список використаних джерел.....	178

ВСТУП

Алгоритмізація і програмування складають фундамент комп'ютерних наук, і відтак є надзвичайно важливими у навчанні інформатичних дисциплін у закладах середньої і фахової вищої освіти.

Наявність фундаментальних знань та навичок з основ програмування та володіння різними мовами програмування є одним з програмних результатів навчання за освітніми програмами підготовки вчителів інформатики закладів середньої освіти, а здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування визначено Державним стандартом вищої освіти за спеціальністю Комп'ютерні науки фаховою компетентністю.

Структура і зміст посібника складені на основі досвіду викладання курсу програмування студентам бакалаврських освітніх програм „Комп'ютерні науки” та „Середня освіта (інформатика)” Національного університету „Чернігівський колегіум” імені Т.Г.Шевченка.

Здобувачі вищої освіти, які навчаються за зазначеними освітніми програмами, опановують алгоритмізацію і програмування упродовж усього терміну навчання. При цьому вивчення програмування починається з мови Python. Вибір Python зумовлений її популярністю, широкою сферою застосування, багатопарадигменністю, простотою і перспективністю. Python вбачається багатьма науковцями цілком вдалим вибором першої мови при навчанні програмування.

Пропонований практичний курс мови Python відповідає лекційному курсу, викладеному в [1]. Окремі теоретичні питання для обговорення на практичних заняттях висвітлено в [4], [5]. Кожне практичне заняття супроводжено переліком питань для обговорення, низкою задач з можливими варіантами розв'язків, завданнями для самостійного виконання. Широкі можливості писати компактні програми з використанням бібліотек мови Python можуть приховати від програміста-початківця структуру певних часто вживаних алгоритмів. Тому у прикладах поєднуються методи розв'язування задач з використанням особливостей мови і з використанням стандартних підходів структурного програмування.

Посібник містить три розділи, присвячені відповідно реалізації у Python основних алгоритмічних структур, основних структури даних та основ об'єктно-орієнтованого програмування.

Практичний курс пропонується розпочати з розгляду основних понять алгоритмізації та програмування: етапи розв'язування практичних задач з допомогою комп'ютера; алгоритмічні мови, мови програмування, поняття про трансляцію і програми-транслятори. Коротка історія розвитку мов програмування, огляд основних парадигм програмування. Базові структури алгоритмів, мова блок-схем [4], [5].

Наступні теми стосуються безпосередньо мови Python: Оператори мови. Оператор присвоєння. Ведення–виведення даних. Класифікація типів даних. Стандартні типи даних та операції над ними. Лінійні програми. Опрацювання виключних ситуацій. Організація розгалужень. Умовний оператор. Організація циклів. Оператори циклу з передумовою. Цикл з параметром. Вкладені цикли. Структури даних. Списки. Функції. Рядкові величини. Множини та словники. Файли [1]. Основи об'єктно-орієнтованого програмування. Об'єктний тип даних. Побудова ієрархічної залежності між об'єктами. Реалізація поліморфізму (пізнього зв'язування). Агрегація. Інкапсуляція.

Для проведення лабораторних занять пропонується практикум з дванадцяти робіт, які охоплюють основний зміст курсу і містять по дванадцять варіантів завдань для самостійного виконання.

Наведені у посібнику задачі можна розглядати як на практичних заняттях, так і на лекціях або у самостійній роботі, зокрема в процесі дистанційного або змішаного навчання. У посібник не включені питання роботи в певному інтегрованому середовищі програмування, як такі, що значною мірою залежать від конкретних умов освітнього процесу.

Матеріал посібника може бути використаний науково-педагогічними працівниками та здобувачами закладів вищої та середньої освіти, а також у процесі самостійного вивчення основ програмування.

1. РЕАЛІЗАЦІЯ ОСНОВНИХ АЛГОРИТМІЧНИХ СТРУКТУР У PYTHON

1.1. Основні поняття алгоритмізації та програмування. Базові структури алгоритмів. Мова блок-схем

1. Поняття алгоритму.
2. Мова, алгоритмічна мова, мова програмування.
3. Системи програмування. Інтерпретація та компіляція.
4. Парадигми програмування.
5. Поняття моделі та моделювання. Інформаційні та математичні моделі.
6. Основні етапи розв'язування задачі за допомогою комп'ютера.
7. Базові структури алгоритмів.
8. Мова блок-схем.
9. Величини, типи величин. Вирази.
10. Логічні вирази, логічні операції. Таблиці істинності.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Горошко Ю.В., Костюченко А.О., Теорія і методика розробки педагогічних програмних засобів. – Ч.: Виготовлення Єрмоленко О.М., 2011. - 144 с.
3. Дудик М.В., Рамський Ю.С., Цибко Г.Ю. Основи програмування: Навчальний посібник для студентів вищих навчальних закладів фізико-математичних та індустріально-педагогічних спеціальностей. – Умань: АЛМІ, 2009. – 162 с.

Практичні завдання

Розв'язати задачу. Побудувати математичну модель, скласти алгоритм, зобразити блок-схему алгоритму.

1. Дано довжину ребра куба. Знайти площу поверхні та об'єм куба.
2. Визначити більше з двох заданих чисел (менше з двох заданих чисел).
3. Визначити більше з трьох заданих чисел (менше з трьох заданих чисел).
4. Розв'язати квадратне рівняння виду $ax^2+bx+c=0$, $a \neq 0$.

5. Дано три числа a , b , c . Якщо трикутник зі сторонами довжиною a , b , c існує, обчислити його площу за формулою Герона. В протилежному випадку вважати площу рівною нулю.

1.2. Оператори мови. Оператор присвоювання. Ведення–виведення даних. Класифікація типів даних. Стандартні типи даних та операції над ними. Лінійні програми. Опрацювання виключних ситуацій.

Питання для обговорення

1. Мова програмування Python: основні поняття.
2. Середовище програмування. Інтерактивний режим. Пакутий режим.
3. Python-програма.
4. Оператори мови. Оператор присвоювання.
5. Оператори мови. Введення–виведення даних.
6. Стандартні типи даних та операції над ними.
7. Стандартні функції. Модуль math.
8. Лінійні програми. Опрацювання виключних ситуацій.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

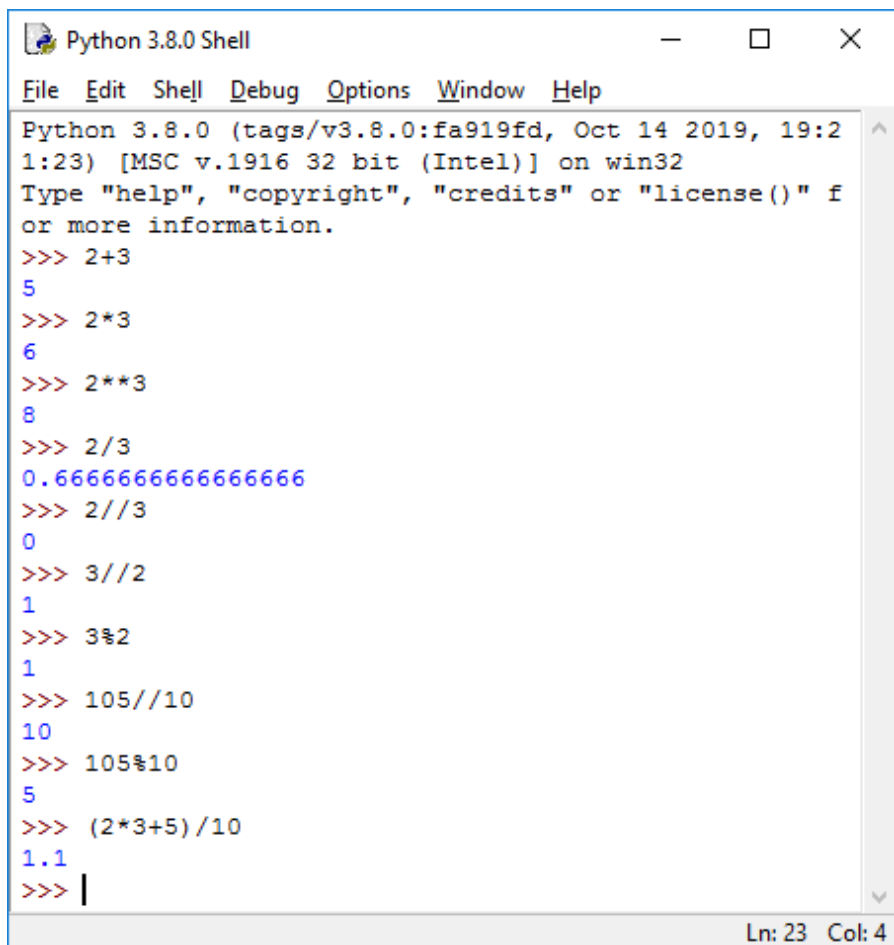
1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Робота в інтерактивному режимі

1. Запустити на виконання інтерпретатор Python.
2. Ознайомитись з виконанням простих арифметичних операцій в інтерактивному режимі.

Ввести вирази, наведені на Рис.1. Проаналізувати одержані результати.



The image shows a screenshot of a Python 3.8.0 Shell window. The window title is "Python 3.8.0 Shell" and it has standard Windows window controls (minimize, maximize, close). The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main content area displays the following text:

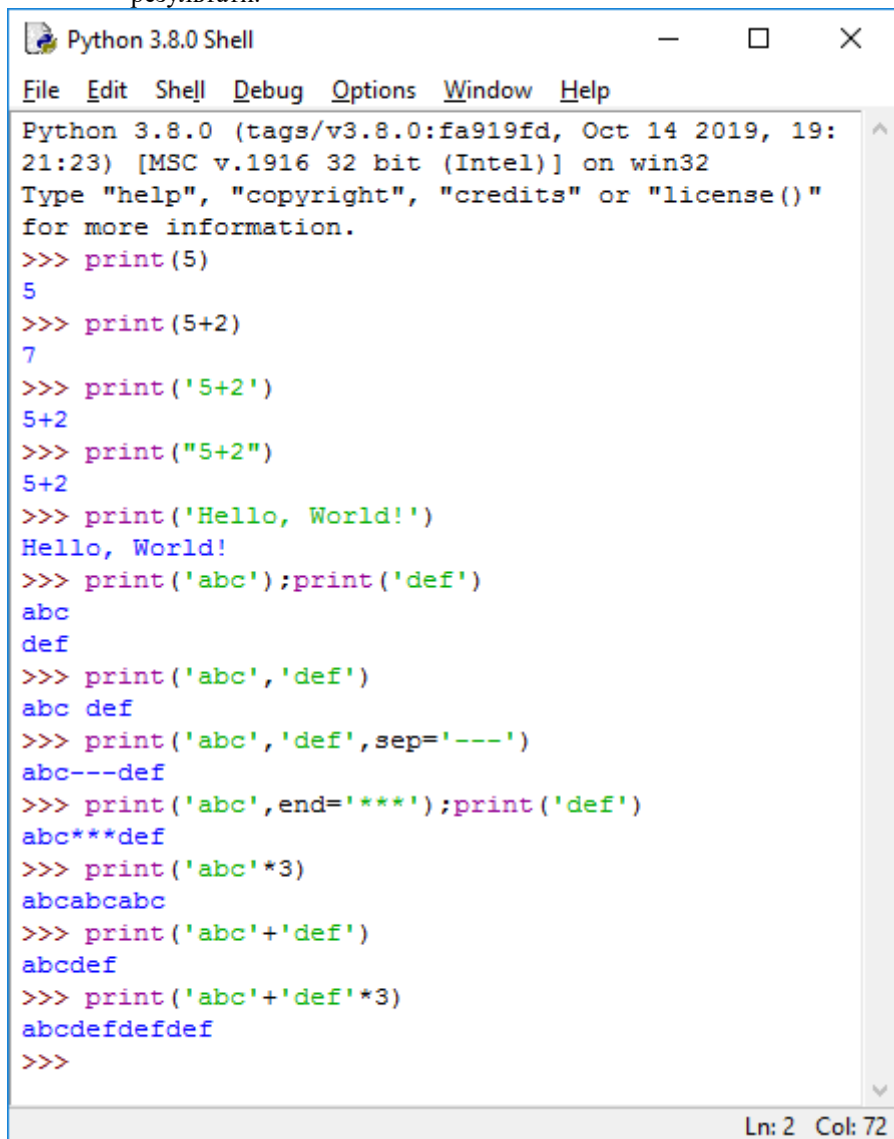
```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> 2*3
6
>>> 2**3
8
>>> 2/3
0.6666666666666666
>>> 2//3
0
>>> 3//2
1
>>> 3%2
1
>>> 105//10
10
>>> 105%10
5
>>> (2*3+5)/10
1.1
>>> |
```

At the bottom right of the window, the status bar shows "Ln: 23 Col: 4".

Рис.1

3. Ознайомитись з виконанням оператора виведення в інтерактивному режимі.

Ввести вирази, наведені на Рис.2. Проаналізувати одержані результати.



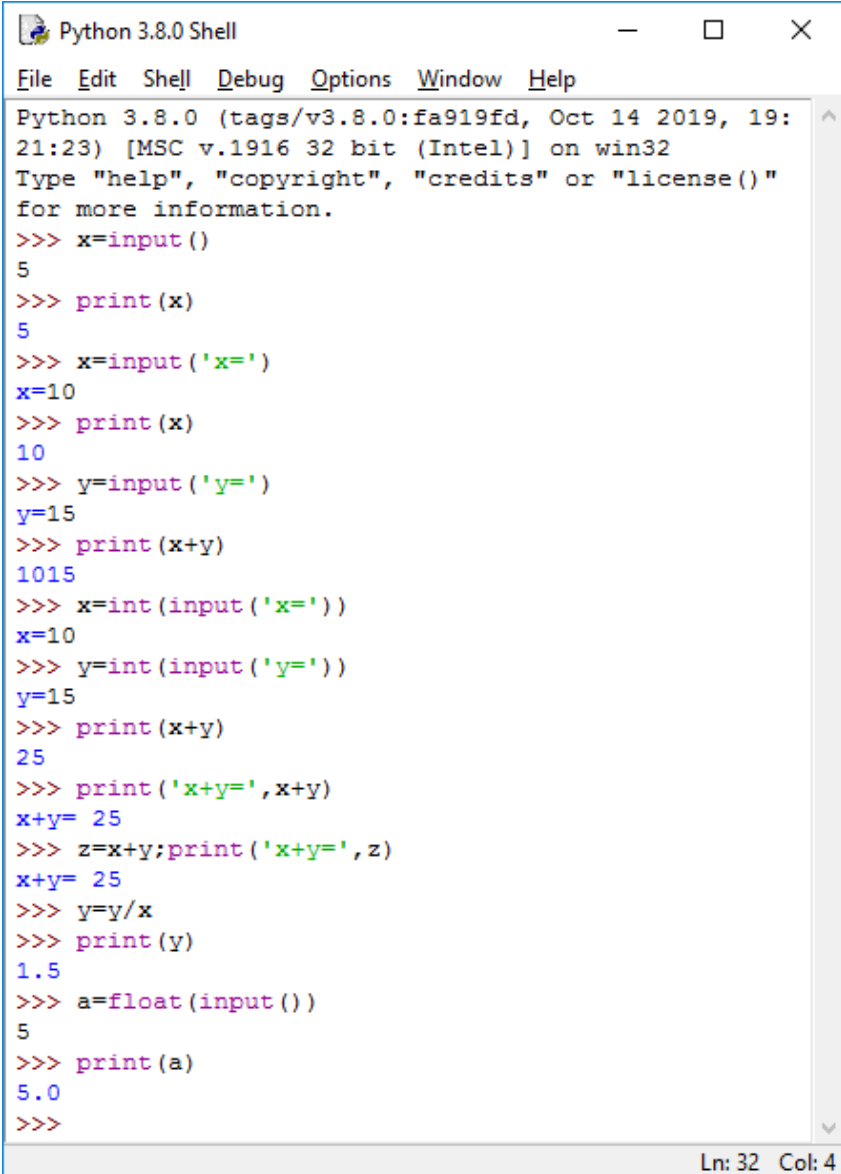
```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:
21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>> print(5)
5
>>> print(5+2)
7
>>> print('5+2')
5+2
>>> print("5+2")
5+2
>>> print('Hello, World!')
Hello, World!
>>> print('abc');print('def')
abc
def
>>> print('abc','def')
abc def
>>> print('abc','def',sep='---')
abc---def
>>> print('abc',end='***');print('def')
abc***def
>>> print('abc'*3)
abcabcabc
>>> print('abc'+ 'def')
abcdef
>>> print('abc'+ 'def'*3)
abcdefdefdef
>>>
```

Ln: 2 Col: 72

Рис. 2

- Ознайомитись з виконанням оператора введення в інтерактивному режимі.

Ввести вирази, наведені на Рис.3. Проаналізувати одержані результати.

A screenshot of a Python 3.8.0 Shell window. The window title is "Python 3.8.0 Shell" and it has standard window controls (minimize, maximize, close). The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area shows the following text:

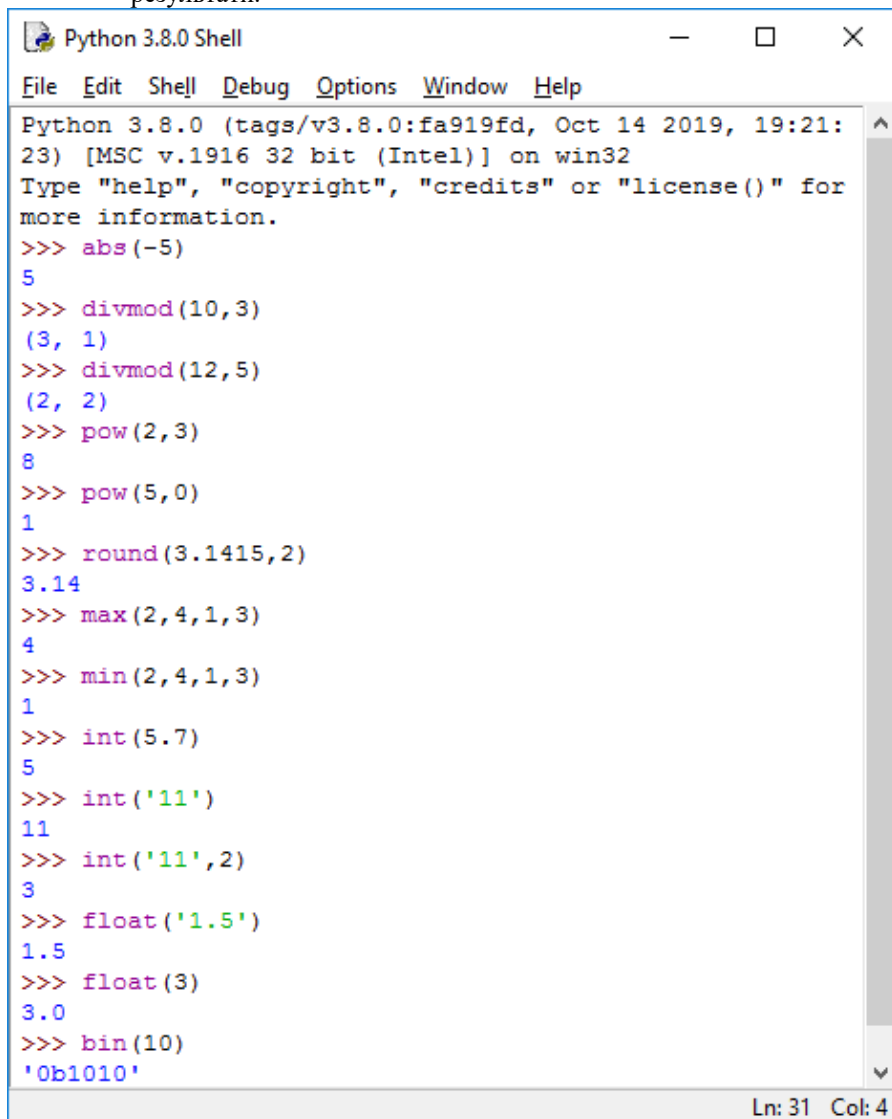
```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:
21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()"
for more information.
>>> x=input()
5
>>> print(x)
5
>>> x=input('x=')
x=10
>>> print(x)
10
>>> y=input('y=')
y=15
>>> print(x+y)
1015
>>> x=int(input('x='))
x=10
>>> y=int(input('y='))
y=15
>>> print(x+y)
25
>>> print('x+y=',x+y)
x+y= 25
>>> z=x+y;print('x+y=',z)
x+y= 25
>>> y=y/x
>>> print(y)
1.5
>>> a=float(input())
5
>>> print(a)
5.0
>>>
```

At the bottom right of the window, it says "Ln: 32 Col: 4".

Рис. 3

5. Ознайомитись з використанням вбудованих функцій опрацювання числових даних в інтерактивному режимі.

Ввести вирази, наведені на Рис.4. Проаналізувати одержані результати.



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> abs(-5)
5
>>> divmod(10,3)
(3, 1)
>>> divmod(12,5)
(2, 2)
>>> pow(2,3)
8
>>> pow(5,0)
1
>>> round(3.1415,2)
3.14
>>> max(2,4,1,3)
4
>>> min(2,4,1,3)
1
>>> int(5.7)
5
>>> int('11')
11
>>> int('11',2)
3
>>> float('1.5')
1.5
>>> float(3)
3.0
>>> bin(10)
'0b1010'
```

Ln: 31 Col: 4

Рис. 4

Робота в пакетному режимі

Приклади розв'язування задач

Завдання 1. Дано довжину ребра куба. Знайти об'єм та площу поверхні куба.

Виконання.

Текст програми може бути таким:

```
a=int(input('a='))
v=a**3
s=6*a*a
print('V=',v,'S=',s)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
a=1 V=1 S=6
a=3 V=27 S=54
```

Завдання 2. Задано тризначне число ABC . Визначити його цифри A , B , C і знайти їх суму.

Виконання.

Текст програми може бути таким:

```
abc=int(input('Chislo abc='))
c=abc%10; print('c=',c)
abc=abc//10
b=abc%10; print('b=',b)
a=abc//10; print('a=',a)
s=a+b+c; print('suma tsyfr chisla abc=',s)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
ABC=123 A=1, B=2 C=3 S=6
ABC=987 A=9 B=8 C=7 S=24
```

Завдання 3. Написати програму обчислення значення виразу

$$e^{x+y} + \frac{5}{\cos(y-x)+3}$$

x , y – дійсні числа.

Виконання.

Текст програми може бути таким:

```
import math
x=float(input('x= '))
```

```
y=float(input('y= '))
r=math.exp(x+y)+5/(math.cos(y-x)+3)
print('Значення виразу = ', r)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
x=0 y=0 Значення виразу = 2.25
x=1 y=1 Значення виразу = 8.63905609893065
x=-4 y=6 Значення виразу = 9.70287632355987
x=-10 y=1 Значення виразу = 1.664334977208902
```

Завдання 4. Задано цілі числа x і y . Обчислити значення виразу $z(x,y)=\ln(x/y)$. Передбачити опрацювання винятків, які можуть виникнути при введенні даних, що не належать області визначення функції z . У випадку виняткової ситуації вивести повідомлення про помилку.

Виконання.

Текст програми може бути таким:

```
import math
try:
    x=int(input('x='))
    y=int(input('y='))
    z=math.log(x/y)
    print(z)
except Exception:
    print('error')
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
x=1 y=1 z=0
x=1 y=1 error
x=1 y=0 error
x=2 y=1 z= 0.6931471805599453
```

1.3. Лабораторна робота №1. Лінійні програми

Зауваження:

1. Інтерфейс програми має бути зрозумілим, тобто запити на введення даних мають бути зрозумілими користувачу.
2. При введенні некоректних вхідних даних чи неможливості проведення обчислень користувачу має виводитися зрозуміле йому повідомлення про помилку українською мовою, і програма має бути стійкою при таких колізіях.
3. Кожне завдання має бути розв'язане в межах окремої програми.
4. Розв'язання завдань не передбачає використання блоків розгалуження.
5. При використанні програми, розробленої для першої задачі передбачається, що користувач не буде вводити даних, які суперечать логіці вхідних змінних.
6. В третій та четвертій задачах використовуються такі позначення: $[x]$ – ціла частина числа x , $\{x\}$ – дробова частина числа x .
7. При розв'язуванні четвертої задачі передбачити виведення як результату обчислення основного виразу, так і результатів проміжних виразів.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Розробити програму, за якою буде обчислюватися середня швидкість автомобіля за заданою відстанню (км) та часом (год), потраченим на проїзд цієї відстані. Вхідні значення є цілими числами.
2. Розробити програму, за якою буде обчислюватися значення функції
$$\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$$
 в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу
$$h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2}$$
, де r - ціле, s та t - дійсні числа, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу
$$S = h \frac{a+b}{2}$$
, де $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$, $z = \sqrt{a^2 + b^2}$, $S = \frac{4}{3} \pi r^3$. Користувачем задаються дійсні числа a , b та c .

Варіант 2.

1. Розробити програму, за якою буде обчислюватися маса тіла за заданим значенням об'єму тіла та щільності матеріалу, з якого його виготовлено. Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $S = a^3 \frac{\sqrt{2}}{12}$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$, де a - ціле, b - дійсне число, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$, де $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$, $S_n = n \frac{a_1 + a_n}{2}$. Користувачем задаються дійсні числа a , b та c .

Варіант 3.

1. Розробити програму, за якою буде обчислюватися довжина та площа кола за заданим значенням радіуса. Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $S_n = \frac{b_1(r^n - 1)}{r - 1}$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $r \neq 1$, де x - ціле, y та z - дійсні числа, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$, де $f(a,b) = \frac{a}{b^2+1} + \frac{b}{a^2+1}$, $\frac{t(a,b)}{2+t(a^2,b-3)} + t(a,b-3) + f(\pi,0)$. Користувачем задаються дійсні числа a , b та c .

Варіант 4.

1. Розробити програму, за якою буде обчислюватися об'єм цеглини за

заданими її розмірами (довжина, ширина, висота). Вхідні дані є цілими числами.

2. Розробити програму, за якою буде обчислюватися значення функції $t(x,y)=y^3+xy+32$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $\frac{\min(a,\min(a+b,ab))}{\min(a^2,b^2)+\min(\sqrt{|a|},\sqrt{|b|})}$, де a - ціле, b - дійсне число, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $f(p+1) - f(p)$, де $f(x)=a_5x^5+a_4x^4 + \dots + a_1x+a_0$, $g(1.2,s)+g(t,s) - g(2s - 1,st)$. Користувачем задаються дійсні числа a , b та c .

Варіант 5.

1. Розробити програму, за якою значення ваги, заданої в кг, буде переводитися у фунти (1 фунт=400 г). Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $g(a,b) = \frac{a^2+b^2}{a^2+b^2+4}$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $y = \frac{1}{x} + \sqrt{x+3} + 6$, де a - ціле, b та c - дійсні числа, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $\frac{1}{8} \cdot \frac{1}{27} \cdot \frac{1}{64} \cdot \dots \cdot \frac{1}{i^3} \cdot \dots$, де $\sum_{i=0}^{\infty} \frac{2i^2}{i!+1}$, $y = \sqrt{\sin(x)} + x^2$, $\frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \frac{9}{5} + \dots + \frac{2i+1}{i+1} + \dots$. Користувачем задаються дійсні числа a , b та c .

Варіант 6.

1. Розробити програму, за якою буде обчислюватися периметр правильного многокутника за довжиною його сторони та кількістю сторін. Вхідні значення є дійсними числами.

2. Розробити програму, за якою буде обчислюватися значення функції $A_i =$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $y = \frac{1}{\sqrt{9-x^2}} + \ln(2-x)$, де a - ціле, b - дійсне число, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $\frac{2}{3} \cdot \frac{4}{9} \cdot \frac{6}{19} \cdot \frac{8}{33} \cdot \dots \cdot \frac{2i}{2i^2+1} \cdot \dots$, де $\sum_{i=1}^{\infty} \frac{(-1)^i}{i!}$, $y = \frac{1}{16-x^2} + \sqrt{x+3}$, $\frac{2}{1} + \frac{3}{1 \cdot 2} + \frac{4}{1 \cdot 2 \cdot 3} + \frac{5}{1 \cdot 2 \cdot 3 \cdot 4} + \dots + \frac{i+1}{i!} + \dots$. Користувачем задаються дійсні числа a , b та c .

Варіант 7.

1. Розробити програму, за якою буде обчислюватися площа прямокутника за введеними значеннями довжин його сторін. Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $\sum_{i=0}^{\infty} \frac{(-1)^i x^i}{1+2^i}$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $y = x^2 + 7x + 10$, де a - ціле, b та c - дійсні числа, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $\arctg(x+y) + \sqrt{x-y \cdot z} - a \cdot b \cdot c \cdot z$, де $x = \ln(a+b)$, $y = \frac{\{e^{a+b+c}\}}{|a+b+c|+1}$, $z = a+b - c^2$. Користувачем задаються дійсні числа a , b та c .

Варіант 8.

1. Розробити програму, за якою буде обчислюватися середнє арифметичне двох чисел. Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $z(x,y) = \sqrt{2^x} - \left(y^2 + (4 + \cos(x-y)) \cdot (1+x \cdot y) \right)$ в заданій точці (числа x та y є цілими та задаються користувачем).

3. Розробити програму, за якою буде обчислюватися значення виразу \ln , де z - ціле, x та y - дійсні числа, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $\operatorname{ctg}(t) + g \cdot t \cdot p + \operatorname{arcctg}(p) + g - t$, де $t = a \cdot b \cdot c^3$, $g = a^2 + p = \left\{ \frac{a}{\sin(b)} \right\} + |b^3| - (a+c) + \cos(c-a)$. Користувачем задаються дійсні числа a , b та c .

Варіант 9.

1. Розробити програму, за якою буде визначатися сума грошей в деякій валюті за заданою сумою грошей у гривнях та вказаним курсом (сума в гривнях за одиницю в іншій валюті). Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $z(x,y) = \frac{\cos(x)}{1+3^x} - \left(x - \frac{y}{2+\cos(x+y)} \right)$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $\sqrt{\frac{\sin(a)}{|b|+2}} + a^2 - \{b^2\}$, де a - ціле, b - дійсне число, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $y \cdot x^3 + \sin(\sqrt{y}) + \operatorname{arctg}(x+c)$, де $x = \left[\frac{a+b+c}{2} \right]$, $y = \ln(a - b + 2c)$. Користувачем задаються дійсні числа a , b та c .

Варіант 10.

1. Розробити програму, за якою буде обчислюватися площа прямокутника за заданими значеннями його довжини та ширини. Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $z(x,y) = 2^x + 3^y - \frac{x \cdot y}{8 + \sin(x-y)} + \sqrt{|\sin(x \cdot y)|}$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $\{\sin(a)\} + |b^2 - 1| - \frac{b+a}{b-\cos(c)}$, де a - ціле, b та c - дійсні числа, що задаються користувачем.

4. Розробити програму, за якою буде обчислюватися значення виразу $a+c \cdot t + \arccos\left(\frac{g}{100}\right) + g - t$, де $t = \cos(a) + |b|^3 - \frac{[b+\cos(c)]}{c-\sin(b)}$, $g = a^2 + b^2 + c^2$. Користувачем задаються дійсні числа a, b та c .

Варіант 11.

1. Розробити програму, за якою буде обчислюватися відстань, подолана тілом за заданими швидкістю тіла $\left(\frac{M}{c}\right)$ та часом руху тіла (c). Вхідні дані є цілими числами.
2. Розробити програму, за якою буде обчислюватися значення функції $z(x,y) = \sin(x) + \frac{\ln(1+2^y)}{\sqrt{x^2+y^2+1}} - y + \cos(x+y)$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $\frac{[\sin(a)-\cos(b)]}{a+2b}$, де a - ціле, b - дійсне число, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $\frac{x+y}{\operatorname{arctg}(z)+5} + \frac{z}{\cos(x+y)+5} - \ln(x^3)$, де $x = a^{5+b}$, $y = \sin(a) + \sin(b)$, $z = \begin{bmatrix} a \\ b \end{bmatrix}$. Користувачем задаються дійсні числа a, b .

Варіант 12.

1. Розробити програму, за якою буде обчислюватися об'єм прямокутного паралелепіпеда за заданими значеннями його довжини, ширини та висоти. Вхідні значення є дійсними числами.
2. Розробити програму, за якою буде обчислюватися значення функції $f(x,y) = \frac{\cos(x)}{1+x^2} + \frac{1.5^x}{2+e^{|x+y|}} - (x \cdot (1 - \sin(x - y) + y))$ в заданій точці (числа x та y є цілими та задаються користувачем).
3. Розробити програму, за якою буде обчислюватися значення виразу $\sqrt{2 - \sin(x - 3 \cdot y)} + \left\{ \frac{x-z}{5-z} \right\}$, де z - ціле, x та y - дійсні числа, що задаються користувачем.
4. Розробити програму, за якою буде обчислюватися значення виразу $\sin(x) + \cos\left(\frac{c}{y}\right) + x \cdot y \cdot z + z$, де $x = \sqrt{a+b^2}$, $y = \operatorname{arctg}\left(\frac{a-c}{|b|+1}\right)$,

$z=a+b+c$. Користувачем задаються дійсні числа a , b та c .

1.4. Організація розгалужень. Умовний оператор

Питання для обговорення.

1. Базові алгоритмічні структури. Розгалуження.
2. Умовний оператор if-else.
3. Умовний оператор if-elif-else.
4. Логічні оператори.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1. Задані ціле число x , дійсне y . Обчислити значення функції $f(x,y)$.

$$f(x,y) = \begin{cases} x^2, & \text{якщо } x - \text{парне число,} \\ x, & \text{якщо } x - \text{непарне число.} \end{cases}$$

Виконання.

Текст програми може бути таким:

```
print('Введіть ціле число x, дійсне число y:')
x=int(input('x='))
y=float(input('y='))
if x%2==0:
    f=x**2+y
else:
    f=x-y
print('f(x,y) =', f)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
x=2 y=1      f(x,y)= 5.0
x= 1 y=2     f(x,y)= -1.0
```

Завдання 2. Відома висота h , на яку потрібно розмістити тіло відносно поверхні моря. Визначити, де саме потрібно розмістити тіло відносно поверхні моря: підняти над поверхнею; розташувати на поверхні; занурити під поверхню. Вивести відповідне повідомлення. Розв'язати задачу двома способами:

- а) без використання вкладеного умовного оператора;
- б) з використанням вкладеного умовного оператора.

Виконання.

Текст програми для способу а) може бути таким:

```
if h>0:
    print('Підняти над поверхнею')
if h==0:
    print('Розташувати на поверхні')
if h<0:
    print('Занурити під поверхню')
```

Текст програми для способу б) може бути таким:

```
h=int(input('h='))
if h>0:
    print('Підняти над поверхнею')
else:
    if h==0:
        print('Розташувати на поверхні')
    else:
        print('Занурити під поверхню')
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

$h=5$ Підняти над поверхнею
 $h=-3$ Занурити під поверхню
 $h=0$ Розташувати на поверхні

Завдання 3. Визначити, чи належить точка $A(x, y)$ фігурі, що утворена лініями $y = x^2$, $y = x + 3$

Виконання.

Текст програми може бути таким:

```
print('Введіть координати точки A(x, y)')
x=float(input('x='))
y=float(input('y='))
if (y>=x**2) and (y<=x+3):
    print('Точка A належить фігурі')
else:
    print('точка A не належить фігурі')
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

x=1 y=2 Точка А належить фігурі

x=-1 y=-1 точка А не належить фігурі

x=-1 y=1 Точка А належить фігурі

x=-1 y=3 точка А не належить фігурі

Завдання 4. Розробити програму, в якій за введеним номером дня тижня від 1 до 7 визначається його назва (понеділок, вівторок, ... , неділя). При введенні неправильних даних має виводитись відповідне повідомлення.

Виконання. Текст програми може бути таким:

```
d=int(input('Введіть номер дня тижня: '))
if d==1:
    print('понеділок')
elif d==2:
    print('вівторок')
elif d==3:
    print('середа')
elif d==4:
    print('четвер')
elif d==5:
    print('п_ятниця')
elif d==6:
    print('субота')
elif d==7:
    print('неділя')
else:
    print('такого дня немає')
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

d=3 среда

d=6 субота

d=9 такого дня немає

Завдання 5. За введеним числом (від 0 до 24), яке вважати годинами доби, визначити частину доби (ранок з 6 до 12, день з 12 до 18, вечір з 18 до 23, ніч з 23 до 6).

Виконання. Текст програми може бути таким:

```
t=int(input('Введіть час доби(від 0 до 24: '))
if t>=6 and t<12:
    print('ранок')
```



```

elif 12<=t<18:
    print('день')
elif 18<=t<23:
    print('вечір')
elif 23<=t<=24 or 0<=t<6:
    print('ніч')
else:
    print('такого часу доби немає')

```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```

7 ранок
15 день
2 ніч
30 такого часу доби немає

```

Завдання 6. Відомі координати двох точок $A(x_a, y_a)$ і $B(x_b, y_b)$, а також координати центра $C(x_c, y_c)$ круга радіуса R . Якщо точки лежать всередині круга, то знайти периметр трикутника ABC , інакше знайти периметр трикутника, який утворений заданими точками і початком координат $O(0,0)$.

Виконання. Текст програми може бути таким:

```

print('Введіть координати точки А:')
xa=float(input('xa='))
ya=float(input('ya='))
print('Введіть координати точки В:')
xb=float(input('xb='))
yb=float(input('yb='))
print('Введіть координати точки С:')
xc=float(input('xc='))
yc=float(input('yc='))
print('Введіть радіус R:')
r=float(input('r='))
ab=((xb-xa)**2+(yb-ya)**2)**0.5
if ((xa-xc)**2+(ya-yc)**2<=r**2) and ((xb-xc)**2+(yb-yc)**2<=r**2):
    bc=((xc-xb)**2+(yc-yb)**2)**0.5
    ca=((xa-xc)**2+(ya-yc)**2)**0.5
    pabc=ab+bc+ca
    print('Pabc=', pabc)
else:
    ao=(xa**2+ya**2)**0.5
    ob=(xb**2+yb**2)**0.5
    paob=ab+ao+ob
    print('Paob=', paob)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

xa=5 ya=2

xb=7 yb=4

xc=5 yc=4 r=3

Рabc= 6.82842712474619

xa=0 ya=-1

xb= -1 yb=0

xc=5 yc=4 r=3

Раob= 3.82842712474619

Завдання 7. Дано чотири числа. Якщо сума найменшого і найбільшого більше суми двох інших чисел, то знайти середнє геометричне всіх заданих чисел, інакше знайти їх середнє арифметичне.

Виконання. Текст програми може бути таким:

```
print ('Введіть 4 числа:')
a=int (input ('a='))
b=int (input ('b='))
c=int (input ('c='))
d=int (input ('d='))
s=a+b+c+d
mi=min (a, b, c, d)
ma=max (a, b, c, d)
s1=mi+ma
s2=s-s1
if s1>s2:
    sg=(a*b*c*d)**0.25
    print ('Середнє геометричне чисел=',sg)
else:
    sa=(a+b+c+d)/4
    print ('Середнє арифметичне чисел=',sa)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

a=1 b=2 c=3 d=5

Середнє геометричне чисел= 2.340347319320716

a=6 b=10 c=1 d=6

Середнє арифметичне чисел= 5.75

1.5. Лабораторна робота №2. Розгалуження

Зауваження:

1. Інтерфейс програми має бути зрозумілим, тобто запити на введення даних мають бути зрозумілими.
2. При введенні некоректних вхідних даних чи неможливості проведення обчислень користувачу має виводитися зрозуміле йому повідомлення про помилку українською мовою, і програма має бути стійкою при таких колізіях.
3. Кожне завдання має бути розв'язане в межах окремої програми.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Нехай D – квадрат, вершини якого мають координати $(3,3)$, $(-3,3)$, $(-3,-3)$, $(3,-3)$. $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, де
$$\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$$
.
2. Розробити програму, в якій за вказаним номером пари (від 1 до 4) визначається час початку та кінця пари.
3. Дано координати чотирьох послідовних вершин опуклого чотирикутника. Розробити програму, в якій буде з'ясовуватися, чи є даний чотирикутник прямокутником, квадратом або ромбом.

Варіант 2.

1. Нехай D – перша координатна чверть. $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, якщо
$$h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2}$$
.
2. Скласти програму, в якій за введеним числом від 0 до 23, що позначає години доби, визначається, чи працює магазин у вказаний час (час роботи магазину з 8 до 20, перерва з 12 до 14).
3. Задано два дійсні числа – прямокутні координати x та y деякої точки на площині. Знайти її полярні координати R та φ . Передбачити в програмі, що точка може лежати в будь-якій координатній площині, на одній з осей або в початку координат.

Варіант 3.

1. Нехай D – коло з центром в точці $(3, 2)$ і радіусом 6 . $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, де $S = h \frac{a+b}{2}$.
2. За вказаним числом (від 0 до 6), яке вважати кількістю років навчання студента в ЗВО, визначити його освіту (4 – бакалавр, 5 – спеціаліст, 6 – магістр, менше 4 – без вищої освіти).
3. Дано значення параметрів k і b прямої $y = kx + b$ і координати двох точок $A(x_1, y_1)$ і $B(x_2, y_2)$. Якщо відрізок AB перетинає задану пряму, то визначити точку перетину. Інакше знайти довжину AB .

Варіант 4.

1. Нехай D – четверта координатна чверть. $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, якщо $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$.
2. Розробити програму, в якій за обраною літерною оцінкою в ECTS форматі (F, Fx, E, D, C, B, A) визначаються межі балів за 100 бальною системою.
3. Знайти відстань від точки $z = \sqrt{a^2 + b^2}$ до відрізка з вершинами $S = \frac{4}{3}\pi r^3$ і $S = a^3 \frac{\sqrt{2}}{12}$.

Варіант 5.

1. Нехай D – друга координатна чверть. $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, якщо $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$.
2. За обраною версією операційної системи Windows (не менше 4-х версій) визначити рік її випуску.
3. Задано три числа. Якщо сума двох найменших з них більше третього числа, то знайти середнє геометричне всіх трьох чисел, інакше знайти їх середнє арифметичне.

Варіант 6.

1. Нехай D – третя координатна чверть. $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, якщо $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$.
2. Розробити програму, в якій за вказаною назвою місяця визначається кількість днів в ньому.

3. Дано три дійсні числа. Розробити програму, в якій з'ясується, чи існує трикутник з довжинами сторін, які відповідають даним числам. Якщо такий трикутник існує, то визначити, чи є трикутник рівнобедреним, рівностороннім або прямокутним.

Варіант 7.

1. Дано числа a , b і c (де $a > 0$). Знайти дійсні корені рівняння $ax^2 + bx + c = 0$.
2. Скласти програму, в якій за вказаною оцінкою за п'ятибальною системою визначаються межі балів за ECTS -шкалою (від 90 до 100 – 5, від 75 до 89 – 4, від 60 до 74 – 3, менше 60 – 2).
3. Дано дійсні додатні числа a , b , c , d . Дайте відповідь, чи можна прямокутник із сторонами a і b розмістити всередині прямокутника зі сторонами c і d так, щоб кожна зі сторін одного прямокутника була паралельною або перпендикулярною кожній стороні другого прямокутника.

Варіант 8.

1. Нехай D – квадрат, вершини якого мають координати $(3,0)$, $(0,3)$, $(-3,0)$, $(0,-3)$. $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, де $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$.
2. Скласти програму, яка за вказаним номером пари від 1 до 3 (4) визначає назву відповідної пари у власній групі у вівторок.
3. Трикутник ABC задано координатами своїх вершин на площині. Визначити, де лежить точка $S_n = n \frac{a_1 + a_n}{2}$ – зовні трикутника, всередині нього або на стороні трикутника.

Варіант 9.

1. З'ясувати, в якій координатній чверті знаходиться трикутник, утворений прямою, заданою рівнянням $y = ax + b$ і осями координат.
2. Розробити програму, в якій за вказаним номером кварталу (періоди по 3 місяці від початку року) визначаються назви місяців, що входять до нього.
3. Дано координати двох центрів кіл та їх радіуси $O(x_1, y_1)$, r_1 , $O(x_2, y_2)$, r_2 . Вивести, як розташовані ці кола (перетинаються, дотикаються, співпадають, знаходяться одне в іншому, не мають спільних точок).

Варіант 10.

1. Дано дійсні числа x і y . З'ясувати, в якій координатній чверті знаходиться точка з координатами (x, y) .
2. Скласти програму, яка за введеним номером місяця від 1 до 12 визначає пору року – осінь, зима, весна, літо.
3. Дано три дійсні числа. Розробити програму, в якій з'ясується, чи існує трикутник з довжинами сторін, які відповідають даним числам. Якщо такий трикутник існує, то визначити, де знаходиться центр описаного кола (зовні трикутника, на стороні трикутника, всередині трикутника).

Варіант 11.

1. Нехай D – четверта координатна чверть. $A(x, y)$ – будь-яка точка площини. Знайти $f(x, y)$, якщо $S_n = \frac{b_1(r^n - 1)}{r - 1}$.
2. Скласти програму, яка за введеним числом від 0 до 100, що є віком людини, визначає, до якої вікової групи належить людина (від 0 до 16 – дитина, від 17 до 21 – юнак, від 22 до 60 – доросла людина, від 61 до 100 – літня людина).
3. Визначити проміжок, на якому виконується система нерівностей $r \neq 1$. Числа a, b, c, d вводяться користувачем. В програмі не передбачається введення нульових коефіцієнтів.

Варіант 12.

1. Визначити, чи належить точка $A(x, y)$ фігурі, що утворена лініями $n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$, $y = x - 3$, $y = x + 3$.
2. Розробити програму, в якій за вказаною назвою предмету (не менше 4 предметів з власної групи) визначається прізвище викладача, який читає лекції.
3. Три точки задані своїми координатами $A(x_1, y_1)$, $B(x_2, y_2)$ і $C(x_3, y_3)$. Знайти ту з трьох точок, яка лежить найближче до початку координат. Координати інших двох точок замінити координатами середини відрізка, вершинами якого є ці точки.

1.6. Організація циклів. Оператори циклу з передумовою. Цикл з параметром. Вкладені цикли.

Питання для обговорення.

1. Базові алгоритмічні структури. Цикл.
2. Цикл з передумовою while.
3. Цикл з параметром for.
4. Інструкції управління циклами.
5. Вкладені цикли.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Цикл з передумовою

Завдання 1а. Обчислити суму цілих додатних непарних чисел, менших 50.

Виконання.

Текст програми може бути таким:

```
i=1 #перше потрібне число - ціле, додатне, непарне,
менше 50
S=0 #початкове значення суми
while i<50:
    S=S+i
    i=i+2 #наступне потрібне число
print('S=', S)
```

Ввести програму, запустити її на виконання, перевірити роботу програми:

S=625

Завдання 1b.

Змінити програму так, щоб у ній обчислювався добуток цілих додатних непарних чисел, менших за 10.

Завдання 1c. Обчислити суму всіх натуральних чисел від 1 до 200, остання цифра яких - нуль.

Виконання.

Текст програми може бути таким:

```
i=10          #перше потрібне число - натуральне, <200,
              закінчується 0
S=0           #початкове значення суми
while i<=200:
    S=S+i
    i=i+10   #наступне потрібне число
print('S=',S)
```

Ввести програму, запустити її на виконання, перевірити роботу програми:
S=2100

Завдання 1d. Змінити програму так, щоб у ній обчислювалась сума всіх натуральних чисел від 1 до 80, остання цифра яких - 5.

Завдання 2a. Обчислити суму перших k додатних членів послідовності:
 $A_n = \sin(n)$. (k вводиться користувачем).

Виконання.

Текст програми може бути таким:

```
import math
n=1      #лічильник усіх членів послідовності
S=0      #початкове значення суми
i=0      #лічильник потрібних - додатних - членів п-сті
k=int(input('k='))
while i<k:
    a=math.sin(n) #обчислення чергового члена п-сті
    if a>0:      #дії у випадку додатного члена п-сті
        S=S+a
        i=i+1
    n=n+1
print('S=',S)
```


Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

k=1 S= 0.8414709848078965

k=2 S= 1.7507684116335782

k=5 S= 3.538233265035616

Завдання 2b. Змінити програму так, щоб у ній виводились на екран додатні члени послідовності: $A_n = \sin(n)$ разом з їхніми номерами у вигляді: $n - a_n$.

Завдання 3a. Виконати табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $y = x^2 - 3x + 2$ на відрізку $[a, b]$ з кроком h . Дійсні числа a, b, h задаються користувачем.

Виконання.

```
print('введіть a, b, a<b, h>0')
a=float(input('a='))
b=float(input('b='))
h=float(input('h='))
x=a
while x<=b:
    y=x*x-3*x+2
    print('{:.4}     {:.4}'.format(x,y)) #форматування
при виведенні
    x=x+h
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

a=1 b=3 h=0.5

1.0 0.0

1.5 -0.25

2.0 0.0

2.5 0.75

3.0 2.0

Цикл з параметром

Завдання 4a. Ввести ціле число N . Вивести всі непарні натуральні числа, що не перевищують N .

Виконання.

Розглянемо два варіанти розв'язку цієї задачі.

Текст програми може бути таким:

```
n=int(input('N='))
for i in range(1,n+1):
    if i%2 !=0:
        print(i)
```

Або таким:

```
n=int(input('N='))
for i in range(1,n+1,2):
    print(i)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
n=6
1
3
5
```

Завдання 4b. Змінити програму так, щоб у ній виводились квадрати цілих чисел від 1 до N.

Завдання 4c. Змінити програму так, щоб у ній обчислювалася сума непарних натуральних чисел від 1 до N.

Завдання 4d. Змінити програму так, щоб у ній обчислювався добуток парних натуральних чисел від 1 до N.

Завдання 4e. Обчислити суму перших N членів послідовності.

$$\left(1 - \frac{1}{2}\right) + \left(1 - \frac{1}{2 \cdot 3}\right) + \left(1 - \frac{1}{2 \cdot 3 \cdot 4}\right) + \left(1 - \frac{1}{2 \cdot 3 \cdot 4 \cdot 5}\right) + \dots \\ + \left(1 - \frac{1}{i!}\right) + \dots$$

Виконання.

Розглянемо два варіанти розв'язку цієї задачі.

Текст програми може бути таким:

```
n=int(input('N='))
s=0          #початкове значення суми
c=1         #початкове значення чисельника доданку
z=2         #початкове значення знаменника доданку
for i in range(1,n+1):
    s=s+c/z; #додавання чергового доданку до суми
    c=c+1;  #збільшення чисельника доданку
    z=z+1   #збільшення знаменника доданку
print('S={:.4}'.format(s))
```

Або таким:

```
n=int(input('N='))
s=0
```

```

for i in range(1,n+1):
    s=s+i/(i+1) #доданок обчисл. за загальною формулою
print('S={:.4}'.format(s))

```

Ввести обидва варіанти програми, запустити їх на виконання, перевірити роботу програми на контрольних прикладах:

```

n=1    S=0.5
n=2    S=1.167
n=5    S=3.55

```

Завдання 4f. Змінити програму так, щоб у ній обчислювався добуток перших N членів указаної послідовності.

Завдання 4g. Обчислити добуток перших N членів послідовності:

$$\frac{1}{2} \times \frac{1}{4} \times \frac{1}{8} \times \frac{1}{16} \times \dots$$

Виконання.

Розглянемо два варіанти розв'язку цієї задачі.

Текст програми може бути таким (з покроковим збільшенням чисельника і знаменника):

```

n=int(input('N='))
p=1 #початкове значення добутку
c=1 #початкове значення чисельника множника
z=2 #початкове значення знаменника множника
for i in range(1,n+1):
    p=p*c/z #множення чергового множника на добуток
    z=z*2 #збільшення знаменника множника
print('P={:.4}'.format(p))

```

Або таким (з використанням загальної формули для множника):

```

n=int(input('N='))
p=1
for i in range(1,n+1):
    p=p*1/pow(2,i)
print('P={:.4}'.format(p))

```

Ввести обидва варіанти програми, запустити їх на виконання, перевірити роботу програми на контрольних прикладах:

```

n=1    P=0.5
n=2    P=0.125
n=5    P=3.052e-05

```

Завдання 4h. Обчислити суму перших N членів послідовності

$$\frac{1}{2} + \frac{3}{4} + \frac{5}{6} + \frac{6}{7} + \dots + \frac{2i-1}{2i} + \dots$$

Виконання.

Розглянемо два варіанти розв'язку цієї задачі.

Текст програми може бути таким (з покроковим збільшенням чисельника і знаменника):

```
n=int(input('N='))
s=0
c=1
z=2
for i in range(1,n+1):
    s=s+c/z
    c=c+2
    z=z+2
print('S={:.4}'.format(s))
```

Або таким (з використанням загальної формули для множника):

```
n=int(input('N='))
s=0
for i in range(1,n+1):
    s=s+(2*i-1)/(2*i)
print('S={:.4}'.format(s))
```

Ввести обидва варіанти програми, запустити їх на виконання, перевірити роботу програми на контрольних прикладах:

```
n=1    S=0.5
n=2    S=1.25
n=5    S=3.858
```

Завдання 4i. Обчислити суму перших N членів послідовності

$$\left(1 - \frac{1}{2}\right) + \left(1 - \frac{1}{2 \cdot 3}\right) + \left(1 - \frac{1}{2 \cdot 3 \cdot 4}\right) + \left(1 - \frac{1}{2 \cdot 3 \cdot 4 \cdot 5}\right) + \dots + \left(1 - \frac{1}{i!}\right) + \dots$$

Виконання.

Текст програми може бути таким:

```
n=int(input('N='))
s=0
f=1 #початкове значення факторіала в знаменнику
for i in range(2,n+2):
    f=f*i #обчислення факторіала в знаменнику
    s=s+(1-1/f)
print('S={:.4}'.format(s))
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
n=1    S=0.5
n=2    S=1.333
```

n=5 S=4.282

Завдання 4j. Обчислити суму ряду $\sum_{i=1}^N$ для заданого N.

Виконання.

Текст програми може бути таким:

```
n=int(input('N='))
s=0
w=-1 #чисельник першого доданку
for i in range(1,n+1):
    s=s+w/(i*(i+1))
    w=-w #зміна знаку доданка
print('S={:.4}'.format(s))
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
n=1    S=-0.5
n=2    S=-0.3333
n=5    S=-0.4
```

Завдання 5a. Задано ціле число N. Сформувати і вивести таблицю з N рядків та N стовпців вказаного виду (наприклад, для N=4):

```
* * * *
* * * *
* * * *
* * * *
```

Виконання.

Текст програми може бути таким:

```
n=int(input('N='))
s=0
for i in range(1,n+1): #опрацювання рядків
    for j in range(1,n+1):#опрацювання ел-тів у рядку
        print('* ',end=' ') #друк рядка з зірочок
    через пропуск
    print(' ') #перехід на новий рядок
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

```
n=1    n=2    n=3    n=4    n=5
```

Завдання 5b. Задано ціле число N. Сформувати і вивести таблицю з N рядків та N стовпців вказаної структури (наприклад, для N=4):

```
110 120 130 0
210 220 0 240
310 0 330 340
0 420 430 440
```

Виконання.

Текст програми може бути таким:

```
n=int(input('N='))
for i in range(1,n+1): #опрацювання рядків
    for j in range(1,n+1):#опрацювання ел-тів у рядку
        if i+j==n+1:
            print(0,end=' ') #на побічній діагоналі
таблиці елементи =0
        else:
            print(100*i+10*j,end=' ')#інші ел-ти
обчислюються за формулою
            print() #перехід на новий рядок (інший спосіб)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

n=1 n=2 n=3 n=4 n=5

Завдання 5с. Задано ціле число N. Сформувати і вивести таблицю з N рядків та N стовпців вказаної структури (наприклад, для N=4):

```
101    102    103    104
100    200    300    400
301    302    303    304
100    200    300    400
```

Виконання.

Текст програми може бути таким:

```
n=int(input('N='))
for i in range(1,n+1):
    for j in range(1,n+1):
        if i%2==1: #ел-ти у непарних рядках
            print(100*i+j,end='\t') #виводяться через
табуляцію
        else:
            print(100*j,end='\t') #елементи у парних
рядках
            print()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

n=1 n=2 n=3 n=4 n=5

Завдання 5d. Змінити програму так, щоб у ній виводилась таблиця виду

```
10 10 10 10
21 22 23 24
30 30 30 30
41 42 43 44
```

1.7. Лабораторна робота №3. Цикли

Зауваження:

1. Інтерфейс програми має бути зрозумілим.
2. Кожне завдання має бути розв'язане в межах окремої програми.
3. При розв'язуванні першої задачі передбачити виведення розглядуваних членів послідовності та їх порядкових номерів.
4. При розв'язуванні другої задачі передбачити задання кількості рядків (стовпців).
5. При розв'язуванні третьої задачі передбачити виведення розглядуваних членів послідовності та їх порядкових номерів.
6. При розв'язуванні третьої задачі передбачити перегляд не більше 100 членів послідовності, навіть якщо потрібна точність не досягнута.
7. При розв'язуванні третьої задачі має використовуватися цикл *while*.
8. При розв'язуванні четвертої задачі кінці відрізка та крок табулювання вважати дійсними числами.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Розробити програму, в якій передбачити обчислення суми перших N членів послідовності $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$.
2. Розробити програму, в якій за заданим цілими числами N та M ($2 \leq N < 99$; $2 \leq M < 99$) формується таблиця з N рядків та M стовпців наступної структури (наприклад, для $N=4$ та $M=4$):
11 12 13 14
20 20 20 20
31 32 33 34
40 40 40 40
3. Розробити програму, в якій передбачити обчислення суми ряду $h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2}$ із заданою точністю ε , вважаючи, що потрібна точність досягнута, якщо модуль наступного доданку менший за ε .
4. Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $S=h \frac{a+b}{2}$ на відріжку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо значення x виходить за межі області допустимих

значень (в цьому випадку має виводитися відповідне повідомлення).

Варіант 2.

1. Створити програму для обчислення суми перших N членів послідовності $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$.
2. Розробити програму, в якій за заданим цілим числом N ($2 \leq N < 99$) формується таблиця з N рядків та N стовпців наступної структури (наприклад, для $N=4$):

```
1 0 1 0
0 2 0 2
3 0 3 0
0 4 0 4
```

3. Розробити програму, в якій передбачити обчислення суми ряду $z = \sqrt{a^2 + b^2}$ із заданою точністю ε , вважаючи, що потрібна точність досягнута, якщо модуль наступного доданку менший за ε .
4. Створити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $S = \frac{4}{3} \pi r^3$ на відріжку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо два обчислених значення u перевищать деяке вказане користувачем число z (у цьому випадку виводиться відповідне повідомлення).

Варіант 3.

1. Розробити програму, в якій передбачити обчислення добутку перших N членів послідовності $S = a^3 \frac{\sqrt{2}}{12}$.
2. Розробити програму, в якій за заданим цілим числом N ($2 \leq N < 99$) формується таблиця з N рядків наступної структури (наприклад, для $N=4$):

```
11 12 13 14
0 22 23 24
0 0 32 34
0 0 0 44
```

3. Розробити програму, в якій передбачити обчислення суми ряду $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ із заданою точністю ε , вважаючи, що

потрібна точність досягнута, якщо модуль наступного доданку менший за ε .

4. Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо два обчислені значення y будуть дорівнювати деякому вказаному користувачем значенню z (у цьому випадку має виводитися відповідне повідомлення).

Варіант 4.

1. Розробити програму, в якій передбачити обчислення добутку перших N членів послідовності $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$.
2. Розробити програму, в якій за заданим цілим числом N ($2 \leq N < 99$) формується таблиця з N рядків та N стовпців наступної структури (наприклад, для $N=4$):

1	2	3	4
201	202	203	204
1	2	3	4
401	402	403	404
3. Створити програму для обчислення суми перших N членів послідовності $S_n = n \frac{a_1 + a_n}{2}$, більших за задане число K .
4. Створити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $S_n = \frac{b_1(r^n - 1)}{r - 1}$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо значення x виходить за межі області допустимих значень (у цьому випадку виводиться відповідне повідомлення).

Варіант 5.

1. Розробити програму, в якій передбачити обчислення добутку перших N членів послідовності $r \neq 1$.
2. Розробити програму, в якій за заданим цілими числами N та M ($2 \leq N < 30$; $2 \leq M < 30$) формується таблиця з N рядків та M стовпців наступної структури (наприклад, для $N=3$ та $M=4$):

```
10 11 12 13 14
20 21 22 23 24
30 31 32 33 34
```

3. Визначити кількість членів ряду, необхідних для обчислення числа π із заданою точністю ε за формулою $n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$.
4. Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $f(a,b) = \frac{a}{b^2+1} + \frac{b}{a^2+1}$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо три обчислені значення у будуть більші деякого вказаного користувачем значення z (у цьому випадку має виводитися відповідне повідомлення).

Варіант 6.

1. Створити програму для обчислення суми перших N членів послідовності $\frac{t(a,b)}{2+t(a^2,b-3)} + t(a,b-3) + f(\pi,0)$.
2. Розробити програму, в якій за заданим цілим числом N ($1 \leq N$) формується таблиця з N рядків наступної структури (наприклад, для $N=4$):

```
110 120 130 140
 1  2  3  4
310 320 330 340
 1  2  3  4
```

3. Розробити програму, в якій передбачити обчислення суми ряду $t(x,y) = y^3 + xy + 32$ із заданою точністю ε , вважаючи, що потрібна точність досягнута, якщо модуль наступного доданку менший за ε .
4. Створити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції

$\frac{\min(a, \min(a+b, ab))}{\min(a^2, b^2) + \min(\sqrt{|a|}, \sqrt{|b|})}$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо два обчислені значення у дорівнюють нулю (в цьому випадку виводиться відповідне повідомлення).

Варіант 7.

1. Створити програму для обчислення суми перших N членів послідовності $f(p+1) - f(p)$.
2. Розробити програму, в якій за заданим цілим числом N ($2 \leq N < 99$) формується таблиця з N рядків та N стовпців наступної структури (наприклад, для $N=4$):

```

11 12 13  0
21 22  0 24
31  0 33 34
 0 42 43 44

```

3. Створити програму для обчислення суми ряду $f(x) = a_5 x^5 + a_4 x^4 + \dots + a_1 x + a_0$ із заданою точністю ϵ , вважаючи, що потрібна точність досягнута, якщо модуль наступного доданку менший за ϵ .
4. Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $g(1.2, s) + g(t, s) - g(2s - 1, st)$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо два обчислені значення у будуть дорівнювати 0 (в цьому випадку має виводитися відповідне повідомлення).

Варіант 8.

1. Розробити програму, в якій передбачити обчислення суми перших N членів послідовності $g(a, b) = \frac{a^2 + b^2}{a^2 + b^2 + 4}$.
2. Розробити програму, в якій за заданим цілим числом N ($2 \leq N < 99$) формується таблиця з N рядків та N стовпців наступної структури (наприклад, для $N=4$):

110 0 130 0
 210 0 230 0
 310 0 330 0
 410 0 430 0

- Створити програму для виведення на екран послідовності чисел Фібоначчі. Виведення припинити, якщо останнє виведене число буде націло ділитись на деяке вказане користувачем число N .
- Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $y = \frac{1}{x} + \sqrt{x+3} + 6$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо значення x виходить за межі області допустимих значень (в цьому випадку має виводитися відповідне повідомлення).

Варіант 9.

- Розробити програму, в якій передбачити обчислення добутку перших N членів послідовності $\frac{1}{8} \cdot \frac{1}{27} \cdot \frac{1}{64} \cdot \dots \cdot \frac{1}{i^3} \cdot \dots$.
- Розробити програму, в якій за заданим цілими числами N та M ($2 \leq N < 30$; $2 \leq M < 30$) формується таблиця з N рядків та M стовпців наступної структури (наприклад, для $N=4$ та $M=4$):

110 0 0 0
 210 220 0 0
 310 320 330 0
 410 420 430 440

- Розробити програму, в якій передбачити обчислення суми ряду $\sum_{i=0}^{\infty} \frac{2i^2}{i+1}$ із заданою точністю ε , вважаючи, що потрібна точність досягнута, якщо модуль наступного доданку менший за ε .
- Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $y = \sqrt{\sin(x)} + x^2$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо значення x виходить за межі області допустимих значень (в цьому випадку має виводитися відповідне повідомлення).

Варіант 10.

1. Створити програму для обчислення суми перших N членів послідовності $\frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \frac{9}{5} + \dots + \frac{2i+1}{i+1} + \dots$.
2. Розробити програму, в якій за заданим цілим числом N ($2 \leq N < 99$) формується таблиця з N рядків та N стовпців наступної структури (наприклад, для $N=4$):
0 12 13 14
21 0 23 24
31 32 0 34
41 42 43 0
3. Створити програму для обчислення суми перших N членів послідовності $A_i =$, більших за задане число K .
4. Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $y = \frac{1}{\sqrt{9-x^2}} + \ln(2-x)$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо значення x виходить за межі області допустимих значень (в цьому випадку має виводитися відповідне повідомлення).

Варіант 11.

1. Розробити програму, в якій передбачити обчислення добутку перших N членів послідовності $\frac{2}{3} \cdot \frac{4}{9} \cdot \frac{6}{19} \cdot \frac{8}{33} \cdot \dots \cdot \frac{2i}{2i^2+1} \cdot \dots$.
2. Розробити програму, в якій за заданим цілим числом N ($2 \leq N < 99$) формується таблиця з N рядків та N стовпців наступної структури (наприклад, для $N=4$):
2 3 4 5
0 0 0 0
4 5 6 7
0 0 0 0
3. Розробити програму, в якій передбачити обчислення суми ряду $\sum_{i=1}^{\infty} \frac{(-1)^i}{i!}$ із заданою точністю ε , вважаючи, що потрібна точність досягнута, якщо модуль наступного доданку менший за ε .
4. Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $y = \frac{1}{16-x^2} + \sqrt{x+3}$ на відрізку $[a, b]$ з кроком h . Передбачити в

програмі припинення табулювання, якщо значення x виходить за межі області допустимих значень (в цьому випадку має виводитися відповідне повідомлення).

Варіант 12.

1. Створити програму для обчислення суми перших N членів послідовності $\frac{2}{1} + \frac{3}{1 \cdot 2} + \frac{4}{1 \cdot 2 \cdot 3} + \frac{5}{1 \cdot 2 \cdot 3 \cdot 4} + \dots + \frac{i+1}{i!} + \dots$.

2. Розробити програму, в якій за заданими цілими числами N та M ($2 \leq N < 30$; $2 \leq M < 30$) формується таблиця з N рядків та M стовпців наступної структури (наприклад, для $N=3$ та $M=6$):

```
1 0 2 0 3 0
3 0 4 0 5 0
5 0 6 0 7 8
```

3. Розробити програму, в якій передбачити обчислення суми ряду $\sum_{i=0}^{\infty} \frac{(-1)^i i}{1+2^i}$ із заданою точністю ε , вважаючи, що потрібна точність досягнута, якщо модуль наступного доданку менший за ε .

4. Розробити програму для табулювання (виведення таблиці значень аргументу та відповідного йому результату) функції $y=x^2 + 7x+10$ на відрізку $[a, b]$ з кроком h . Передбачити в програмі припинення табулювання, якщо два обчислені значення y будуть дорівнювати 0 (в цьому випадку має виводитися відповідне повідомлення).

2. ОСНОВНІ СТРУКТУРИ ДАНИХ У PYTHON. ФУНКЦІЇ КОРИСТУВАЧА

2.1. Структури даних. Списки

Питання для обговорення.

1. Списки.
2. Задання списків.
3. Доступ до елементів списку. Зрізи.
4. Зміна та вилучення елементів списку.
5. Змінюваність типу список.
6. Методи списків.
7. Порівняння списків.
8. Вкладені списки.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Ввести список з 10 цілих чисел. Вивести елементи списку в рядок. Обчислити суму, добуток і середнє арифметичне елементів масиву.

Виконання.

Текст програми може бути таким:

```
#введення елементів списку з клавіатури
```

```
a=[] #початкове значення списку
```

```
for i in range(10):
```

```
    x=int(input('a['+str(i)+']=')) #введення елемента
```

```
списку
```

```
    a+=[x] #поповнення списку елементом
```

```

#виведення елементів списку в рядок
print('Список:')
for x in a:
    print(x, end=' ')
print()
#обчислення суми елементів списку
s=sum(a)
print('Сума = ',s)
#обчислення добутку елементів списку
d=1
for x in a:
    d=d*x
print('Добуток = ',d)
#обчислення середнього арифметичного елементів списку
sa=s/len(a)
print('Середнє арифметичне = ',sa)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольному прикладі:

a[0]=1

a[1]=1

a[2]=1

a[3]=1

a[4]=1

a[5]=1

a[6]=1

a[7]=1

a[8]=1

a[9]=1

Список:

1 1 1 1 1 1 1 1 1 1

Сума = 10

Добуток = 1

Середнє арифметичне = 1.0

Завдання 1b. Створити список з указаної користувачем кількості цілих чисел (не більше 10). Подвоїти всі додатні елементи списку. Вивести елементи перетвореного списку в стовпчик.

Виконання.

Текст програми може бути таким:

```
#введення списку
n=int(input('Введіть кількість елементів списку = '))
print('Введіть елементи списку через Enter')
a=[]
for i in range(n):
    x=int(input())
    a+= [x]
#перетворення елементів списку
for i in range(n):
    if a[i]>0:
        a[i]=a[i]*2
#виведення елементів списку
print('Перетворений список:')
for x in a:
    print(x)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах.

Введіть кількість елементів списку = 4

Введіть елементи списку через Enter

-2

-3

2

3

Перетворений список:

-2

-3

4

6

Завдання 1с. Змінити програму так, щоб усі від'ємні елементи списку замінювались на їх абсолютні величини, а елементи зміненого списку виводились у рядок.

Завдання 1а. Сформувати список з уведеної користувачем кількості цілих псевдовипадкових чисел (кількість елементів не більше 10, значення елементів не перевищують 100). Вивести одержаний список як перелік елементів у квадратних дужках. Визначити кількість парних елементів у ньому.

Виконання.

Текст програми може бути таким:

```
import random          #використання генератора
псевдовипадкових чисел
n=int(input('введіть n (n<=10): '))
a=[]                  #початкове значення
списку
for i in range(n):
    x=random.randint(1,100)      #генерування
псевдовипадкового числа
    a+=[x]                      #поповнення списку
новим елементом
print(a)                #виведення списку
#обчислення кількості парних елементів
k=0
for i in range(n):
    if a[i]%2==0:
        k+=1
print('Кількість парних елементів списку = ',k)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах.

Введіть n (n<=10): 7

[82, 60, 79, 91, 52, 59, 50]

Кількість парних елементів списку = 4

або таким:

```
import random          #використання генератора
псевдовипадкових чисел
n=int(input('Введіть n (n<=10): '))
#генерування списку з n псевдовипадкових чисел
a=[random.randint(1, 100) for i in range(n)]
print('Список: ',a)      #виведення списку
#обчислення кількості парних елементів
k=0
for i in range(n):
    if a[i]%2==0:
        k+=1
print('Кількість парних елементів списку = ',k)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах.

Введіть n (n<=10): 7

Список: [87, 41, 87, 90, 85, 10, 5]

Кількість парних елементів списку = 2

Завдання 1e. Змінити програму так, щоб у ній обчислювалась сума непарних елементів списку.

Завдання 1f. Ввести списки А і В (кожний з n цілих псевдовипадкових чисел, n<=10, елементи списків перебувають у межах від 1 до 20). Сформувати список С за правилом:

$$c_i = a_i^2 + b_i^2, i=0, \dots, n-1.$$

Текст програми може бути таким:

```
import random
n=int(input('Введіть n (n<=10): '))
a=[random.randint(1, 20) for i in range(n)]
b=[random.randint(1, 20) for i in range(n)]
print('Елементи списку a: ', a)
print('Елементи списку b: ', b)
c=[]
for i in range(n):
    c+=[a[i]**2+b[i]**2]
    #c.append(a[i]**2+b[i]**2) інший спосіб поповнення
    списку
print('Елементи списку c: ', c)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах.

Введіть n (n<=10): 5

Елементи списку a: [15, 6, 3, 16, 8]

Елементи списку b: [3, 5, 16, 4, 5]

Елементи списку c: [234, 61, 265, 272, 89]

Завдання 2a. Ввести список з указаної користувачем кількості цілих чисел (не більше 10). Елементи списку мають бути введені користувачем. Вивести одержаний список. Визначити максимальне значення елементів списку і його індекс. Якщо елементів з максимальним значенням декілька, в якості результату взяти найбільший індекс.

Приклад:

кількість елементів списку = 7

список:

<i>індекс</i>	0	1	2	3	4	5	6
<i>елемент</i>	3	5	4	5	2	5	0

максимальне значення = 5

індекси елементів з максимальним значенням: 1; 3; 5

найбільший індекс елемента з максимальним значенням = 5

Виконання.

Текст програми може бути таким:

```
#введення списку
```

```
n=int(input('Введіть кількість елементів списку n  
(n<=10):'))
```

```
print('Введіть елементи списку через Enter')
```

```
a=[]
```

```
for i in range(n):
```

```
    x=int(input())
```

```
    a+=[x]
```

```
#пошук максимального елемента
```

```
mm=a[0] #початкове значення максимального елемента
```

```
imm=0 #початкове значення індекса максимального  
елемента
```

```
for i in range(n):
```

```
    if a[i]>=mm: #знак = для пошуку найбільшого індексу
```

```
        mm=a[i] #зміна mm і imm, якщо знайдено
```

```
більший елемент
```

```
        imm=i
```

```
print('Список: ',a)
```

```
print('Максимальний елемент: ',mm)
```

```
print('Найбільший індекс максимального елемента: ',imm)
```

або таким (з використанням стандартної функції для пошуку максимального елемента списку:

```
#введення списку
```

```
n=int(input('Введіть кількість елементів списку n  
(n<=10):'))
```

```
print('Введіть елементи списку через Enter')
```

```
a=[]
```

```
for i in range(n):
```

```
    x=int(input())
```

```
    a+=[x]
```

```
#обчислення максимального елемента за допомогою  
стандартної функції
```

```

mm=max(a)
imm=0      #початкове значення індекса максимального
елемента
for i in range(n):
    if a[i]==mm:      #перебір максимальних значень
        imm=i        #переприсвоєння індекса
print('Список: ',a)
print('Максимальний елемент: ',mm)
print('Найбільший індекс максимального елемента: ',imm)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть кількість елементів списку n (n<=10):5

Введіть елементи списку через Enter

3

4

2

4

4

Список: [3, 4, 2, 4, 4]

Максимальний елемент: 4

Найбільший індекс максимального елемента: 4

Завдання 2b. Змінити програму так, щоб у ній визначалося мінімальне значення елемента списку і його індекс. Якщо елементів з мінімальним значенням декілька, в якості результату взяти найменший індекс.

Використати методи списків.

Завдання 2с. Ввести список з указаної користувачем кількості цілих чисел (не більше 10). Елементи списку мають бути введені користувачем. Визначити максимальне значення серед непарних елементів списку. Підрахувати кількість елементів списку з таким значенням.

Виконання.

Текст програми може бути таким:

```

import math
#введення списку
n=int(input('Введіть кількість елементів списку n
(n<=10):'))
print('Введіть елементи списку через Enter')
a=[]
for i in range(n):
    x=int(input())
    a+= [x]
#пошук максимального елемента

```

```

#початкове значення змінної mm - найменше можливе
число,
#яке при порівнянні буде перевищене будь-яким непарним
елементом списку
mm=-math.inf
for i in range(n):
    if a[i]%2==1 and a[i]>=mm:
        mm=a[i]
#пошук у списку кількості елементів зі значенням mm
k=0
for i in range(n):
    if a[i]==mm:
        k+=1
#або з використанням стандартної функції
#k=a.count(mm)
print('Список: ',a)
print('Максимальний непарний елемент: ',mm)
print('Кількість таких елементів: ',k)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть кількість елементів списку n ($n \leq 10$):6

Введіть елементи списку через Enter

3
2
1
3
3
1

Список: [3, 2, 1, 3, 3, 1]

Максимальний непарний елемент: 3

Кількість таких елементів: 3

Завдання 2d. Змінити програму так, щоб у ній список утворювався з псевдовипадкових чисел від 1 до 100 і визначалося мінімальне значення серед елементів списку, кратних трьом.

Завдання 3a. Ввести список з указаної користувачем кількості цілих чисел (не більше 10). Елементи списку мають бути введені користувачем. Ввести ціле число k. Визначити, чи містяться у списку елементи зі значенням k. Якщо містяться, то вивести кількість таких елементів та їх індекси. Якщо у списку немає елементів з таким значенням, вивести повідомлення про їх нульову кількість.

Виконання.

Текст програми може бути таким:

```
#введення списку
n=int(input('Введіть кількість елементів списку n
(n<=10):'))
print('Введіть елементи списку через Enter')
a=[]
for i in range(n):
    x=int(input())
    a+=[x]
k=int(input('Введіть ціле число k: '))
#Використання функції обчислення кількості входжень
елемента в список
v=a.count(k)
print('Список: ',a)
print('Кількість входжень ',k,'у список:',v)
if v>0:
    for i in range(n):
        if a[i]==k:
            print('значення ', k, ' має індекс',i)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть кількість елементів списку n (n<=10):6

Введіть елементи списку через Enter

1
2
3
1
2
5

Введіть ціле число k: 2

Список: [1, 2, 3, 1, 2, 5]

Кількість входжень 2 у список: 2

значення 2 має індекс 1

значення 2 має індекс 4

>>>

Введіть кількість елементів списку n (n<=10):4

Введіть елементи списку через Enter

3
2
4
5

Введіть ціле число k: 1

Список: [3, 2, 4, 5]

Кількість входжень 1 у список: 0

Завдання 3b. Змінити програму так, щоб у ній визначалась кількість елементів списку, більших за введене користувачем число.

Завдання 3c. Змінити програму так, щоб у ній визначалось середнє арифметичне елементів списку, менших за введене користувачем число.

Завдання 4a. Ввести список значень середньомісячної температури повітря в певному місті в період з місяця з номером m1 по місяць з номером m2 ($1 \leq m1 < m2 \leq 12$). Відобразити дані у вигляді двох стовпців: у першому стовпці містяться номери місяців, а у другому – значення температур. Визначити кількість місяців у цей період, коли середньомісячна температура була нижче за нуль.

Виконання.

Текст програми може бути таким:

```
print('Введіть номери місяців m1,m2 (1<=m1<m2<=12): ')
m1=int(input('m1 = '))
m2=int(input('m2 = '))
print('Введіть середньомісячну температуру:')
a=[]
for i in range(m1,m2+1):
    x=int(input('a['+str(i)+']='))
    a+=[x]
c=0
for m in a:
    if m<0:
        c+=1
print('Місяць ', 'Температура', sep='\t')
for i in range(m1,m2+1):
    print(i, a[i-m1], sep='\t')
print('Кількість місяців з темп. нижче нуля =', c)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть номери місяців m1,m2($1 \leq m1 < m2 \leq 12$):

m1 = 2

m2 = 5

Введіть середньомісячну температуру:

a[2]=-2

a[3]=-1

a[4]=3

a[5]=6

Місяць	Температура
2	-2
3	-1
4	3
5	6

Кількість місяців з темп. нижче нуля = 2

Завдання 4b. Змінити програму так, щоб у ній обчислювалась різниця між максимальним і мінімальним значеннями середньомісячної температури за вказаний період.

Завдання 5a. Сформувати список з уведеної користувачем кількості цілих псевдовипадкових чисел (кількість елементів списку не більше 10, значення елементів не перевищують 100). Вивести одержаний список. Упорядкувати елементи списку за зростанням і за спаданням, використавши стандартні методи.

Виконання.

Текст програми може бути таким:

```
import random
n=int(input('Введіть n (n<=10): '))
#генерування списку з n псевдовипадкових чисел
a=[random.randint(1, 100) for i in range(n)]
print('Список: ',a)
a.sort()
print('Список, упорядкований за зростанням: ',a)
a.sort(reverse=True)
print('Список, упорядкований за спаданням: ',a)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть n (n<=10): 10

Список: [67, 60, 45, 66, 49, 56, 41, 81, 9, 89]

Список, упорядкований за зростанням: [9, 41, 45, 49, 56, 60, 66, 67, 81, 89]

Список, упорядкований за спаданням: [89, 81, 67, 66, 60, 56, 49, 45, 41, 9]

Завдання 6a. Дано x, y, z, – списки розміру n, складені з псевдовипадкових чисел від -20 до 20. В тому зі списків x, y і z, де найбільша кількість від’ємних елементів, всі його додатні елементи замінити на їх куби, якщо це список x або список z, і на їх обернені значення, якщо це список y. Якщо найбільша кількість від’ємних елементів виявиться у кількох списків, розглянути останній з них.

Приклад:

розмір списків = 7

задані списки:

X	-2	2	-1	-2	-3	-2	3
Y	3	-5	4	5	2	5	0
Z	-2	4	1	-3	0	-2	4

кількість від'ємних елементів у списку X = 5

кількість від'ємних елементів у списку Y = 1

кількість від'ємних елементів у списку Z = 3

найбільша кількість від'ємних елементів у списку X, отже, список X буде змінено так:

X	-2	8	-1	-2	-3	-2	27
---	----	---	----	----	----	----	----

списки Y і Z залишаться без змін.

Виконання.

Текст програми може бути таким:

```
import random
n=int(input('Введіть n (n<=10): '))
#генерування списків з n псевдовипадкових чисел
x=[random.randint(-20, 20) for i in range(n)]
print('Список x: ',x)
y=[random.randint(-20, 20) for i in range(n)]
print('Список y: ',y)
z=[random.randint(-20, 20) for i in range(n)]
print('Список z: ',z)
#створення списку з кількостей від'ємних елементів
кожного вектора
k=[]
vx=0; vy=0; vz=0
for i in range(n):
    if x[i]<0:
        vx+=1
k+=vx]
for i in range(n):
    if y[i]<0:
        vy+=1
k+=vy]
for i in range(n):
    if z[i]<0:
        vz+=1
k+=vz]
print('Кількості від_ємних елементів у списках:',k)
#пошук максимального елемента і його індекса у списку k
mm=max(k)
imm=0
for i in range(3):
    if k[i]==mm:
        imm=i
```

```

print('Максимальна кількість від_емних елементів у
списках:',mm)
print('Шуканий номер списку:',imm)
#перетворення списків
#перетворення першого списку (x)
if imm==0:
    x1=x.copy() #створення копії списку x
    x=[] #підготовка списку x до оновлення
    for el in x1:
        if el>0:
            el=el**3
            x+=[el]
#перетворення другого списку (y)
if imm==1:
    y1=y.copy()
    y=[]
    for el in y1:
        if el>0:
            el=1/el
            y+=[el]
#перетворення третього списку (z)
if imm==2:
    z1=z.copy()
    z=[]
    for el in z1:
        if el>0:
            el=el**3
            z+=[el]
print('Перетворені списки:')
print('Список x: ',x)
print('Список y: ',y)
print('Список z: ',z)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть n ($n \leq 10$): 7

Список x: [-18, -20, -20, 17, -19, -10, 0]

Список y: [20, -4, -9, -1, 1, -17, -5]

Список z: [-4, -12, 12, -6, -7, 4, -16]

Кількості від_емних елементів у списках: [5, 5, 5]

Максимальна кількість від_емних елементів у списках: 5

Шуканий номер списку: 2

Перетворені списки:

Список x: [-18, -20, -20, 17, -19, -10, 0]

Список y: [20, -4, -9, -1, 1, -17, -5]
Список z: [-4, -12, 1728, -6, -7, 64, -16]

Завдання 7а. Ввести цілочисельну квадратну матрицю розмірності 4. Введення здійснити поелементно, по рядках. Вивести матрицю у вигляді таблиці. Обчислити слід матриці – суму елементів її головної діагоналі.

Виконання.

Текст програми може бути таким:

```
print ('Введіть елементи матриці 4 x 4:')
a=[]
for i in range(4):
    b=[]
    for j in range(4):
        x=int(input('a ['+str(i)+']['+str(j)+']='))
        b+= [x]
    a+= [b]
print ('Матриця:')
for row in a:
    for el in row:
        print (el,end='\t')
    print ()
t=0
for i in range(4):
    for j in range(4):
        if i==j:
            t=t+a[i][j]
# інший спосіб обчислення сліду
#for i in range(4):
#    t=t+a[i][i]
print ('Слід матриці = ',t)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть елементи матриці 4 x 4:

```
a[0][0]=1
a[0][1]=0
a[0][2]=0
a[0][3]=0
a[1][0]=1
a[1][1]=2
a[1][2]=0
a[1][3]=0
a[2][0]=1
a[2][1]=2
a[2][2]=3
```

```
a[2][3]=0
a[3][0]=1
a[3][1]=2
a[3][2]=3
a[3][3]=4
```

Матриця:

```
1    0    0
1    2    0    0
1    2    3    0
1    2    3    4
```

Слід матриці = 10

Завдання 7b. Змінити програму так, щоб у ній розмірність матриці вводилась користувачем.

Завдання 7с. Ввести цілочисельну матрицю розміром $m \times n$ ($m < 10$, $n < 10$) з псевдовипадкових чисел від 1 до 50. Знайти максимальний елемент матриці на периметрі матриці та його індекси.

Виконання.

Текст програми може бути таким:

```
import random
m=int(input('Введіть кількість рядків матриці m
(m<=10): '))
n=int(input('Введіть кількість стовпців матриці n
(n<=10): '))
a=[]
for i in range(m):
    b=[]
    for j in range(n):
        x=random.randint(1,50)
        b.append(x)
    a.append(b)
print('Матриця: ')
for row in a:
    for el in row:
        print(el,end='\t')
    print()
mm=a[1][1]
imm=1
jmm=1
for i in range(m):
    for j in range(n):
        if (i==0 or i==m-1 or j==0 or j==n-1) and
(a[i][j]>mm):
```

```

mm=a [ i ] [ j ]
imm=i
jmm=j

print ('Максимальний елемент на периметрі матриці =
', mm, ' [ ', imm, ' , ', jmm, ' ] ')

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть кількість рядків матриці m ($m \leq 10$): 3
Введіть кількість стовпців матриці n ($n \leq 10$): 6

Матриця:

15	39	32	27	3	16
29	1	22	43	37	46
42	45	14	2	38	26

Максимальний елемент на периметрі матриці = 46 [1 , 5]

Завдання 7d. Змінити програму так, щоб у ній обчислювався мінімальний серед парних елементів на периметрі матриці і його індекси.

Завдання 7e. Сформувати і вивести одиничну матрицю розмірності n ($n < 10$).

Виконання.

Текст програми може бути таким:

```

n=int(input('Введіть розмірність матриці n (n<=10): '))
a=[]
for i in range(n):
    b=[]
    for j in range(n):
        if i==j:
            x=1
        else:
            x=0
        b.append(x)
    a.append(b)
print('Матриця:')
for row in a:
    for el in row:
        print(el,end=' ')
    print()

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть розмірність матриці n ($n \leq 10$): 5

Матриця:

```
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

Завдання 7f. Дано ціле число n ($n \leq 10$). Сформувати і вивести квадратну матрицю розмірності n , для якої $a_{ij}=0$ при $i=j$ і $a_{ij}=i+2j$ у протилежному випадку.

Виконання.

Текст програми може бути таким:

```
n=int(input('Введіть розмірність матриці n (n<=10): '))
a=[]
for i in range(n):
    b=[]
    for j in range(n):
        if i==j:
            x=0
        else:
            x=i+2*j
        b.append(x)
    a.append(b)
print('Матриця: ')
for row in a:
    for el in row:
        print(el, end=' ')
    print()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть розмірність матриці n ($n \leq 10$): 4

Матриця:

```
0 2 4 6
1 0 5 7
2 4 0 8
3 5 7 0
```

Завдання 7g. Дано ціле число n ($n \leq 15$). Сформувати і вивести квадратну матрицю розмірності n , елементи матриці ввести за допомогою генератора псевдовипадкових чисел (значення елементів від -10 до 10). Обчислити суму елементів матриці на побічній діагоналі і над нею.

Виконання.

Текст програми може бути таким:

```
import random
n=int(input('Введіть розмірність матриці n (n<=10): '))
a=[]
for i in range(n):
    b=[]
    for j in range(n):
        x=random.randint(-10,10)
        b.append(x)
    a.append(b)
print('Матриця: ')
for row in a:
    for el in row:
        print(el,end='\t')
    print()
s=0
for i in range(n):
    for j in range(n):
        if i+j<=n-1:
            s+=a[i][j]

print('Сума елементів на побічній діагоналі і над нею =
',s)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть розмірність матриці n (n<=10): 4

Матриця:

-5	1	9	3
-7	4	6	1
0	3	3	-7
5	-9	-3	3

Сума елементів на побічній діагоналі і над нею = 19

Завдання 7h. Змінити програму так, щоб у ній обчислювалась сума елементів під головною діагоналлю.

Завдання 8a. Сформувати і вивести цілочисельну матрицю розміром m×n (m<10, n<10) з псевдовипадкових чисел від 1 до 20. Визначити максимальний елемент матриці і номер рядка, в якому він знаходиться. Елементи цього рядка відсортувати за зростанням. Вивести перетворену матрицю.

Виконання.

Текст програми може бути таким:

```
import random
m=int(input('Введіть кількість рядків матриці m
(m<=10): '))
n=int(input('Введіть кількість стовпців матриці n
(n<=10): '))
a=[]
for i in range(m):
    b=[]
    for j in range(n):
        x=random.randint(1,20)
        b.append(x)
    a.append(b)
print('Матриця: ')
for row in a:
    for el in row:
        print(el,end='\t')
    print()
mm=a[1][1]
imm=1
for i in range(m):
    for j in range(n):
        if a[i][j]>mm:
            mm=a[i][j]
            imm=i
print('Максимальний елемент матриці = ',mm)
a[imm].sort()
print('Перетворена матриця: ')
for row in a:
    for el in row:
        print(el,end='\t')
    print()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть кількість рядків матриці m (m<=10): 4

Введіть кількість стовпців матриці n (n<=10): 3

Матриця:

19	18	20
14	16	1
11	2	17
6	17	11

Максимальний елемент матриці = 20

Перетворена матриця:

18	19	20
14	16	1
11	2	17
6	17	11

Завдання 8b. Сформувати і вивести цілочисельну матрицю розміром $m \times n$ ($m < 10$, $n < 10$) з псевдовипадкових чисел від 0 до 2. Відсортувати за зростанням один зі стовпців цієї матриці, в якому передостанній елемент дорівнює нулю.

Зауваження. В даній задачі доцільно задати матрицю, як список стовпців. Оскільки традиційно матрицю виводять на екран порядково, доведеться дещо змінити фрагменти програми, що відповідають за друк матриці

Виконання.

Текст програми може бути таким:

```
import random
m=int(input('Введіть кількість рядків матриці m
(m<=10): '))
n=int(input('Введіть кількість стовпців матриці n
(n<=10): '))
a=[]
for i in range(n):
    b=[]
    for j in range(m):
        x=random.randint(0,2)
        b.append(x)
    a.append(b)
print('Матриця: ')
for i in range(m):
    for j in range(n):
        print(a[j][i],end='\t')
    print()
k=-1 #пошук стовпців, де передостанній елемент дорівнює
нулю
for j in range(n):
    if a[j][m-2]==0:
        k=j
if k==-1:
    print('немає стовпців для сортування')
    exit(0) #вихід з програми
else:
    print('Стовпець матриці для сортування k=',k)
    a[k].sort(reverse=False)
```

```
print('Змінена матриця: ')

for i in range(m):
    for j in range(n):
        print(a[j][i],end='\t')
    print()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть кількість рядків матриці m ($m \leq 10$): 7

Введіть кількість стовпців матриці n ($n \leq 10$): 9

Матриця:

```
2   1   2   1   0   1   2   1   2
1   1   1   2   0   0   0   1   0
2   2   0   2   1   2   1   1   2
1   2   2   2   1   2   2   0   0
2   1   0   1   2   0   1   2   0
2   1   0   2   0   2   2   0   2
2   0   0   0   0   2   1   0   2
```

Стовпець матриці для сортування k= 7

Змінена матриця:

```
2   1   2   1   0   1   2   0   2
1   1   1   2   0   0   0   0   0
2   2   0   2   1   2   1   0   2
1   2   2   2   1   2   2   1   0
2   1   0   1   2   0   1   1   0
2   1   0   2   0   2   2   1   2
2   0   0   0   0   2   1   2   2
```

Завдання 9а. Сформувати і вивести цілочисельну матрицю розміром m x n ($m \leq 10$, $n \leq 10$) з псевдовипадкових чисел від 0 до 2. В цій матриці визначити рядки, в яких елементи упорядковані за зростанням, знайти суму першого і останнього елементів у кожному з таких рядків і замінити мінімальною з таких сум елементи периметру заданої матриці.

Виконання.

Текст програми може бути таким:

```
import random
m=int(input('Введіть кількість рядків матриці m
(m<=10): '))
n=int(input('Введіть кількість стовпців матриці n
(n<=10): '))
a=[]
for i in range(m):
```

```

b=[]
for j in range(n):
    x=random.randint(0,2)
    b.append(x)
a.append(b)
print('Матриця: ')
for row in a:
    for el in row:
        print(el,end='\t')
    print()
#формування списку, кількість ел-тів якого дорівнює
кількості рядків матриці
#якщо рядок упорядкований, ел-т = 1, інакше = 0
pp=[]
for i in range(m):
    p=1
    for j in range(n-1):
        for k in range(j+1,n):
            if a[i][j]>a[i][k]:
                p=0
    pp.append(p)
print('pp=',pp)
#форм. списку, ел-ти якого - суми першого і останнього
ел-тів упорядкованих рядків матриці
s=[]
for i in range(m):
    if pp[i]==1:
        sum=a[i][0]+a[i][n-1]
        s.append(sum)
print('s=',s)
#перетворення матриці у випадку непорожнього списку s
if s!=[]:
    mm=min(s)
    print('min(s)=',mm)
    for i in range(m):
        for j in range(n):
            if (i==0 or i==m-1 or j==0 or j==n-1):
                a[i][j]=mm
    print('Змінена матриця: ')
    for row in a:
        for el in row:
            print(el,end='\t')
        print()
else:
    print('шуканих рядків немає')

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Введіть кількість рядків матриці m ($m \leq 10$): 5

Введіть кількість стовпців матриці n ($n \leq 10$): 3

Матриця:

```
1      0      2
1      0      1
0      2      1
2      2      1
2      0      1
```

pp= [0, 0, 0, 0, 0]

s= []

шуканих рядків немає

>>>

Введіть кількість рядків матриці m ($m \leq 10$): 4

Введіть кількість стовпців матриці n ($n \leq 10$): 3

Матриця:

```
1      2      1
1      1      1
1      1      2
2      1      1
```

pp= [0, 1, 1, 0]

s= [2, 3]

min(s)= 2

Змінена матриця:

```
2      2      2
2      1      2
2      1      2
2      2      2
```

Завдання 9б. Сформувати і вивести цілочисельну матрицю порядку n ($n \leq 10$) з псевдовипадкових чисел від 1 до 2. В цій матриці визначити рядки, які складаються з однакових елементів, і всі елементи таких рядків замінити нулями.

2.2. Лабораторна робота №4. Списки

Зауваження:

1. Інтерфейс програми має бути зрозумілим.
2. При введенні некоректних вхідних даних чи неможливості проведення обчислень користувачу має виводитися зрозуміле йому повідомлення про помилку українською мовою, і програма має бути стійкою при таких колізіях.
3. Кожне завдання має бути розв'язане в межах окремої програми.
4. При розв'язуванні задач, окрім першої задачі, передбачити введення розмірності матриць.
5. При розв'язуванні другої задачі передбачити виведення отриманої матриці.
6. При розв'язуванні третьої і четвертої задачі введення елементів матриці має виконувати користувач.
7. При розв'язуванні третьої задачі передбачити наявність на екрані заданої та результуючої матриць.
8. При розв'язуванні четвертої задачі передбачити наявність на екрані заданої та перетвореної матриць.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Розробити програму для введення значень щогодинних спостережень температури повітря від K до M години ($0 \leq K < M \leq 23$) та знаходження мінімальної температури повітря за цей час.
2. Створити програму для генерування матриці
$$\frac{17f(0.25)+2f(1+y)}{6-f(y^2-1)}$$
 ($3 \leq N \leq 15$), елементи якої задаються формулою
$$h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2}$$
 та знаходження кількості елементів, що знаходяться на головній діагоналі цієї матриці та є більшими за її останній елемент.
3. Створити програму для введення елементів дійсної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). Елементи передостаннього рядка цієї матриці відсортувати за спаданням.
4. Розробити програму для знаходження середнього арифметичного у кожному стовпці цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$) і, визначивши мінімальне з них, додати його значення до елементів того стовпця, в якому воно знаходиться.

Варіант 2.

1. Розробити програму для введення значень щомісячної заробітної плати за минулий рік починаючи з K -го місяця ($1 \leq K < 12$) та знаходження номеру місяця з найбільшою заробітною платою.
2. Створити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ та знаходження кількості від'ємних елементів, що знаходяться на побічній діагоналі цієї матриці.
3. Розробити програму для введення елементів дійсної прямокутної матриці розмірності $(2N+1) \times M$ ($1 \leq N \leq 3$, $2 \leq M \leq 10$). Передбачити можливість сортування за зростанням середнього рядка цієї матриці.
4. Створити програму для введення елементів цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). Впорядкувати всі елементи матриці за спаданням, додати до них максимальний елемент матриці та записати одержані елементи порядково у матрицю такої ж розмірності.

Варіант 3.

1. Розробити програму для введення значень розміру стипендії від K -го до M -го року ($2002 \leq K < M \leq 2022$) та знаходження середнього розміру стипендії за цей час.
2. Створити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $S = \frac{4}{3}\pi r^3$ та знаходження кількості таких елементів, що знаходяться над головною діагоналлю цієї матриці, які більші за деяке задане число K .
3. Розробити програму для введення елементів цілочисельної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 15$, $2 \leq M \leq 13$). Передбачити можливість сортування за спаданням одного з рядків цієї матриці, в якому перший елемент більший за останній.
4. Розробити програму для знаходження рядка з мінімальною кількістю додатних елементів дійсної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$) і подальшого віднімання цієї мінімальної кількості додатних елементів від елементів усіх інших рядків.

Варіант 4.

1. Створити програму для введення значень середньої вартості персональних комп'ютерів з K по M рік ($1980 \leq K < M \leq 2022$) та знаходження такого року, коли їх вартість була мінімальною.
2. Розробити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ та підрахунку кількості додатних елементів без урахування елементів останнього стовпця та першого рядка цієї матриці.
3. Розробити програму для введення елементів дійсної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). Елементи другого стовпця цієї матриці відсортувати за зростанням.
4. Розробити програму для знаходження максимального і мінімального елементів дійсної матриці розмірності $N \times N$ ($2 \leq N \leq 15$) і у випадку, якщо максимальний елемент виявиться на головній діагоналі, всі елементи головної діагоналі замінити на мінімальний елемент.

Варіант 5.

1. Розробити програму для введення зведеної відомості результатів іспиту з інформатики (кількість оцінок: A, B, C, D, E, Fx, F) та обчислення відношення кількості студентів до кількості п'ятірок та четвірок (A, B, C).
2. Створити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$ та знаходження парних елементів, що знаходяться над побічною діагоналлю цієї матриці.
3. Розробити програму для введення елементів цілочисельної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 9$, $2 \leq M \leq 9$). Передбачити можливість сортування за зростанням одного зі стовпців цієї матриці, в якому передостанній елемент від'ємний.
4. Створити програму для введення в матрицю дійсних координат N точок на площині ($2 \leq N \leq 10$). Отримати матрицю відстаней між цими точками та знайти таку точку, сума відстаней від якої до інших e мінімальною.

Варіант 6.

1. Створити програму для введення значень вимірювання опору електричного провідника в залежності від його температури, що змінювалась в діапазоні від T_1 до T_2 градусів ($20 \leq T_1 < T_2 \leq 60$) та знаходження такої температури, коли зміна опору була найменшою.
2. Розробити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої – дійсні числа на відрізку $[1; 100]$ та знаходження суми елементів останнього рядка та останнього стовпця цієї матриці.
3. Розробити програму для введення елементів дійсної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). Елементи головної діагоналі цієї матриці відсортувати за зростанням.
4. Розробити програму, в якій для заданої цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$) елементи рядка, в якому знаходиться мінімальний елемент матриці, будуть замінені на максимальний елемент цієї матриці.

Варіант 7.

1. Розробити програму для введення даних про кількість студентів в групах першого курсу та визначення групи з мінімальною кількістю студентів.
2. Створити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $r \neq 1$, та знаходження суми від'ємних елементів під головною діагоналлю цієї матриці.
3. Створити програму для введення в матрицю дійсних координат N точок на площині ($2 \leq N \leq 10$). Елементи матриці відсортувати за зростанням першої координати кожної точки. Відповідність координат всіх точок повинна залишитись без змін.
4. Створити програму для введення елементів цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). В цій матриці визначити рядки, в яких елементи упорядковані за зростанням, знайти суму елементів в кожному з таких рядків та додати максимальну з таких сум до всіх елементів заданої матриці.

Варіант 8.

1. Створити програму для введення значень щогодинних спостережень температури повітря від K до M години ($0 \leq K < M \leq 23$) та підрахунку загальної кількості годин, коли температура була нижча за нуль.
2. Розробити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $f(a,b) = \frac{a}{b^2+1} + \frac{b}{a^2+1}$, та знаходження суми додатних елементів під головною діагоналлю цієї матриці.
3. Розробити програму для введення елементів дійсної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 15$). Передбачити можливість сортування за спаданням вказаного стовпця K цієї матриці ($1 \leq K \leq M$).
4. Створити програму для введення елементів цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). В цій матриці визначити стовпчики, в яких елементи розташовані за спаданням, знайти максимальний елемент в кожному з таких стовпчиків та відняти мінімальний з них від усіх елементів заданої матриці.

Варіант 9.

1. Створити програму для введення значень щогодинних спостережень температури повітря від K до M години ($0 \leq K < M \leq 23$) та підрахунку загальної кількості разів, коли температура знижувалась (наступне значення було менше за попереднє).
2. Розробити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $t(x,y) = y^3 + xy + 32$ та знаходження суми не додатних елементів під побічною діагоналлю цієї матриці.
3. Розробити програму для введення елементів цілочисельної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 15$, $2 \leq M \leq 13$). Передбачити можливість сортування за спаданням одного з рядків цієї матриці, в якому перший елемент більший за останній.
4. Розробити програму для знаходження стовпця з максимальною кількістю нульових елементів цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$) і подальшого віднімання елементів цього стовпця від відповідних елементів усіх інших стовпців.

Варіант 10.

1. Створити програму для введення значень щорічного вимірювання зросту людини від K до M років ($0 \leq K < M \leq 20$) та знаходження такого року, коли збільшення зросту було максимальним.
2. Розробити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої – дійсні числа на відрізку $[0; 1]$ та знаходження суми елементів, менших за 0.5, що знаходяться над головною діагоналлю цієї матриці.
3. Розробити програму для введення елементів дійсної прямокутної матриці розмірності $N \times (2M+1)$ ($2 \leq N \leq 10$, $1 \leq M \leq 5$). Передбачити можливість сортування за зростанням середнього стовпця цієї матриці.
4. Створити програму для введення елементів цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). Впорядкувати всі елементи матриці за зростанням, відняти від кожного з них мінімальний елемент матриці та записати одержані елементи за стовпцями в матрицю такої ж розмірності.

Варіант 11.

1. Створити програму для введення значень вимірювання опору електричного провідника в залежності від його температури, що змінювалась в діапазоні від T_1 до T_2 градусів ($20 \leq T_1 < T_2 \leq 60$), та знаходження такої температури, коли зміна опору була найменшою .
2. Створити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $f(x) = a_5 x^5 + a_4 x^4 + \dots + a_1 x + a_0$, та знаходження кількості від'ємних елементів на побічній діагоналі цієї матриці .
3. Розробити програму для введення елементів цілочисельної прямокутної матриці розмірності $N \times M$ ($2 \leq N \leq 9$, $2 \leq M \leq 9$). Передбачити можливість сортування за зростанням одного з стовпців цієї матриці, в якому передостанній елемент від'ємний.
4. Створити програму для введення елементів цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). В цій матриці визначити стовпці, в яких елементи упорядковані за спаданням, знайти максимальний елемент в кожному з таких стовпців та відняти мінімальний з них від усіх елементів заданої матриці .

Варіант 12.

1. Розробити програму для введення значень щомісячної заробітної плати за минулий рік, починаючи з K -го місяця ($1 \leq K < 12$) та знаходження номеру місяця з найбільшою заробітною платою.
2. Створити програму для генерування матриці $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$ ($2 \leq N \leq 10$), елементи якої задаються формулою $g(a,b) = \frac{a^2+b^2}{a^2+b^2+4}$, та знаходження парних елементів, що знаходяться над побічною діагоналлю цієї матриці .
3. Розробити програму для введення елементів дійсної прямокутної матриці розмірності $(2N+1) \times M$ ($1 \leq N \leq 3$, $2 \leq M \leq 10$). Передбачити можливість сортування за зростанням середнього рядка цієї матриці .
4. Розробити програму, в якій для заданої цілочисельної матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$) елементи рядка, в якому знаходиться мінімальний елемент матриці, будуть замінені на максимальний елемент цієї матриці.

2.3. Функції користувача

Питання для обговорення.

1. Опис та виклик функцій.
2. Розширене використання параметрів та аргументів.
3. Глобальні та локальні змінні.
4. Правила локалізації.
5. Lambda функції.
6. Рекурсія.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Обчислення значення виразу $f(t, -2s, 1.17) + f(2.2, t, s-t)$, де $f(a, b, c) = (2a - b \cdot \sin(c)) / (5 + c)$. Використати підпрограму обчислення значення f . Вхідні аргументи s, t є дійсними числами.

Виконання.

Текст програми може бути таким:

```
import math
def f(a, b, c):
    return (2*a-b-math.sin(c))/(5+c)
print('Введіть дійсні числа s, t: ')
s=float(input('s='))
t=float(input('t='))
r=f(t, -2*s, 1.17)+f(2.2, t, s-t)
print('Результат = ', r)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

s= 0 t=0 r= 0.7307697572550835
s= 3 t=3 r= 2.0756644087947915

Завдання 1b. Розробити програму для обчислення значення виразу $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$, де $\text{mymin}(x,y)$ – функція, що повертає мінімальне з двох чисел x та y . Функцію $\text{mymin}(x,y)$ описати окремою підпрограмою. Вхідні дані є дійсними числами.

Виконання.

Текст програми може бути таким:

```
import math
def mymin(x, y):
    if x<=y:
        m=x
    else:
        m=y
    return m

print('Введіть дійсні числа a, b: ')
a=float(input('a='))
b=float(input('b='))
r=mymin(a,mymin(a+b,a*b))/(mymin(a*a,b*b)+mymin(math.sqrt(abs(a)),math.sqrt(abs(b))))
print('Результат = ', r)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

a= 1 b=2 r=0.50
a=2 b=1 r=1.00

Завдання 1с. Змінити програму так, щоб у ній використовувалась функція $\text{mymax}(x,y)$, що повертає максимальне з двох чисел x та y .

Завдання 2a. Сформувати і вивести квадратну матрицю $A_{N \times N}$, ($2 \leq N \leq 10$), елементи якої задаються за формулою $A_{ij}=2\sin(i)-3\cos(j)$. Обчислити суму елементів головної діагоналі матриці та кількість від'ємних елементів побічної діагоналі матриці. Створити і використати підпрограми: генерування матриці, опрацювання матриці, виведення матриці.

Виконання.

Текст програми може бути таким:

```
import math as m
```

```

def input_matr(n):
    a=[]
    for i in range(n):
        b=[]
        for j in range(n):
            x=2*m.sin(i)-3*m.cos(j)
            b.append(x)
        a.append(b)
    return a

def print_matr(a):
    print('Матриця:')
    for row in a:
        for el in row:
            print('{:.3}'.format(el),end='\t')
        print()

def my_sum(a):
    s=0
    for i in range(n):
        s+=a[i][i]
    return s

def my_count(a):
    c=0
    for i in range(n):
        for j in range(n):
            if (i+j==n-1) and (a[i][j]<0):
                c+=1
    return(c)

n=int(input('Введіть розмірність матриці n (n<=10): '))
a=input_matr(n)
print_matr(a)
print('Сума ел. гол. діаг.: ',
      '{:.5}'.format(my_sum(a)))
print('Кільк. від'емних. ел. поб. діаг.: ',my_count(a))

```

Ввести програму, запустити її на виконання, перевірити роботу програми на контрольних прикладах:

Введіть розмірність матриці n (n<=10): 1

Матриця:

-3.0

Сума ел. гол. діаг.: -3.0

Кільк. від'ємних. ел. поб. діаг.: 1

Введіть розмірність матриці n ($n \leq 10$): 2

Матриця:

-3.0 -1.62

-1.32 0.062

Сума ел. гол. діаг.: -2.938

Кільк. від'ємних. ел. поб. діаг.: 2

Введіть розмірність матриці n ($n \leq 10$): 5

Матриця:

-3.0 -1.62 1.25 2.97 1.96

-1.32 0.062 2.93 4.65 3.64

-1.18 0.198 3.07 4.79 3.78

-2.72 -1.34 1.53 3.25 2.24

-4.51 -3.13 -0.265 1.46 0.447

Сума ел. гол. діаг.: 3.8286

Кільк. від'ємних. ел. поб. діаг.: 2

Завдання 2b. Змінити програму так, щоб у ній обчислювалась кількість від'ємних елементів на головній діагоналі матриці та під нею.

Завдання 3a. Задано чотири списки з введеної користувачем кількості N ($N \leq 10$) цілих чисел. Для кожного зі списків обчислити кількість додатних непарних елементів. Визначити найбільше, найменше і середнє арифметичне з отриманих значень. Використати підпрограми створення списку, виведення списку, обчислення кількості додатних непарних елементів списку, знаходження максимального, мінімального, середнього арифметичного елементів списку. Для введення даних використати псевдовипадкові числа від -20 до 20.

Виконання.

Текст програми може бути таким:

```
import random
def input_list(n):
    x=[random.randint(-20, 20) for i in range(n)]
    return x

def print_list(name,x):
    print('Список '+name+':',x)

def count_list(x):
    c=0
    for i in range(n):
        if (x[i]>0) and (x[i]%2==1):
```



```

        c+=1
    return (c)

def max_4(x):
    mmax=x[0]
    for i in range(4):
        if x[i]>=mmax:
            mmax=x[i]
    return mmax

def min_4(x):
    mmin=x[0]
    for i in range(4):
        if x[i]<=mmin:
            mmin=x[i]
    return mmin

def avg_4(x):
    s=0
    for i in range(4):
        s+=x[i]
    av=s/4
    return av

n=int(input('Введіть кількість елементів списків n
(n<=10): '))
a1=input_list(n)
a2=input_list(n)
a3=input_list(n)
a4=input_list(n)
print_list('a1',a1)
print_list('a2',a2)
print_list('a3',a3)
print_list('a4',a4)

#формування списку з кількостей додатних непарних ел-
тів списків
cc=[]
cc+=[count_list(a1)]
cc+=[count_list(a2)]
cc+=[count_list(a3)]
cc+=[count_list(a4)]
print('Кількості додатних непарних ел-тів списків:')
print(cc)

```

```
print ('Максимум: ', max_4(cc))
print ('Мінімум: ', min_4(cc))
print ('Середнє арифметичне: ', avg_4(cc))
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклади виконання:

Введіть кількість елементів списків n ($n \leq 10$): 5

Список a1: [17, 1, -18, 10, 18]

Список a2: [-9, 4, -19, -11, -11]

Список a3: [-1, -14, 17, -15, -14]

Список a4: [15, -14, 4, -2, -17]

Кількості додатних непарних ел-тів списків:

[2, 0, 1, 1]

Максимум: 2

Мінімум: 0

Середнє арифметичне: 1.0

Введіть кількість елементів списків n ($n \leq 10$): 8

Список a1: [4, -19, -1, -17, 4, 19, 15, -13]

Список a2: [16, 1, 17, -8, -8, 19, -6, 0]

Список a3: [-6, -17, 7, 19, 4, -16, 2, 20]

Список a4: [14, -20, -11, 7, 11, 4, -5, 19]

Кількості додатних непарних ел-тів списків:

[2, 3, 2, 3]

Максимум: 3

Мінімум: 2

Середнє арифметичне: 2.5

Завдання 3b. Змінити програму так, щоб у ній обчислювалась сума парних елементів кожного списку.

Завдання 4a. Задано три цілочисельні матриці розмірності $N \times N$ ($N \leq 10$).

Для кожної матриці визначити суму додатних елементів побічної діагоналі. Визначити найбільше з обчислених значень. Використати підпрограми. Для введення даних використати псевдовипадкові числа від -50 до 50.

Виконання.

Текст програми може бути таким:

```
import random
def input_matr(n):
    a=[]
    for i in range(n):
        b=[]
```

```

        for j in range(n):
            x=random.randint(-50,50)
            b.append(x)
        a.append(b)
    return a

def print_matr(name,a):
    print('Матриця '+name+':')
    for row in a:
        for el in row:
            print(el,end='\t')
        print()

def count_matr(a):
    s=0
    for i in range(n):
        for j in range(n):
            if (i+j==n-1) and (a[i][j]>0):
                s+=a[i][j]
    return s

def max_3(x):
    mmax=x[0]
    for i in range(3):
        if x[i]>=mmax:
            mmax=x[i]
    return mmax

n=int(input('Введіть розмірність матриці n (n<=10): '))
m1=input_matr(n)
print_matr('m1',m1)
m2=input_matr(n)
print_matr('m2',m2)
m3=input_matr(n)
print_matr('m3',m3)

v=[]
v+=[count_matr(m1)]
v+=[count_matr(m2)]
v+=[count_matr(m3)]

print('Суми додатних елементів побічних діагоналей:')
print(v)

print('Максимум: ',max_3(v))

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклади виконання:

Введіть розмірність матриці n ($n \leq 10$): 2

Матриця $m1$:

-25 36

-9 3

Матриця $m2$:

-37 -15

29 33

Матриця $m3$:

4 46

-11 21

Суми додатних елементів побічних діагоналей:

[36, 29, 46]

Максимум: 46

Введіть розмірність матриці n ($n \leq 10$): 3

Матриця $m1$:

45 0 -14

-11 9 43

-43 25 33

Матриця $m2$:

2 -18 8

1 -49 45

-42 -12 -9

Матриця $m3$:

-17 -14 -28

-40 -25 -24

-7 -2 -42

Суми додатних елементів побічних діагоналей:

[9, 8, 0]

Максимум: 9

Завдання 4b. Змінити програму так, щоб у ній обчислювалась кількість від'ємних елементів головної діагоналі кожної матриці.

Завдання 5a. Задано три матриці розмірності $M \times N$ ($2 \leq M \leq 10$, $2 \leq N \leq 10$). Розробити програму для знаходження тієї з цих матриць, що має найбільшу кількість стовпчиків, всі елементи яких додатні. Використати підпрограми. Для введення даних використати псевдовипадкові числа від -10 до 50.

Зауваження. В даній задачі задамо матрицю, як список стовпців. Оскільки традиційно матрицю виводять на екран порядково, децю змінимо фрагменти програми, що відповідають за друк матриці.

Виконання.

Текст програми може бути таким:

```
import random
def input_matr(m,n):
    a=[]
    for i in range(n):
        b=[]
        for j in range(m):
            x=random.randint(-10,50)
            b.append(x)
        a.append(b)
    return a

def print_matr(name,a):
    print('Матриця '+name+':')
    for i in range(m):
        for j in range(n):
            print(a[j][i],end='\t')
        print()
    print()

def count_matr(a):
    k=0
    for j in range(n):
        t=0
        for i in range(m):
            if a[j][i]>0:
                t+=1
        if t==m:
            k+=1
    return k

def imax_3(x):
    mmax=x[0]
    imax=0
    for i in range(3):
        if x[i]>=mmax:
            mmax=x[i]
            imax=i
    return imax
```

```

m=int(input('Введіть кількість рядків матриці m(
m<=10): '))
n=int(input('Введіть кількість стовпців матриці n
(n<=10): '))
m1=input_matr(m,n)
print_matr('m1',m1)
m2=input_matr(m,n)
print_matr('m2',m2)
m3=input_matr(m,n)
print_matr('m3',m3)

v=[]
v+=[count_matr(m1)]
v+=[count_matr(m2)]
v+=[count_matr(m3)]

print('Кількості стовпців з усіма додатними ел-тами:')
print(v)
print('Номер максимуму: ',imax_3(v))

print('Шукана матриця - це:')
if imax_3(v)==0:
    print_matr('m1',m1)
elif imax_3(v)==1:
    print_matr('m2',m2)
else:
    print_matr('m3',m3)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклади виконання:

Введіть кількість рядків матриці m(m<=10): 2

Введіть кількість стовпців матриці n (n<=10): 4

Матриця m1:

-1	33	27	17
43	6	26	36

Матриця m2:

12	48	15	27
9	2	8	25

Матриця m3:

35	-7	-9	23
38	14	-8	-4

Кількості стовпців з усіма додатними ел-тами:

[3, 4, 1]

Номер максимуму: 1

Шукана матриця - це:

Матриця m2:

12	48	15	27
----	----	----	----

9	2	8	25
---	---	---	----

Введіть кількість рядків матриці m ($m \leq 10$): 4

Введіть кількість стовпців матриці n ($n \leq 10$): 6

Матриця m1:

10	10	23	43	4	47
----	----	----	----	---	----

36	47	3	36	47	-8
----	----	---	----	----	----

31	40	23	32	-10	49
----	----	----	----	-----	----

1	41	17	50	17	-1
---	----	----	----	----	----

Матриця m2:

36	42	0	47	-1	47
----	----	---	----	----	----

50	45	24	31	29	-4
----	----	----	----	----	----

30	8	17	23	-4	14
----	---	----	----	----	----

43	-1	-2	27	44	46
----	----	----	----	----	----

Матриця m3:

37	23	25	7	1	28
----	----	----	---	---	----

37	-9	18	3	22	-2
----	----	----	---	----	----

-5	38	45	37	22	-3
----	----	----	----	----	----

1	22	18	7	15	37
---	----	----	---	----	----

Кількості стовпців з усіма додатними ел-тами:

[4, 2, 3]

Номер максимуму: 0

Шукана матриця - це:

Матриця m1:

10	10	23	43	4	47
----	----	----	----	---	----

36	47	3	36	47	-8
----	----	---	----	----	----

31	40	23	32	-10	49
----	----	----	----	-----	----

1	41	17	50	17	-1
---	----	----	----	----	----

Завдання 5b. Змінити програму так, щоб у ній визначалась матриця з найбільшою кількістю від'ємних елементів.

Завдання 5c. Змінити програму так, щоб у ній визначалась матриця з найбільшою кількістю рядків, де перший і останній елементи є парними числами.

Завдання 5d. Змінити програму так, щоб у ній визначалась матриця з найбільшою сумою елементів на периметрі.

2.4. Лабораторна робота №5. Функції

Зауваження:

1. Інтерфейс програми має бути зрозумілим.
2. Кожне завдання має бути розв'язане в межах окремої програми.
3. При розв'язуванні задач повторювані блоки операторів оформити у вигляді підпрограм.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Розробити програму для обчислення значення виразу $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$, де $h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2}$ є окремо описаною підпрограмою, яка визначає максимальне значення елементів вектора a з п'яти елементів. Вхідний аргумент n та елементи векторів y, z є цілими числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Розробити програму для обчислення середнього арифметичного кожного з трьох векторів, заданих своїми координатами у п'ятивимірному просторі.
4. Задано три матриці розмірності $N \times N$ ($3 \leq N \leq 10$). Розробити програму для знаходження тієї з цих матриць, що має найбільшу суму елементів на периметрі.

Варіант 2.

1. Розробити програму для обчислення значення виразу $S = h \frac{a+b}{2}$, де $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ є окремо описаною підпрограмою. Вхідний аргумент t є цілим числом.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.

3. Розробити програму для обчислення максимального із визначників трьох матриць третього порядку, використавши підпрограму обчислення визначника матриці другого порядку.
4. Розробити програму для введення K матриць ($2 \leq K \leq 5$) розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$), та виведення тієї з них, що має найменшу кількість нульових елементів рядків, у яких є принаймні два не додатні елементи.

Варіант 3.

1. Розробити програму для обчислення значення виразу $z = \sqrt{a^2 + b^2}$, де $S = \frac{4}{3} \pi r^3$ є окремо описаною підпрограмою. Вхідні дані є цілими числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Розробити програму, в якій для заданих координатами вершин трьох трикутників обчислюються їх площі.
4. Задано три матриці розмірності $N \times N$ ($3 \leq N \leq 15$). Розробити програму для знаходження тієї з цих матриць, в якій найбільша кількість додатних елементів з парною сумою індексів.

Варіант 4.

1. Розробити програму для обчислення значення виразу $S = a^3 \frac{\sqrt{2}}{12}$, де $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ є окремо описаною підпрограмою. Вхідні аргументи $s, a \in [0..3]$ є цілими числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Задані N натуральних чисел ($2 \leq N \leq 5$). Розробити програму, у якій визначається максимальний з найбільших спільних дільників будь-якої пари цих чисел.
4. Розробити програму для введення K матриць ($2 \leq K \leq 5$) розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$) та виведення тієї з них, що має найменшу суму елементів стовпців, в яких є принаймні два додатні елементи.

Варіант 5.

1. Розробити програму для обчислення значення виразу
$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots,$$
 де
$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$
 є окремо описаною підпрограмою. Вхідний аргумент t є дійсним числом.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. N зведених квадратних рівняння ($3 \leq N \leq 6$) задаються своїми коренями x_1, x_2 . Визначити та надрукувати те з них, що має найбільший додатний коефіцієнт при x (вивести рівняння в звичайному математичному вигляді).
4. Розробити програму для введення K матриць ($2 \leq K \leq 5$) розмірності $N \times M$ ($2 \leq N \leq 10, 2 \leq M \leq 10$) та виведення тієї з них, що має найбільшу суму елементів рядків, в яких є нульові елементи.

Варіант 6.

1. Розробити програму для обчислення значення виразу $S_n = n \frac{a_1 + a_n}{2}$, де $\max(x, y)$ – функція, що повертає максимальне з двох чисел x та y . Функцію $\max(x, y)$ описати окремою підпрограмою. Вхідні дані є цілими числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Розробити програму, в якій для заданих коефіцієнтами 3-х квадратних рівнянь визначається те з них, що має найбільший корінь.
4. Задано три матриці розмірності $N \times N$ ($3 \leq N \leq 15$). Розробити програму знаходження тієї з цих матриць, що має найбільшу суму додатних елементів на діагоналях.

Варіант 7.

1. Розробити програму для обчислення значення виразу $S_n = \frac{b_1(r^n - 1)}{r - 1}$, де $r \neq 1$ є окремо описаною підпрограмою. Вхідний аргумент u є дійсним числом.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши

підпрограми: генерування списку, опрацювання списку, виведення списку.

3. Розробити програму, у якій визначається, чи є серед трьох трикутників, заданих координатами вершин, тупокутні, і якщо є, виводиться величина тупого кута в градусах.
4. Розробити програму для введення K матриць ($2 \leq K \leq 5$) розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$), та виведення тієї з них, що має найменшу кількість нулів у стовпці з максимальним елементом.

Варіант 8.

1. Розробити програму для обчислення значення виразу $n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$, де $f(a,b) = \frac{a}{b^2+1} + \frac{b}{a^2+1}$ є окремо описаною підпрограмою. Вхідні дані є цілими числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Розробити програму, в якій для заданих координатами вершин чотирьох трикутників обчислюються довжини їх бісектрис.
4. Задано три матриці розмірності $N \times N$ ($3 \leq N \leq 15$). Розробити програму для визначення матриці з найбільшою сумою не додатних елементів під побічною діагоналлю.

Варіант 9.

1. Розробити програму для обчислення значення виразу $\frac{t(a,b)}{2+t(a^2,b-3)} + t(a,b-3) + f(\pi,0)$, де $t(x,y) = y^3 + xy + 32$ є окремо описаною підпрограмою. Вхідні аргументи a, b є дійсними числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Чотири вектори у N -вимірному просторі ($2 \leq N \leq 5$) задаються своїми координатами. Розробити програму для обчислення скалярного добутку двох векторів з найбільшою та найменшою нормами (норму вектора визначити як суму координат вектора).
4. Розробити програму для введення K матриць ($2 \leq K \leq 5$) розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$), та виведення тієї з них, в якій сума

елементів рядка з максимальним елементом є мінімальною.

Варіант 10.

1. Розробити програму для обчислення значення виразу
$$\frac{\min(a, \min(a+b, ab))}{\min(a^2, b^2) + \min(\sqrt{|a|}, \sqrt{|b|})}$$
, де $\min(x, y)$ – функція, що повертає мінімальне з двох чисел x та y . Функцію $\min(x, y)$ описати окремою підпрограмою. Вхідні дані є цілими числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Розробити програму, в якій для заданих координатами вершин 4-х трикутників визначаються довжини їх висот.
4. Задано три матриці розмірності $N \times N$ ($3 \leq N \leq 15$). Розробити програму для знаходження тієї з цих матриць, що має найменше середнє арифметичне елементів стовпця, в якому міститься максимальний елемент матриці.

Варіант 11.

1. Задано числа a_0, a_1, \dots, a_5 . Розробити програму для обчислення значення виразу $f(p+1) - f(p)$, де $f(x) = a_5 x^5 + a_4 x^4 + \dots + a_1 x + a_0$. Функцію $f(x)$ описати окремою підпрограмою. Вхідні дані є цілими числами.
2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Розробити програму, в якій три вектори, задані в 4-вимірному евклідовому просторі, будуть пронормовані та буде проведена перевірка на ортогональність кожної пари векторів.
4. Задано три матриці розмірності $N \times N$ ($3 \leq N \leq 10$). Розробити програму для знаходження тієї з цих матриць, що має найбільшу кількість від'ємних елементів над побічною діагоналлю.

Варіант 12.

1. Розробити програму для обчислення значення виразу

$g(1.2,s)+g(t,s) - g(2s - 1,st)$, де $g(a,b) = \frac{a^2+b^2}{a^2+b^2+4}$ є окремо описаною підпрограмою. Вхідні аргументи t, s є дійсними числами.

2. Розв'язати другу задачу лабораторної роботи № 4 (Списки), створивши підпрограми: генерування списку, опрацювання списку, виведення списку.
3. Розробити програму, в якій для трьох трикутників, заданих координатами вершин, буде виведена площа того трикутника, який має найбільший периметр.
4. Задано три матриці розмірності $N \times M$ ($2 \leq N \leq 10$, $2 \leq M \leq 10$). Розробити програму для знаходження тієї з цих матриць, що має найбільшу кількість рядків, які містять нулі.

2. 5. Рядкові величини

Питання для обговорення.

1. Рядкові літерали та їх завдання.
2. Завдання рядків.
3. Доступ до символів рядку. Зрізи.
4. Виконання дій над рядками та їхніми елементами.
5. escape послідовності.
6. Методи рядків.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Ввести текстовий рядок. Підрахувати у ньому кількість цифр.

Виконання.

Текст програми може бути таким:

```
s=input('Введіть рядок: ')
l=len(s)
dig='0123456789'
c=0
for i in range(l):
    if s[i] in dig: #символ s[i] входить до рядка dig
        c+=1
print('Кількість цифр у рядку=',c)
```

або таким:

```
s=input('Введіть рядок: ')
```

```

l=len(s)
dig='0123456789'
c=0
for i in range(l):
    if dig.find(s[i])>-1: #символ s[i] входить до
рядка dig
        c+=1
print('Кількість цифр у рядку=',c)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1b. Змінити програму так, щоб у ній підраховувалась кількість великих голосних латинських літер.

Завдання 1c. Змінити програму так, щоб у ній порівнювалась кількість великих і маленьких голосних латинських літер.

Завдання 1d. Ввести текстовий рядок. Порівняти кількість символів '*' і '?' у ньому. Якщо таких символів немає, вивести відповідне повідомлення.

Виконання.

Текст програми може бути таким:

```

print('Введіть рядок:')
s = input('рядок = ')
if s.count('*')==0 and s.count('?')==0:
    print('Символів "*" і "?" у рядку немає.')
else:
    if s.count('*')>s.count('?'):
        print('Символів "*" більше.')
    else:
        if s.count('*')<s.count('?'):
            print('Символів "?" більше.')
        else:
            print('Символів "*" і "?" однакова
кількість')

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 2a. Ввести текстовий рядок. Визначити, чи цей рядок паліндромом (таким, що однаково читається в прямому та зворотному напрямках).

Виконання.

Текст програми може бути таким:

```
s=input('Введіть рядок: ')
if s==s[::-1]:
    print('Паліндром')
else:
    print('Не паліндром')
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 2b. Змінити програму так, щоб заданий рядок виводився у зворотному напрямку (наприклад, *інформатика => акитамрофні*).

Завдання 3a. Ввести текстовий рядок, що складається зі слів, розділених довільною кількістю пропусків (наприклад, *мама мила рату*). Вивести слова, довжина яких більше трьох символів.

Виконання.

Текст програми може бути таким:

```
s = str(input('Введіть рядок: '))
l = list(s.split(' ')) #утворення списку слів шляхом
розбиття рядка за роздільником-пропуском
print('Список слів, утворених з рядка:')
print(l)
print('Список слів, довжина яких більша за 3:')
for w in l:
    if len(w)>3:
        print(w) #виведення слів довжиною більше трьох
символів
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклад виконання:

Введіть рядок: abc defg 1234 123 12345 ghjk er ewq

Список слів, утворених з рядка:

['abc', 'defg', '1234', '', '', '123', '12345', 'ghjk', 'er', '', 'ewq']

Список слів, довжина яких більша за 3:

defg

1234

12345

Ghjk

Як можна бачити з прикладу виконання, у випадку наявності між словами більше ніж одного пропуску, порожні слова між пропусками також потрапляють до списку.

Завдання 3b. Змінити програму так, щоб позбутися порожніх слів, а також можливих розділових знаків між ними (крапка, кома, знак оклику тощо).

Виконання.

Текст програми може бути таким:

```
s = str(input('Введіть рядок: '))
l = list(s.split(' ')) #утворення списку слів шляхом
розбиття рядка за роздільником-пропуском
print('Список слів:')
print(l)
ll=[] #створення нового списку без
порожніх слів і розділових знаків
for sl in l:
    w=sl.strip('.;.,!?!') #вилучення зі слова початкових
і кінцевих розділових знаків
    if w!='': #якщо слово не порожнє
        ll.append(w) #додати його до нового списку
print('Список непорожніх слів без розділових знаків:')
print(ll)
print('Список слів, довжина яких більша за 3:')
for w in ll:
    if len(w)>3:
        print(w) #виведення слів
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклади виконання:

Введіть рядок: bcd, abcde. df ertyu; dfgh, sdf, .. ghjk ?

Список слів:

```
['bcd,', ', ', 'abcde.', ', ', 'df', 'ertyu;', 'dfgh,', ', ', ', ', 'sdf,', '!', ', ', 'ghjk', '?']
```

Список непорожніх слів без розділових знаків:

```
['bcd', 'abcde', 'df', 'ertyu', 'dfgh', 'sdf', 'ghjk']
```

Список слів, довжина яких більша за 3:

abcde

ertyu

dfgh

ghjk

Завдання 3с. Змінити програму так, щоб у ній виводились слова, довжина яких більша за вказану користувачем.

Завдання 3д. Ввести текстовий рядок, що складається зі слів, розділених пропусками і розділовими знаками. Вивести слова парної довжини, які починаються з цифри.

Виконання.

Текст програми може бути таким:

```
s=str(input('Введіть рядок: '))
l=list(s.split(' '))
ll=[]
for sl in l:
    w=sl.strip('.;.,!?!')
    if w!='':
        ll.append(w)
print('Список слів парної довжини, що починаються з цифри:')
for w in ll:
    if (len(w)%2==0) and (w[0] in '0123456789'):
        print(w)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклади виконання:

Введіть рядок: abc 5f e4df 3def dddd 8oo

Список слів парної довжини, що починаються з цифри:

```
5f
3def
```

Завдання 3е. Змінити програму так, щоб у ній виводились слова, в яких перша і остання літери однакові.

Завдання 3ф. Змінити програму так, щоб у кожному слові передостання літера була замінена на символ '*' (зірочку).

Виконання.

Текст програми може бути таким:

```
s=str(input('Введіть рядок: '))
l=list(s.split(' '))
ll=[]
for sl in l:
    w=sl.strip('.;.,!?!')
    if w!='':
        ll.append(w)
```

```
print('Список змінених слів:')
for w in ll:
    ww=w[:len(w)-2]+'*'+w[len(w)-1]
    print(ww)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 3г. Змінити програму так, щоб у кожному слові друга літера була замінена на символ '+' (плюс).

Завдання 3н. Ввести текстовий рядок, що складається зі слів, розділених пропусками. Вивести слова, остання літера яких входить до них ще раз.

Виконання.

Текст програми може бути таким:

```
s=str(input('Введіть рядок: '))
l=list(s.split(' '))
ll=[]
for sl in l:
    w=sl.strip('.;,!?')
    if w!='':
        ll.append(w)
l11=[w for w in ll if w.count(w[len(w)-1])==2]
print('Список слів, остання літера яких входить до них ще раз:')
for w in l11: print(w)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 4а. Ввести текстовий рядок, що складається зі слів, розділених пропусками. Вивести слова, літери яких упорядковані за алфавітом (наприклад, abce, mpst).

Виконання.

Текст програми може бути таким:

```
s=input('Введіть рядок: ')
l=list(s.split(' '))
ll=[i.strip('.,!?') for i in l if i!='']
l11=[]
```

```

for w in ll:
    p=True #припущення, що слово упорядковане
    for i in range(len(w)-1):
        if w[i]>w[i+1]: #порушення умови упорядкованості
            flag=False
    if p:
        lll.append(w)
if len(lll)==0:
    print('Упорядкованих слів немає.')
else:
    print('Упорядковані слова:')
    for w in lll:
        print(w)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 4b. Змінити програму так, щоб у ній виводились слова, в яких парна кількість маленьких латинських літер 'o'.

Завдання 4c. Змінити програму так, щоб у ній виводились слова, які є паліндромами. Використати підпрограму, що визначає, чи є слово паліндромом.

Завдання 5a. Ввести текстовий рядок, що складається зі слів, розділених пропусками. Визначити слово, яке у рядку зустрічається найчастіше. Визначити кількість повторів цього слова у рядку.

Виконання.

Текст програми може бути таким:

```

print('Введіть рядок:')
s = str(input('рядок = '))
l = list(s.split(' '))
ll = [i.strip(',.!?') for i in l if i!='']
maxw = ''
maxn = 0
for w in ll:
    if (ll.count(w)>=maxn) and (w!=maxw):
        maxw = w
        maxn = ll.count(w)
print('Найчастіше зустрічається слово ',maxw)
print('Кількість повторів цього слова =',maxn)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 5b. Ввести текстовий рядок, що складається зі слів, розділених пропусками. Вивести слова, в яких максимальна кількість цифр.

Приклад.

Заданий рядок: 'abc m1nk v12 n345m zx19 678bbb k1 g56h'

Шукані слова: 'n345m' '678bbb'

Завдання 5c. Змінити програму так, щоб у ній виводились слова, що містять максимальну кількість літер «R».

2.6. Лабораторна робота №6. Рядкові величини

Зауваження:

1. *Інтерфейс програми має бути зрозумілим.*
2. *Кожне завдання має бути розв'язане в межах окремої програми.*
3. *Завдання мають бути розв'язані з максимальним використанням підпрограм.*
4. *Передбачається, що вхідний текстовий рядок не містить символів кирилиці.*

Варіанти завдань для самостійного виконання

Варіант 1.

1. Розробити програму, в якій буде визначатися, яких літер – «F» чи «f» – більше в заданому текстовому рядку.
2. Створити програму, в якій із заданого текстового рядка будуть виведені всі слова, довжина яких більше трьох символів.
3. Розробити програму, в якій із заданого текстового рядка будуть виведені слова, перша літера яких входить до них ще раз.
4. Створити програму, що визначає слово, яке в заданому реченні зустрічається максимальну кількість разів.

Варіант 2.

1. Створити програму, що визначає, скільки разів в даному текстовому рядку зустрічається перша літера.
2. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова, в яких остання літера замінена на «_» (нижнє підкреслення).
3. Створити програму, що в заданому реченні з кожного слова вилучає всі наступні входження першої літери.
4. Розробити програму, в якій із заданого текстового рядка будуть виведені окремими рядками слова парної та непарної довжини.

Варіант 3.

1. Розробити програму, в якій буде визначатися, чи містить заданий текстовий рядок більше ніж три літери «W».

2. Створити програму, в якій із заданого текстового рядка будуть виведені всі слова, довжина яких дорівнює довжині першого слова.
3. Розробити програму, в якій слова з заданого текстового рядка будуть виведені в зворотному порядку.
4. Створити програму, що визначає такі слова речення, які містять ті ж самі літери, що і перше слово цього речення.

Варіант 4.

1. Створити програму, що підраховує кількість цифр у заданому текстовому рядку.
2. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова з вилюченою другою літерою.
3. Створити програму, що в заданому реченні шукає всі слова-паліндроми (слова, що однаково читаються в прямому та зворотному напрямках).
4. Створити програму, що визначає слова, які в заданому реченні зустрічаються більше одного разу.

Варіант 5.

1. Створити програму, що в заданому текстовому рядку визначає номер позиції останньої коми.
2. Створити програму, в якій із заданого текстового рядка будуть виведені всі слова, в яких вилучено останню літеру.
3. Створити програму, що в заданому реченні визначає слова, в яких немає літер, що повторюються.
4. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова з однаковою кількістю літер «F» окремими рядками.

Варіант 6.

1. Створити програму, що визначає, чи є заданий текстовий рядок паліндромом (таким, що однаково читається в прямому та зворотному напрямках).
2. Створити програму, в якій із заданого текстового рядка будуть виведені всі слова непарної довжини.
3. Розробити програму, в якій із заданого текстового рядка будуть виведені слова, в яких немає літер, що повторюються.

4. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова однакової довжини окремими рядками.

Варіант 7.

1. Створити програму, що знаходить максимальну цифру в заданому текстовому рядку.
2. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова, в яких зустрічається номер Вашої академічної групи.
3. Створити програму, що в заданому реченні заходить слова, в яких більше голосних літер, ніж приголосних.
4. Створити програму, що визначає такі слова речення, які не містять літер, з яких складається останнє слово цього речення.

Варіант 8.

1. Розробити програму, в якій визначається, чи містить заданий текстовий рядок більше ніж чотири літери «X».
2. Створити програму, в якій із заданого текстового рядка будуть виведені всі слова, які не містять літери "a".
3. Розробити програму, в якій із заданого текстового рядка будуть виведені слова, що містять максимальну кількість літер «R».
4. Створити програму, що в заданому реченні знаходить слова, кількість літер в яких більша, ніж середня кількість літер в усіх словах речення.

Варіант 9.

1. Розробити програму, в якій буде підраховуватися кількість цифр у заданому текстовому рядку, що складають номер Вашої академічної групи.
2. Створити програму, в якій із заданого текстового рядка будуть виведені всі слова, в яких додано останню літеру «Z».
3. Створити програму, що в заданому реченні знаходить слова, в яких кількість великих літер збігається з кількістю маленьких.
4. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова, що зустрічаються не менше трьох разів в цьому рядку.

Варіант 10.

1. Розробити програму, в якій буде визначатися, яких літер – «G» чи «g» – більше в даному текстовому рядку.
2. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова з перенесеною останньою літерою на початок слова.
3. Створити програму, що в заданому реченні знаходить слова, в яких всі літери різні.
4. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова однакової довжини окремими рядками.

Варіант 11.

1. Розробити програму, в якій буде підраховуватися кількість великих літер в даному текстовому рядку.
2. Розробити програму, в якій із заданого текстового рядка будуть виведені всі слова парної довжини.
3. Розробити програму, в якій слова з заданого текстового рядка будуть виведені в зворотному порядку.
4. Розробити програму, в якій із заданого текстового рядка будуть виведені всі різні слова, що містяться в цьому рядку.

Варіант 12.

1. Створити програму, що в заданому текстовому рядку визначає номер позиції останнього знаку запитання.
2. Розробити програму, в якій із заданого текстового рядка будуть виведені слова, в яких зустрічається принаймні два рази номер Вашої академічної групи.
3. Створити програму, що в заданому реченні з кожного слова вилучає всі входження останньої літери .
4. Створити програму, що визначає слово, яке в заданому реченні зустрічається мінімальну кількість разів.

2.7. Множини та словники

Питання для обговорення.

1. Словники. Створення словників.
2. Виконання дій над елементами словника.
3. Методи словників.
4. Множини. Задання множини.
5. Виконання дій над елементами множини.
6. Порівняння множин.
7. Методи множин.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання Приклади розв'язування задач

Завдання 1а. Ввести рядок з латинських літер. Вивести всі різні літери цього рядка.

Виконання. Скористаємось властивістю множини, згідно з якою всі елементи множини різні.

Текст програми може бути таким:

```
st=str(input('Введіть рядок: '))
s=set(st)                # створення множини з символів
                          # рядка
print('Різні символи рядка:')
print(s)                 # множина виводиться у
                          # природному вигляді
```

або таким:

```
st=str(input('Введіть рядок: '))
s=set(st)
print('Різні символи рядка:')
for el in s:             #елементи множини виводяться у
                          #стовпчик
    print(el)
```

або таким:

```
st=str(input('Введіть рядок: '))
s=set(st)
print('Різні символи рядка:')
print(*s) # елементи множини виводяться
простим переліком
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклад виконання в першому випадку:

Введіть рядок: jgfgfjfgajgflafhnaksfnakf

Різні символи рядка:

```
{'g', 'k', 'n', 'f', 'j', 'h', 'l', 'a', 's'}
```

Приклад виконання в другому випадку:

Введіть рядок: jgfgfjfgajgflafhnaksfnakf

Різні символи рядка:

```
g
k
h
n
f
a
j
l
s
```

Приклад виконання в третьому випадку:

Введіть рядок: jgfgfjfgajgflafhnaksfnakf

Різні символи рядка:

```
l j g s k h f n a
```

Завдання 1b. Ввести рядок. Визначити кількість різних символів у ньому.

Виконання.

Текст програми може бути таким:

```
st=str(input('Введіть рядок: '))
s=set(st)
print('Кількість різних символів рядка=', len(s))
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1с. Ввести два текстові рядки. Створити з цих рядків дві множини, вивести елементи цих множин, упорядковані за зростанням, знайти перетин, об'єднання, різницю першої і другої множин.

Виконання.

Текст програми може бути таким:

```
st1=str(input('Введіть перший рядок: '))
s1=set(st1)
st2=str(input('Введіть другий рядок: '))
s2=set(st2)
l1=list(s1) # створ.списку з елементів множини для
подальшого упорядкування
l2=list(s2)
print('Упорядковані елементи першої множини:')
l1.sort()
print(l1)
print('Упорядковані елементи другої множини:')
l2.sort()
print(l2)
print('Перетин множин: ',s1&s2)
print('Об_єднання множин: ',s1|s2)
print('Різниця множин: ',s1-s2)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклад виконання:

Введіть перший рядок: ambmbvnb

Введіть другий рядок: mbjkghn

Упорядковані елементи першої множини:

['a', 'b', 'm', 'n', 'v']

Упорядковані елементи другої множини:

['b', 'g', 'h', 'j', 'k', 'm', 'n']

Перетин множин: {'n', 'm', 'b'}

Об_єднання множин: {'b', 'j', 'g', 'k', 'm', 'v', 'h', 'n', 'a'}

Різниця множин: {'a', 'v'}

Завдання 1d. Ввести текстовий рядок. Створити з символів цього рядка дві множини, перша з яких складається з цифр, друга – з кожного третього символу рядка. Якщо друга множина є підмножиною першої, знайти об'єднання множин, інакше знайти їхній перетин.

Виконання.

Текст програми може бути таким:

```
st=str(input('Введіть рядок: '))
```

```

s1=set ()
s2=set ()
sd=('0','1','2','3','4','5','6','7','8','9') #або
sd='0123456789'
l=len(st)
for i in range(l):
    if st[i] in sd:
        s1.add(st[i])
        if (i+1)%3==0: #визначення кожного третього
елемента: номери 2, 5, ...
            s2.add(st[i])
print ('Множина цифр: ',s1)
print ('Множина з кожного третього символа: ',s2)
if s2<=s1:
    print ('Об'єднання множин: ',s1|s2)
else:
    print ('Перетин множин :',s1&s2)

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклад виконання:

Введіть рядок: ab1cd3ef2gh3ab1cd3ef2gh3456abcdef

Множина цифр: {'6', '5', '1', '4', '3', '2'}

Множина з кожного третього символа: {'c', '6', '1', 'f', '3', '2'}

Перетин множин : {'1', '3', '2', '6'}

Введіть рядок: 234658365865936596296501747

Множина цифр: {'5', '6', '2', '8', '9', '4', '3', '1', '0', '7'}

Множина з кожного третього символа: {'5', '6', '8', '4', '1', '7'}

Об'єднання множин: {'5', '6', '2', '8', '9', '4', '3', '1', '0', '7'}

Завдання 1е. Вивести в порядку зростання всі цілі числа від 1 до 25, які можна подати у вигляді m^2+n^2 , $m,n \geq 0$

Виконання.

Текст програми може бути таким:

```

s1=set ()
for m in range(6):
    for n in range(6):
        s1.add(m*m+n*n) #формування множини із
потрібних ел-тів
print ('числа виду m^2+n^2')
for p in range(26):
    if p in s1:
        print(p, end=' ')

```

Ввести програму, запустити її на виконання, проаналізувати результат.

Завдання 2а. Ввести текстовий рядок, що складається зі слів, розділених довільною кількістю пропусків і, можливо, розділовими знаками. Вивести слова, довжина яких більше трьох символів. Для кожного такого слова побудувати і вивести множину, що складається з великих голосних латинських літер слова. Знайти і вивести об'єднання всіх отриманих множин.

Виконання.

Використаємо функцію, яка застосовується до окремого слова і повертає множину з великих латинських літер цього слова.

Текст програми може бути таким:

```
def setw(ww): #функція, що формує зі слова ww множину з
    великих лат. літер sw
    sw=set()
    lit='AEIOUY'
    for i in ww:
        if i in lit:
            sw.add(i)
    return sw

s = str(input('Введіть рядок: '))
l = list(s.split(' ')) #утворення списку слів шляхом
розбиття рядка за роздільником-пропуском
ll=[]
for sl in l:
    w=sl.strip('::,;!') #вилучення зі слова початкових
і кінцевих розділових знаків
    if w!='': #якщо слово не порожнє
        ll.append(w) #додати його до нового списку
print('Список слів, довжина яких більша за 3, і
відповідних множин:')
for w in ll:
    if len(w)>3:
        print('Слово: ',w,' Множина: ',setw(w))
res=set() #початкове значення об'єднання
всіх множин
for w in ll:
    if len(w)>3:
        res=res|setw(w) #додавання до об'єднання
чергової множини
print('Об'єднання множин: ',res)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклад виконання:

Введіть рядок: abcDEUF, OOPEEefgh.. IEO ADEFgh., OPQRY
Список слів, довжина яких більша за 3, і відповідних множин:
Слово: abcDEUF Множина: {'E', 'U'}
Слово: OOPEEefgh Множина: {'O', 'E', 'T'}
Слово: ADEFgh Множина: {'A', 'E'}
Слово: OPQRY Множина: {'O', 'Y'}
Об'єднання множин: {'A', 'E', 'Y', 'O', 'U', 'T'}

Завдання 2b. Змінити програму так, щоб для кожного слова парної довжини будувалась множина з цифр слова.

Завдання 3a. Описати словник, який містить відомості про людину: ім'я, стать, зріст. Створити і вивести список з вказаної користувачем кількості відомостей про людей.

Виконання.

Використаємо функцію, яка формує словник (запис) з відомостями про окрему людину. Параметром функції може бути номер запису для полегшення введення даних. Функція може не мати параметрів.

Текст програми може бути таким:

```
def inp_data(num):          # функція введення елементів
словника
    print('Запис № ', num, ':')
    d=dict()
    s=input('name: ')
    d['name']=s
    s=input('gender (m/f): ')
    d['gender']=s
    s=int(input('height: '))
    d['heidht']=s
    return d
n=int(input('Введіть кількість записів. n='))
dd=[]                       # створення списку словників
for i in range(n):
    d=inp_data(i)
    dd.append(d)
print('Отримані дані:', dd)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклад виконання:

Введіть кількість записів. n=3

Запис № 0 :

name: Ivan
gender (m/f): m
height: 175

Запис № 1 :

name: Maria
gender (m/f): f
height: 165

Запис № 2 :

name: Petro
gender (m/f): m
height: 180

Отримані дані: [{'gender': 'm', 'name': 'Ivan', 'height': '175'}, {'gender': 'f', 'name': 'Maria', 'height': '165'}, {'gender': 'm', 'name': 'Petro', 'height': '180'}]

Завдання 3б. Доповнити програму так, щоб у ній визначався середній зріст жінок зі списку.

Виконання.

Текст програми може бути таким:

```
def inp_data(num):
    print('Запис № ', num, ':')
    d=dict()
    s=input('name: ')
    d['name']=s
    s=input('gender (m/f): ')
    d['gender']=s
    s=int(input('height: '))
    d['height']=s
    return d
n=int(input('Введіть кількість записів. n='))
dd=[]
for i in range(n):
    d=inp_data(i)
    dd.append(d)
print('Отримані дані:', dd)
sf=0 # сумарний зріст жінок
cf=0 # кількість жінок
for i in range(n): # один спосіб перегляду списку
    (інший див. в задачі 3с)
    if dd[i]['gender']=='f':
        cf+=1
        sf+=dd[i]['height']
print('Середній зріст жінок = ', sf/cf)
```


Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 3с. Змінити попередню програму так, щоб у ній визначалося ім'я найвищого чоловіка зі списку.

Виконання.

Текст програми може бути таким:

```
def inp_data(num):
    print('Запис № ', num, ':')
    d=dict()
    s=input('name: ')
    d['name']=s
    s=input('gender (m/f): ')
    d['gender']=s
    s=int(input('height: '))
    d['height']=s
    return d
n=int(input('Введіть кількість записів. n='))
dd=[]
for i in range(n):
    d=inp_data(i)
    dd.append(d)
print('Отримані дані:', dd)
maxm=0
for i in dd:
    # інший спосіб перегляду списку
    if (i['gender']=='m') and (i['height']>=maxm):
        maxm=i['height']
        nm=i['name']
print('Найвищий чоловік: ', nm)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 3д. Змінити попередню програму так, щоб у ній визначалося, чи є у групі хоча б дві людини однакового зросту.

Виконання.

Текст програми може бути таким:

```
def inp_data(num):
    print('Запис № ', num, ':')
    d=dict()
    s=input('name: ')
    d['name']=s
    s=input('gender (m/f): ')
```

```

    d['gender']=s
    s=int(input('height: '))
    d['height']=s
    return d
n=int(input('Введіть кількість записів. n='))
dd=[]
for i in range(n):
    d=inp_data(i)
    dd.append(d)
print('Отримані дані:',dd)
p=False
for i in range(len(dd)-1):
    for j in range(i+1,len(dd)):
        if dd[i]['height']==dd[j]['height']:
            p=True
if p:
    print('У списку є люди однакового зросту')
else:
    print('У списку немає людей однакового зросту')

```

Завдання 3e. Доповнити словник, створений у *Завданні 3a*, відомостями про вік людини. Визначити і порівняти середній вік чоловіків і жінок зі списку.

Завдання 4a. Доповнити словник, створений у *Завданні 3a*, елементом, який є словником і містить відомості про адресу проживання особи: місто, вулиця, будинок, квартира.

Вивести прізвища будь-яких двох людей зі списку, що живуть в різних містах, але за однаковими адресами.

Виконання.

Текст програми може бути таким:

```

def inp_data(num):
    print('Запис № ', num, ':')
    d=dict()
    s=input('name: ')
    d['name']=s
    s=input('gender (m/f): ')
    d['gender']=s
    s=int(input('height: '))
    d['height']=s
    d1=dict() #створення нового елемента-словника
    t=input('town: ')
    d1['town']=t
    st=input('srteet: ')
    d1['street']=st

```

```

h=input('house: ')
d1['house']=h
f=input('flat: ')
d1['flat']=f
d['addr']=d1 #надання ключа новому елементу-
СЛОВНИКУ
return d
n=int(input('Введіть кількість записів. n='))
dd=[]
for i in range(n):
    d=inp_data(i)
    dd.append(d)
print('Отримані дані:',dd)
for i in range(n-1):
    for j in range(i+1,n):
        p1 = dd[i]['addr']['town']!=dd[j]['addr']['town']
        p2 = dd[i]['addr']['street']==dd[j]['addr']['street']
        p3 = dd[i]['addr']['house']==dd[j]['addr']['house']
        p4 = dd[i]['addr']['flat']==dd[j]['addr']['flat']
        if p1 and p2 and p3 and p4:
            print(dd[i]['name'], ' i ', dd[j]['name'],
'мають однакові адреси в різних містах')

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Приклад виконання:

Введіть кількість записів. n=3

Запис № 0 :

name: Ivan
gender (m/f): m
height: 180
town: Kyiv
srteet: Dovga
house: 1
flat: 1
Запис № 1 :
name: Maria
gender (m/f): f
height: 170
town: Kyiv
srteet: Dovga
house: 1
flat: 1

Запис № 2 :

name: Petro

gender (m/f): m

height: 175

town: Odessa

street: Dovga

house: 1

flat: 1

Отримані дані: [{'name': 'Ivan', 'height': 180, 'addr': {'house': '1', 'flat': '1', 'street': 'Dovga', 'town': 'Kyiv'}}, {'name': 'Maria', 'height': 170, 'addr': {'house': '1', 'flat': '1', 'street': 'Dovga', 'town': 'Kyiv'}}, {'name': 'Petro', 'height': 175, 'addr': {'house': '1', 'flat': '1', 'street': 'Dovga', 'town': 'Odessa'}}, {'name': 'Petro', 'height': 175, 'addr': {'house': '1', 'flat': '1', 'street': 'Dovga', 'town': 'Odessa'}, 'gender': 'm'}]

Ivan і Petro мають однакові адреси в різних містах

Maria і Petro мають однакові адреси в різних містах

2.8. Лабораторна робота №7. Множини та словники

Зауваження:

1. Інтерфейс програми має бути зрозумілим.
2. Кожне завдання має бути розв'язане в межах окремої програми.
3. Завдання мають бути розв'язані з максимальним використанням підпрограм .
4. В умовах задач три та чотири після вказання необхідного поля словника в дужках міститься вказівка щодо його можливого значення, а саме: “додатне ціле (дробове) число” – значення поля має обмежуватися можливим введенням додатного цілого (дробового) числа, “так/ні” – значення поля обмежується вибором «Так» або «Ні»
5. При розв'язуванні третьої задачі передбачити виконання пошуку словника з масиву за деякою умовою. Умову пошуку придумати самостійно.
6. При розв'язуванні четвертої задачі передбачити виконання пошуку словника з масиву за деякою умовою (умова має накладатися на поле вкладеного масиву). Умову пошуку придумати самостійно.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Розробити програму, в якій за двома заданими текстовими рядками формуються дві множини: S1 – множина голосних латинських літер першого рядка, S2 – множина голосних латинських літер другого рядка. Знайти різниці множин S2- S1 та S1-S2 .
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина великих літер латинського алфавіту і знаходиться об'єднання всіх сформованих множин.
3. Описати словник, який містить основні відомості про овочеву культуру: назва культури, країна походження насіння, середня врожайність (в ц/Га, додатне дійсне число), необхідність обробки від шкідників (так/ні), метод збирання врожаю. Розробити програму для створення масиву з відомостями про декілька овочевих культур.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про плоди овочевої культури, а саме: колір, середня вага одного плода (додатне ціле число), одиниці ваги плода. Внести

відповідні зміни до програми.

Варіант 2.

1. Розробити програму, в якій із заданого текстового рядка формуються дві множини: S_1 – множина символів, що знаходяться на парних позиціях, S_2 – множина символів, що знаходяться на непарних позиціях. Якщо множина S_1 є підмножиною S_2 , то знайти різницю множин $S_2 - S_1$, інакше знайти їх об'єднання.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина символів цифр і знаходиться перетин всіх сформованих множин.
3. Описати словник, який містить основні відомості про овочеву культуру: назва культури, країна походження насіння, середня врожайність (в ц/Га, додатне дійсне число), необхідність обробки від шкідників (так/ні), метод збирання врожаю. Розробити програму для створення масиву з відомостями про декілька овочевих культур.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про плоди овочевої культури, а саме: колір, середня вага одного плода (додатне ціле число), одиниці ваги плода. Внести відповідні зміни до програми.

Варіант 3.

1. Розробити програму, в якій за заданими двома текстовими рядками формуються дві множини: S_1 – множина голосних латинських літер першого рядка, S_2 – множина голосних латинських літер другого рядка. Знайти різниці множин $S_2 - S_1$ та $S_1 - S_2$.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи лабораторної роботи (рядкові величини), формується множина великих літер латинського алфавіту і знаходиться об'єднання всіх сформованих множин.
3. Описати словник, який містить основні відомості про авіарейс: номер рейсу (додатне ціле число), назву літака, місце призначення, кількість місць для пасажирів (шт., додатне ціле число), кількість проданих квитків (шт., додатне ціле число). Розробити програму для створення масиву з відомостями про декілька авіарейсів.

4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про багаж, а саме: тип багажу, загальна вага багажу (в кг., додатне дійсне число), оціночна вартість багажу (в грн., додатне ціле число). Внести відповідні зміни до програми.

Варіант 4.

1. Розробити програму, в якій за заданими двома текстовими рядками формуються дві множини: S1 – множина великих латинських літер першого рядка, S2 – множина великих латинських літер другого рядка. Знайти перетин цих множин.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина символів цифр і знаходиться перетин всіх сформованих множин.
3. Описати словник, який містить основні відомості про туристичну путівку: місяць відпочинку, місце відпочинку, кількість зірок в готелі (ціле число від 1 до 4), кількість туристів (додатне ціле число), вартість за 1-го туриста (в \$, додатне дійсне число). Розробити програму для створення масиву з відомостями про декілька туристичних путівок.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про транспортний засіб, а саме: назва, необхідність харчування в дорозі (так/ні), час в дорозі (год. додатне дійсне число). Внести відповідні зміни до програми.

Варіант 5.

1. Розробити програму, в якій із заданого текстового рядка формуються три множини: S1 – множина цифр, S2 – множина великих літер, S3 – множина символів на парних позиціях. Знайти об'єднання множин S1 і S2 та S2 і S3.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина великих літер латинського алфавіту і знаходиться об'єднання всіх сформованих множин.
3. Описати словник, який містить основні відомості про овочеву культуру: назва культури, країна походження насіння, середня врожайність (в ц/Га, додатне дійсне число), необхідність обробки від шкідників

(так/ні), метод збирання врожаю. Розробити програму для створення масиву з відомостями про декілька овочевих культур.

4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про плоди овочевої культури, а саме: колір, середня вага одного плода (додатне ціле число), одиниці ваги плода. Внести відповідні зміни до програми.

Варіант 6.

1. Розробити програму, в якій із заданого текстового рядка формуються дві множини: S_1 – множина маленьких латинських літер, що знаходяться на парних позиціях, S_2 – множина великих латинських літер, що знаходяться на непарних позиціях. Визначити, яка з множин містить більше елементів.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина символів цифр і знаходиться перетин всіх сформованих множин.
3. Описати словник, який містить основні відомості про бандероль: місто призначення, прізвище отримувача, місяць відправлення, дата відправлення (ціле число від 1 до 31), вага бандеролі (гр., додатне дійсне число), відмітка, що бандероль є цінною (так/ні). Розробити програму для створення масиву з відомостями про декілька туристичних путівок.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про адреса відправника бандеролі, а саме: місто відправлення, вулиця відправника, прізвище відправника. Внести відповідні зміни до програми.

Варіант 7.

1. Розробити програму, в якій із заданого текстового рядка формуються дві множини: S_1 – множина символів, що знаходяться на парних позиціях, S_2 – множина символів, що знаходяться на непарних позиціях. Якщо множина S_1 є підмножиною S_2 , то знайти різницю множин $S_2 - S_1$, інакше знайти їх об'єднання.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина великих літер латинського алфавіту і

знаходиться об'єднання всіх сформованих множин.

3. Описати словник, який містить основні відомості про постачання товару до магазину: номер магазину (додатне ціле число), назва товару, кількість одиниць товару (додатне ціле число), одиниці виміру, ціна одиниці товару (в грн., додатне дійсне число), надбавка магазину на товар (в %, додатне ціле число). Розробити програму для створення масиву з відомостями про декілька магазинів.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про виробника товару, а саме: назва фірми виробник, країна виробник, статутний капітал фірми виробника (в млн. \$, додатне дійсне число). Внести відповідні зміни до програми.

Варіант 8.

1. Розробити програму, в якій за заданими двома текстовими рядками формуються дві множини: S_1 – множина голосних латинських літер першого рядка, S_2 – множина голосних латинських літер другого рядка. Знайти різницю множин $S_2 - S_1$ та $S_1 - S_2$.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина символів цифр і знаходиться перетин всіх сформованих множин.
3. Описати словник, який містить основні відомості про авіарейс: номер рейсу (додатне ціле число), назву літака, місце призначення, кількість місць для пасажирів (шт., додатне ціле число), кількість проданих квитків (шт., додатне ціле число). Розробити програму для створення масиву з відомостями про декілька авіарейсів.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про багаж, а саме: тип багажу, загальна вага багажу (в кг., додатне дійсне число), оціночна вартість багажу (в грн., додатне ціле число). Внести відповідні зміни до програми.

Варіант 9.

1. Розробити програму, в якій за заданими двома текстовими рядками формуються дві множини: S_1 – множина великих латинських літер першого рядка, S_2 – множина великих латинських літер другого рядка. Знайти перетин цих множин.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до

умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина великих літер латинського алфавіту і знаходиться об'єднання всіх сформованих множин.

3. Описати словник, який містить основні відомості про авіарейс: номер рейсу (додатне ціле число), назву літака, місце призначення, кількість місць для пасажирів (шт., додатне ціле число), кількість проданих квитків (шт., додатне ціле число). Розробити програму для створення масиву з відомостями про декілька авіарейсів.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про багаж, а саме: тип багажу, загальна вага багажу (в кг., додатне дійсне число), оціночна вартість багажу (в грн., додатне ціле число). Внести відповідні зміни до програми.

Варіант 10.

1. Розробити програму, в якій із заданого текстового рядка формуються три множини: S1 – множина цифр, S2 – множина великих літер, S3 – множина символів, що знаходяться на парних позиціях. Знайти об'єднання множин S1 і S3 та S2 і S3.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина символів цифр і знаходиться перетин всіх сформованих множин.
3. Описати словник, який містить основні відомості про овочеву культуру: назва культури, країна походження насіння, середня врожайність (в ц/Га, додатне дійсне число), необхідність обробки від шкідників (так/ні), метод збирання врожаю. Розробити програму для створення масиву з відомостями про декілька овочевих культур.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про плоди овочевої культури, а саме: колір, середня вага одного плода (додатне ціле число), одиниці ваги плода. Внести відповідні зміни до програми.

Варіант 11.

1. Розробити програму, в якій за заданими двома текстовими рядками формуються дві множини: S1 – множина великих латинських літер першого рядка, S2 – множина великих латинських літер другого рядка. Знайти перетин цих множин .

2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина великих літер латинського алфавіту і знаходиться об'єднання всіх сформованих множин.
3. Описати словник, який містить основні відомості про туристичну путівку: місяць відпочинку, місце відпочинку, кількість зірок в готелі (ціле число від 1 до 4), кількість туристів (додатне ціле число), вартість за 1-го туриста (в \$, додатне дійсне число). Розробити програму для створення масиву з відомостями про декілька туристичних путівок.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про транспортний засіб, а саме: назва, необхідність харчування в дорозі (так/ні), час в дорозі (год. додатне дійсне число). Внести відповідні зміни до програми.

Варіант 12.

1. Розробити програму, в якій за заданими двома текстовими рядками формуються дві множини: S_1 – множина голосних латинських літер першого рядка, S_2 – множина голосних латинських літер другого рядка. Знайти різницю множин $S_2 - S_1$ та $S_1 - S_2$.
2. Розробити програму, в якій для кожного вибраного слова, відповідно до умови задачі №2 власного варіанту лабораторної роботи №6 (рядкові величини), формується множина великих літер латинського алфавіту і знаходиться об'єднання всіх сформованих множин.
3. Описати словник, який містить основні відомості про туристичну путівку: місяць відпочинку, місце відпочинку, кількість зірок в готелі (ціле число від 1 до 4), кількість туристів (додатне ціле число), вартість за 1-го туриста (в \$, додатне дійсне число). Розробити програму для створення масиву з відомостями про декілька туристичних путівок.
4. Доповнити словник, описаний при розв'язанні третьої задачі цієї лабораторної роботи, полем, яке є окремим словником і містить відомості про транспортний засіб, а саме: назва, необхідність харчування в дорозі (так/ні), час в дорозі (год., додатне дійсне число). Внести відповідні зміни до програми.

2.9. Файли

Питання для обговорення.

1. Відкриття та закриття файлу.
2. Атрибути файлового об'єкта.
3. Читання з файлу.
4. Запис у файл.
5. Додаткові методи роботи з файлами.
6. Використання менеджера контексту.

Рекомендована література

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Створити текстовий файл з ім'ям, вказаним користувачем. Записати у файл рядки довжиною не більш 10 символів, що вводяться користувачем. При введенні рядка більшої довжини у файл записуються перші 10 символів рядка. При введенні рядка 'zzz' припинити запис рядків у файл.

Виконання.

Текст програми може бути таким:

```
filename=input('Введіть ім'я нового файлу: ')
f=open(filename,'w') # файл відкривається для запису
даних
print('Вв. рядки файлу. zzz - закінчення введення ')
str=input()
while str!='zzz':
    f.write(str[:10]+'\\n') #у файл записуються перші 10
символів рядка і символ закінчення рядка
    str=input() # введення чергового рядка
f.close() #закриття файлу
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складеному контрольному прикладі. Перевірити, що на диску створився потрібний файл. Відкрити його і переконатися у правильності виконання програми.

Завдання 1b. Вивести на екран всі елементи файла, створеного у попередньому завданні.

Виконання.

Текст програми може бути таким:

```
filename=input('Введіть ім'я файлу для виведення даних:
')
f=open(filename, 'r') # файл відкривається для
читання з нього даних
print('Вміст файла:')
for str in f: # виведення усіх рядків з
файлу
    print(str, end='')
f.close()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складеному контрольному прикладі.

Завдання 1c. Дописати в кінець файла, створеного в попередньому завданні, нові довільні рядки. Якщо значення рядка, що вводиться, дорівнює 'zzz', припинити введення. Вивести вміст поповненого файла.

Виконання.

Текст програми може бути таким:

```
filename=input('Введіть ім'я файлу для додавання даних:
')
f=open(filename, 'a') # файл відкривається для
додавання даних
print('Вводіть рядки. zzz - закінчення введення ')
str=input()
while str!='zzz':
    f.write(str+'\n')
    str=input()
f.close()
f=open(filename, 'r') # відкриття файлу для читання
print('Вміст файла:')
for str in f: # виведення усіх рядків з
файлу
    print(str, end='')
```

```
f.close()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1d. Дано текстовий файл. Підрахувати кількість рядків файла.

Виконання. Для розв'язування цього завдання можна використати файл, створений у попередньому завданні, або будь-який інший текстовий файл.

Текст програми може бути таким:

```
filename=input('Введіть ім'я файлу для опрацювання: ')
f=open(filename, 'r')
k=0
for str in f:
    k+=1
print('Кількість рядків файла = ',k)
f.close()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1e. Дано текстовий файл. Порівняти у ньому кількість рядків парної довжини і рядків, що містять малу латинську літеру “к”.

Виконання.

Текст програми може бути таким:

```
filename=input('Введіть ім'я файлу для опрацювання: ')
f=open(filename, 'r')
p=0
k=0
for str in f:
    if (len(str)-1)%2==0: # -1, щоб не враховувався
        символ кінця рядка
        p+=1
        if 'k' in str:
            k+=1
print('p=',p,'k=',k)
if p>k:
    print('Рядків парної довжини більше.')
else:
    if p<k:
        print('Рядків з "к" більше.')
    else:
```

```
        print('Рядків однакова кількість')
f.close()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1f. Дано текстовий файл. Рядки файла, що мають максимальну довжину, вивести на екран і записати до іншого файла.

Виконання.

Текст програми може бути таким:

```
fname1=input('Введіть ім"я файла для опрацювання: ')
f=open(fname1, 'r')
fname2=input('Введіть ім"я нового файла: ')
g=open(fname2, 'w')
m=0
for str in f:
    if len(str)>m:
        m=len(str)

print('Максимальна довжина рядка =',m)
f.seek(0)
for str in f:
    if len(str)==m:
        g.write(str+'\n')
g.close()
g=open(fname2, 'r')
print('Вміст нового файла: ')
for str in g:
    print(str, end='')
f.close()
g.close()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 2a. Дано текстовий файл f, кожен рядок якого містить одне ціле число. Отримати новий файл g з елементів файла f, які кратні трьом і мають парні номери (0,2,4, ...).

Виконання.

Текст програми може бути таким:

```

fname1=input('Введіть ім"я файлу для опрацювання: ')
f=open(fname1, 'r')
fname2=input('Введіть ім"я нового файлу: ')
g=open(fname2, 'w')
i=0 #лічильник номерів елементів
for str in f:
    num=int(str)
    if (i%2==0) and (num%3==0):
        g.write(str)
    i+=1
g.close()
g=open(fname2, 'r')
print('Вміст нового файлу: ')
for str in g:
    print(str, end='')
f.close()
g.close()

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 2б. Дано текстовий файл f, кожен рядок якого містить три цілих числа. Вивести на екран максимальне з чисел у кожному рядку файла.

Виконання.

Текст програми може бути таким:

```

fname1=input('Введіть ім"я файлу для опрацювання: ')
f=open(fname1, 'r')
for str in f:
    a, b, c=str.split()
    a, b, c=int(a), int(b), int(c)
    m=max(a,b,c)
    print(m)
f.close()

```

або таким:

```

fname1=input('Введіть ім"я файлу для опрацювання: ')
f=open(fname1, 'r')
for str in f:
    m=max(map(int, str.split()))
    print(m)

```



```
f.close()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 3а. Сформувати цілочисельну квадратну матрицю розмірності n (n вводиться користувачем) з псевдовипадкових чисел від 1 до 5. Записати цю матрицю у природному вигляді (рядки і стовпці) у текстовий файл. Вміст файлу на екран не виводити, а переглянути за допомогою доступного текстового редактора.

Виконання.

Текст програми може бути таким:

```
import random
# формування матриці
n=int(input('Введіть розмірність матриці: '))
a=[]
for i in range(n):
    b=[]
    for j in range(n):
        x=random.randint(1,5)
        b.append(x)
    a.append(b)

#запис матриці до файлу
fname=input('Введіть ім'я нового файлу: ')
f=open(fname,'w')
for row in a:
    for el in row:
        print(str(el), sep=',', end=' ',file=f)
    print('',file=f)
f.close()
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 3б. Сформувати цілочисельну квадратну матрицю розмірності n з псевдовипадкових чисел від 10 до 20. Записати цю матрицю у природному вигляді (рядки і стовпці) у текстовий файл f . Прочитати вміст файлу і вивести його на екран. Непарні елементи прочитаної з

файлу матриці замінити нулями і записати перетворену матрицю у файл g. Вивести вміст файлу g на екран.

```
import random
n=int(input('Введіть розмірність матриці: '))
a=[]
for i in range(n):
    b=[]
    for j in range(n):
        x=random.randint(10,20)
        b.append(x)
    a.append(b)
fname=input('Введіть ім'я файлу f: ')
f=open(fname,'w')
for row in a:
    for el in row:
        print(str(el), sep='', end=' ',file=f)
    print('',file=f)
f.close()
f=open(fname,'r')
#читання матриці з файлу f у список matr
matr=[]
a=list(f)
for row in a:
    el=list(map(int, row.split()))
    matr.append(el)
print('Вміст файлу f: ')
for row in matr:
    for el in row:
        print(el,end=' ')
    print()
#перетворення елементів матриці і запис у файл g
gname=input('Введіть ім'я файлу g: ')
g=open(gname,'w')
for row in matr:
    for el in row:
        if el%2==1:
            el=0
        print(str(el), sep='', end=' ',file=g)
    print('',file=g)
f.close()
g.close()

g=open(gname,'r')
print('Вміст файлу g: ')
```

```

for str in g:
    print(str, end='')
g.close()

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 3с. Змінити програму так, щоб у ній використовувався файл *f* потрібної структури, заздалегідь створений на диску.

Завдання 3д. Сформувати цілочисельну матрицю розміром $m \times n$ ($m < 10$, $n < 10$) з псевдовипадкових чисел від 1 до 20. Записати цю матрицю у природному вигляді (рядки і стовпці) у текстовий файл *f*. Прочитати вміст файлу в матрицю і вивести його на екран. Визначити максимальний елемент у прочитаній з файла матриці і номер рядка, в якому він знаходиться. Якщо таких рядків декілька, розглянути перший з них. Елементи цього рядка відсортувати за зростанням. Записати перетворену матрицю у файл *g*. Вивести вміст файлу *g* на екран.

Виконання.

Текст програми може бути таким:

```

import random
m=int(input('Введіть кількість рядків матриці m
(m<=10): '))
n=int(input('Введіть кількість стовпців матриці n
(n<=10): '))
a=[]
for i in range(n):
    b=[]
    for j in range(n):
        x=random.randint(1,20)
        b.append(x)
    a.append(b)
fname=input('Введіть ім'я файлу f: ')
f=open(fname,'w')
for row in a:
    for el in row:
        print(str(el), sep='', end=' ',file=f)
    print('',file=f)
f.close()
f=open(fname,'r')
#читання матриці з файлу f у список matr
matr=[]

```

```

a=list(f)
for row in a:
    el=list(map(int, row.split()))
    matr.append(el)
print('Вміст файлу f: ')
for row in matr:
    for el in row:
        print(el,end=' ')
    print()
#перетворення елементів матриці і запис у файл g
gname=input('Введіть ім"я файлу g: ')
g=open(gname,'w')
mm=matr[1][1]
imm=1
for i in range(m):
    for j in range(n):
        if matr[i][j]>mm:
            mm=matr[i][j]
            imm=i
print('Максимальний елемент матриці = ',mm)
matr[imm].sort()
for row in matr:
    for el in row:
        print(str(el), sep='', end=' ',file=g)
    print('',file=g)
f.close()
g.close()
g=open(gname,'r')
print('Вміст файлу g: ')
for str in g:
    print(str, end='')
g.close()

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 3е. Змінити програму так, щоб у ній використовувався файл f потрібної структури, заздалегідь створений на диску.

2.10. Лабораторна робота №8. Текстові файли

Зауваження:

1. *Інтерфейс програми має бути зрозумілим.*
2. *Кожне завдання має бути розв'язане в межах окремої програми.*
3. *Завдання виконувати з максимальним використанням підпрограм.*
4. *Файли для опрацювання необхідно створити самостійно, використовуючи будь-який текстовий редактор.*
5. *При розв'язуванні четвертої задачі ім'я файлу з матрицею-результатом та його розміщення має збігатися з іменем та розміщенням вхідного файлу лише із зміненням розширенням на «res».*

Варіанти завдань для самостійного виконання

Варіант 1.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що містять вказану послідовність символів.
2. Дано текстовий файл, кожен рядок якого містить одне дійсне та одне ціле число. Розробити програму, в якій обчислюється сума цілих чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких одночасно присутні два символи, вказані користувачем.
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 2.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що

починаються з указанного символу.

2. Дано текстовий файл, кожен рядок якого містить два цілих числа. Розробити програму, в якій обчислюється добуток від'ємних чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки непарної довжини в яких першим, середнім або останнім символом є пропуск (місце знаходження символу вказує користувач).
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 3.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що містять вказану послідовність символів.
2. Дано текстовий файл, кожен рядок якого містить одне дійсне та одне ціле число. Розробити програму, в якій обчислюється сума цілих чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких одночасно присутні два символи вказані користувачем.
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 4.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що є парними або непарними в залежності від вибору користувача.
2. Дано текстовий файл, кожен рядок якого містить два дійсних числа. Розробити програму, в якій визначається найбільша сума пар чисел, що розміщуються в одному рядку даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких два символи з указаними номерами будуть однаковими.
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 5.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що починатимуться з вказаного символу.
2. Дано текстовий файл, кожен рядок якого містить два цілих числа. Розробити програму, в якій обчислюється добуток від'ємних чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки непарної довжини, в яких першим, середнім або останнім символом є пропуск (місце знаходження символу вказує користувач).
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить

певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 6.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що містять вказаний символ на вказаній позиції.
2. Дано текстовий файл, кожен рядок якого містить одне ціле число. Розробити програму, в якій визначається мінімальне з чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких декілька (кількість вказується користувачем) перших символів збігаються з такою ж кількістю останніх символів.
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 7.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються ті рядки довжина яких більша за вказану.
2. Дано текстовий файл, кожен рядок якого містить одне дійсне число. Розробити програму, в якій обчислюється сума додатних чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких вказаний символ зустрічається більше

одного разу.

4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 8.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що міститимуть вказану послідовність символів.
2. Дано текстовий файл, кожен рядок якого містить одне дійсне та одне ціле число. Розробити програму, в якій обчислюється сума цілих чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких одночасно присутні два символи, вказані користувачем.
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 9.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що є парними або непарними в залежності від вибору користувача.
2. Дано текстовий файл, кожен рядок якого містить два дійсних числа. Розробити програму, в якій визначається найбільша сума пар чисел, що розміщуються в одному рядку даного файлу.

3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких два символи з указаними номерами однакові.
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 10.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що починаються з вказаного символу.
2. Дано текстовий файл, кожен рядок якого містить два цілих числа. Розробити програму, в якій обчислюється добуток від'ємних чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки непарної довжини, в яких першим, середнім або останнім символом є пропуск (місце знаходження символу вказує користувач).
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 11.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що

- міститимуть вказаний символ на вказаній позиції.
2. Дано текстовий файл, кожен рядок якого містить одне ціле число. Розробити програму, в якій визначається мінімальне з чисел даного файлу.
 3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки, в яких декілька (кількість вказується користувачем) перших символів збігаються з такою ж кількістю останніх символів.
 4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

Варіант 12.

1. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу вибираються рядки, що починаються з вказаного символу.
2. Дано текстовий файл, кожен рядок якого містить два цілих числа. Розробити програму, в якій обчислюється добуток від'ємних чисел даного файлу.
3. Дано текстовий файл, кожен рядок якого містить деякий текст. Розробити програму, в якій із заданого файлу будуть переписані до іншого файлу всі рядки непарної довжини, в яких першим, середнім або останнім символом є пропуск (місце знаходження символу вказує користувач).
4. Доопрацювати третю задачу свого варіанту лабораторної роботи № 4 (Списки) таким чином, щоб дані про елементи матриці можна було завантажувати з текстового файлу наступної структури: файл містить певну кількість рядків (кожен з яких визначає коефіцієнти відповідного рядка матриці), в кожному з рядків містяться числа (розділені пропуском), які визначають коефіцієнти відповідних стовпців даного рядка матриці. Після перетворення матриця-результат має зберігатися в текстовому файлі тієї ж структури, що і файл з вхідною матрицею.

3. ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ У PYTHON

3.1. Об'єктний тип даних. Побудова ієрархічної залежності між об'єктами

Питання для обговорення.

1. Поняття об'єктно-орієнтованого програмування.
2. Основні концепції об'єктно-орієнтованого програмування.
3. Клас. Приклади опису класів.
4. Поняття об'єкту.
5. Конструктор, деструктор. Способи й цілі використання.
6. Поняття наслідування.

Рекомендована література

1. Крєневич А.П. PYTHON у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Режим доступу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/01/python-u-prykladakh-i-zadachakh.-ch2-oor.pdf>
2. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
3. Python 3.8.0 documentation. Режим доступу: <https://docs.python.org/3/>.
4. Васильєв А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
5. Програмування на мові Python (3.x). Початковий курс. Режим доступу: <https://sites.google.com/site/pythonukr/>.
6. ООП на Python: концепции, принципы и примеры реализации. Режим доступу: <https://proglib.io/p/python-oor>

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Описати клас “товар”, який характеризується назвою та ціною виробника. Вважати, що відпускна ціна товару в магазині більша за ціну виробника на 15%. В описаному класі передбачити наявність методу визначення відпускної ціни.

Виконання.

Текст програми може бути таким:

```
class Product:
    def __init__(self, name="", price=0):
        self.name=name
        self.price=price
    def newprice(self):
        return self.price*1.15

print("Введіть назву і ціну товару:")
n=input("Назва товару: ")
pr=int(input("Ціна товару: "))
p=Product(n,pr)
print(p.name, p.price)
print("Нова ціна товару:")
print(p.newprice())
```

Зауваження. `__init__` - конструктор за замовчуванням, в нього передаються початкові значення всіх полів з указанням їх значень за замовчуванням. Так, в програмі можна вказати `p=Product()` або `p=Product("Підручник")` Першим параметром будь-якого метода є посилання на конкретний об'єкт - `self`

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1в. Описати клас “Депозит”, який характеризується назвою та відсотковою ставкою. Ставка може бути збільшена в залежності від терміну дії депозиту. В описаному класі передбачити наявність методу обчислення відсоткової ставки за правилом: якщо термін більше 12-ти місяців, то ставка збільшується на 1%, якщо термін більше 24 місяців – на 2%, якщо термін більше 36 місяців – на 3%.

Виконання.

Текст програми може бути таким:

```
class Deposit:
    def __init__(self,name="",rate=0):
        self.name=name
        self.rate=rate
    def newrate(self,months=0): #months - локальна
змінна, self не
        inc=0 # вказується
        if 12<months<24:
            inc=1
        if 24<=months<36:
            inc=2
        if months>=36:
            inc=3
        return self.rate+inc
```

```
d=Deposit("Стандарт",4)
print("Назва депозиту і відсоткова ставка:")
print(d.name, d.rate)
m=int(input("Введіть кількість місяців: "))
print("Нова відсоткова ставка:", d.newrate(m))
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1с. Описати клас “Науковець”, який характеризується прізвищем та текстовими значеннями результатів наукової діяльності за останній рік (тези, стаття, підручник). В описаному класі передбачити наявність метод підрахунку кількості набраних балів до рейтингу науковця (за тези додається 2 бали, за статтю – 5 балів, за підручник – 10 балів).

Виконання.

Зуваження. Результати діяльності можна зокрема подати у вигляді текстового рядка, в якому тези позначені літерою *t*, , стаття – *s*, підручник – *p*. Символи записані в рядок згідно з хронологічним порядком публікації робіт, наприклад: "ttptssts"

Текст програми може бути таким:

```
class Scientist:
    def __init__(self,surname="",works=""):
```

```

        self.surname=surname
        self.works=works
def points(self):
    sum=0
    for w in self.works:
        if w=="t":
            sum+=2
        if w=="s":
            sum+=5
        if w=="p":
            sum+=10
    return sum

s=Scientist("Петренко","ssttppsstp")
print("Прізвище і доробок особи:")
print(s.surname, s.works)
print("Додано до рейтингу:"+str(s.points())+" балів.")

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах. Забезпечити введення полів користувачем.

Завдання 1d. Описати клас “метеостанція”, який характеризується назвою та значеннями середньомісячної температури повітря, вимірними упродовж 12 місяців.

В описаному класі передбачити наявність методів підрахунку середньої температури за цей період та різниці між найбільшою та найменшою температурою.

Виконання.

Зауваження. Значення температур можна зокрема подати у вигляді списку псевдовипадкових чисел.

Текст програми може бути таким:

```

import random
class Meteo:
    def __init__(self,name="",temp=[]):
        self.name=name
        self.temp=temp
    def av_temp(self):
        return sum(self.temp)/len(self.temp)
    def differ(self):
        return abs(max(self.temp)-min(self.temp))

```



```

m=input("Введіть назву метеостанції: ")
t=[random.randint(-20,20) for i in range(12)]
print("Значення температур:")
print(t)
met=Meteo(m,t)
print("Середня температура = ",met.av_temp())
print("Різниця температур = ",met.differ())

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах. Забезпечити введення полів користувачем.

Завдання 2а. Описати клас „вчитель”, який характеризується ім’ям та стажом роботи. В описаному класі визначити метод обчислення розміру заробітної плати, яка обчислюється на основі стажу роботи і мінімальної зарплати за такою формулою:

$$\text{мінімальна_зарплата} + \text{стаж} * \text{мінімальна_зарплата} / 10$$

Мінімальною зарплатою вважати 6000.

Описати похідний клас „вчитель-методист”, який характеризується відповідним званням. В похідному класі перевизначити метод обчислення розміру заробітної плати, враховуючи, що заробітна плата вчителя-методиста на 15% більша заробітної плати звичайного вчителя.

Виконання.

Текст програми може бути таким:

```

class Teacher:
    def __init__(self,surname="",years=0):
        self.surname=surname
        self.years=years
    def salary(self):
        return self.years*600+6000
# похідний клас від класу Teacher
class Met(Teacher):
# в конструктор додається нове поле rank
    def __init__(self,surname="",years=0,rank=""):
# успадкований конструктор
        Teacher.__init__(self,surname,years)
# ініціалізація нового поля
        self.rank=rank
# перевизначення методу обчислення зарплати
    def salary(self):

```

```

        return Teacher.salary(self)*1.15

a=Teacher("Петренко", 3)
print("Зарплата вчителя:")
print(a.salary())
b=Met("Ivanenko", 3, "методист")
print("Зарплата вчителя-методиста:")
print(b.salary())

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 2b. Описати клас „квартира”, який характеризується адресою та площею. В описаному класі визначити метод обчислення розміру орендної плати за квартиру, яка обчислюється за такою формулою: $\text{площа} \cdot 100$.

Описати похідний клас „квартира з ремонтом”, який характеризується типом ремонту (простий, європейський або люкс). В похідному класі перевизначити метод обчислення розміру орендної плати, враховуючи, що при простому ремонті вона збільшується на 1000, при європейському — на 2000, і при ремонті люкс — на 3000.

Виконання.

Текст програми може бути таким:

```

class Apartment:
    def __init__(self, addr="", square=0):
        self.addr=addr
        self.square=square
    def rent(self):
        return self.square*100
class Repairs(Apartment):
    def __init__(self, addr="", square=0, type=0):
        Apartment.__init__(self, addr, square)
        self.type=type
    def rent(self):
        add=0
        if self.type==0:
            add=1000
        if self.type==1:
            add=2000
        if self.type==2:

```

```
        add=3000
    return Apartment.rent(self)+add

a1=input("Введіть адресу квартири: ")
s1=int(input("Введіть площу: "))
ap1=Apartment(a1,s1)
print("Орендна плата = ", ap1.rent())

a2=input("Введіть адресу квартири: ")
s2=int(input("Введіть площу: "))
t2=int(input("Введіть тип ремонту: 0 - простий, 1-
євро, 2 - люкс: "))
ap2=Repairs(a2,s2,t2)
print("Орендна плата = ", ap2.rent())
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

3.2. Лабораторна робота № 9. Наслідування

Зауваження:

1. *Інтерфейс програми має бути зрозумілим.*
2. *Кожне завдання має бути розв'язане в межах окремої програми.*
3. *При розв'язуванні задач повторювані блоки операторів оформити у вигляді підпрограми.*
4. *В основній програмі першої задачі передбачити створення екземпляру описаного класу та виведення всіх його даних.*
5. *В основній програмі другої задачі передбачити послідовне створення екземплярів описаних класів та виведення всіх даних.*

Варіанти завдань для самостійного виконання

Варіант 1.

1. Описати клас “книга”, який характеризується назвою та ціною видавництва. Вважати, що відпускна ціна книги в магазині більша за ціну видавництва на 20%. В описаному класі передбачити наявність методу визначення відпускної ціни.
2. Описати клас „CD-диск”, який характеризується назвою та ємністю. В описаному класі визначити метод обчислення часу запису, яка залежить від ємності диску (відповідне співвідношення придумати самостійно). Описати похідний клас „CDRW-диск”, який характеризується кількістю виконаних перезаписів. В похідному класі перевизначити метод обчислення часу запису, враховуючи, що час збільшується на 30 секунд при кожному перезапису диску.

Варіант 2.

1. Описати клас “вчитель”, який характеризується прізвищем та окладом. Заробітна плата вчителя залежить від педагогічного стажу. В описаному класі передбачити наявність методу підрахунку заробітної плати за правилом: якщо стаж більше 3-ти років, то заробітна плата збільшується на 5%, якщо стаж більше 10 років, то – на 10%, якщо стаж більше 20 років, то на - 15%.
2. Описати клас „тварина”, який характеризується видом тварини, кількістю років життя та кількістю нащадків, що можуть народитися за один рік. В описаному класі визначити метод обчислення кількості нащадків за роки життя, вважаючи, що перші три роки тварина не народжує. Описати похідний клас „домашня тварина”, який характеризується прізвищем. В похідному класі перевизначити метод обчислення кількості нащадків за роки життя, враховуючи, що

домашня тварина народжує один раз кожні два роки.

Варіант 3.

1. Описати клас “футбольна команда”, який характеризується назвою команди та текстовими значеннями результатів за останні 5 матчів (виграли, програли, нічия). В описаному класі передбачити наявність метод підрахунку кількості набраних очок (за перемогу команді нараховується 2 очка, за нічию – 1 очко, за програш – 0 очок).
2. Описати клас „студент”, який характеризується ім’ям, спеціальністю та курсом. В описаному класі визначити метод обчислення номеру групи, яка залежить від курсу та спеціальності. Описати похідний клас „студент-магістр”, який характеризується темою магістерської роботи. В похідному класі перевизначити метод обчислення номеру групи, враховуючи, що до номеру групи всіх студентів магістрів в кінці дописують нуль (0).

Варіант 4.

1. Описати клас “трапеція”, який характеризується довжинами основ та висотою. В описаному класі передбачити наявність методів знаходження середньої лінії та площі трапеції.
2. Описати клас „корова”, який характеризується прізвиськом та віком. В описаному класі визначити метод обчислення ваги корови, яка залежить від її віку (відповідне співвідношення придумати самостійно). Описати похідний клас „корова-молочна”, який характеризується середньоденною кількістю видоєного молока. В похідному класі перевизначити метод обчислення ваги корови, враховуючи, що вага молочної корови на 10% менша від ваги звичайної корови такого ж віку.

Варіант 5.

1. Описати клас “двохоперандна арифметична операція”, який характеризується двома операндами. В описаному класі передбачити наявність методів виконання двох арифметичних операцій (суми і добутку).
2. Описати клас „викладач”, який характеризується ім’ям та стажем роботи. В описаному класі визначити метод обчислення розміру заробітної плати, яка залежить від стажу роботи (залежність розміру заробітної плати від стажу придумати самостійно). Описати похідний клас „викладач-кандидат”, який характеризується кандидатським

науковим ступенем. В похідному класі перевизначити метод обчислення розміру заробітної плати, враховуючи, що заробітна плата викладача-кандидата на 15% більша заробітної плати звичайного викладача.

Варіант 6.

1. Описати клас “потяг”, який характеризується місцем призначення та дальністю слідування. В описаному класі передбачити наявність методу обчислення кількості днів поїздки (за 1 день потяг може проїхати лише 500 км).
2. Описати клас „машина”, який характеризується маркою та кількістю циліндрів. В описаному класі визначити метод обчислення потужності, яка залежить від кількості циліндрів (відповідне співвідношення придумати самостійно). Описати похідний клас „вантажівка”, який характеризується вантажністю кузова. В похідному класі перевизначити метод обчислення потужності, враховуючи, що потужність вантажного автомобіля на 25% більша, ніж потужність звичайного автомобіля з такою ж кількістю циліндрів.

Варіант 7.

1. Описати клас “круг”, який характеризується радіусом. В описаному класі передбачити наявність методів знаходження довжини та площі кола.
2. Описати клас „літак”, який характеризується маркою та вантажопідйомністю. В описаному класі визначити метод обчислення розмаху крил, який залежить від вантажопідйомності (відповідне співвідношення придумати самостійно). Описати похідний клас „аеробус”, який характеризується мінімальною висотою польоту. В похідному класі перевизначити метод обчислення розмаху крил, враховуючи, що розмах крил аеробусу на 10% менший, ніж розмах крил звичайного літака.

Варіант 8.

1. Описати клас “робітник”, який характеризується прізвищем та розміром заробітної плати. В описаному класі передбачити наявність методу обчислення податку з заробітної плати. Податок на заробітну плату визначаються наступним чином: якщо заробітна плата робітника не перевищує 1100 грн, то вона податком не обкладається, якщо заробітна плата більша 1100 грн., але не перевищує 4000 грн. то податок

становить 10%, у випадку коли заробітна плата перевищує 4000 грн. то податок становить 20%.

2. Описати клас „студент”, який характеризується ім'ям, спеціальністю та курсом. В описаному класі визначити метод обчислення номеру групи, яка залежить від курсу та спеціальності.. Описати похідний клас „студент-практикант”, який характеризується назвою закладу проходження практики. В похідному класі перевизначити метод обчислення номеру групи, враховуючи, що всі студенти-практиканти тимчасово переводяться в групу з номером 40.

Варіант 9.

1. Описати клас “прямокутник”, який характеризується довжинами своїх сторін. В описаному класі передбачити наявність методів знаходження периметру та площі.
2. Описати клас „людина”, який характеризується іменем, зростом та вагою. В описаному класі визначити метод обчислення індексу ваги тіла ($\text{вага тіла (в кг.)} / (\text{зріст} * \text{зріст (в м.)})$). Описати похідний клас „неповнолітній”, який характеризується кількістю років, що залишилося до повноліття. В похідному класі перевизначити метод обчислення індексу ваги тіла, враховуючи, що індекс ваги тіла, розрахований для звичайної людини, потрібно зменшити на число, яке є часткою кількості років, що залишилося до повноліття, на 10.

Варіант 10.

1. Описати клас “студент”, який характеризується прізвищем та оцінками з 4 іспитів за п'ятибальною системою. В описаному класі передбачити наявність методів підрахунку середнього балу та рейтингу студента (% відношення максимальної можливої оцінки до середнього балу студента).
2. Описати клас „друковане видання”, який характеризується назвою та кількістю сторінок. В описаному класі визначити метод обчислення ціни, яка залежить від кількості сторінок (відповідне співвідношення придумати самостійно). Описати похідний клас „періодичне видання”, який характеризується періодом видання (в днях). В похідному класі перевизначити метод обчислення ціни, враховуючи, що ціна екземпляру періодичного друкованого видання збільшується на певну кількість відсотків, які залежать від періоду видання.

Варіант 11.

1. Описати клас “товар”, який характеризується назвою, кількістю і ціною одиниці. В описаному класі передбачити наявність методів підрахунку вартості наявного товару та вартості за зберігання товару (вартість зберігання одиниці товару складає 0,13% від його ціни).
2. Описати клас „бур'ян”, який характеризується назвою та сімейством. В описаному класі визначити метод обчислення періоду цвітіння, який залежить від сімейства (відповідне співвідношення придумати самостійно). Описати похідний клас „багаторічний-бур'ян”, який характеризується кількістю років життя. В похідному класі перевизначити метод обчислення періоду цвітіння, враховуючи, що період цвітіння багаторічного бур'яну зменшується на 3% за кожен рік життя по відношенню до цвітіння не багаторічного бур'яну.

Варіант 12.

1. Описати клас “лайнер”, який характеризується назвою, кількістю пасажиро-мість та водотоннажністю. В описаному класі передбачити наявність методів обчислення податку на проходження лайнера деякою протокою, вважаючи, що податок на проходження протоки становить 1 грн. за кожен тону водотоннажності лайнера та ще по 0,15 грн. за кожне пасажиро-місце.
2. Описати клас „країна”, який характеризується назвою, площею та кількістю населення. В описаному класі визначити метод обчислення щільності населення, яка залежить від площі країни та кількості населення. Описати похідний клас „президентська країна”, який характеризується прізвищем президента. В похідному класі перевизначити метод обчислення щільності населення, враховуючи, що президентська свита, яка складається з 1000 осіб, не входить в офіційну кількість населення, хоча і проживає в країні.

3.3. Основи об'єктно-орієнтованого програмування. Поліморфізм

Питання для обговорення.

1. Основні поняття об'єктно-орієнтованого програмування.
2. Поняття поліморфізму.
3. Перевизначення методу.
4. Статичні методи та методи класу.

Рекомендована література

1. Крєневич А.П. PYTHON у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Режим доступу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/01/python-u-prykladakh-i-zadachakh.-ch2-oor.pdf>
2. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
3. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
4. Васильєв А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
5. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.
6. ООП на Python: концепции, принципы и примеры реализации <https://proglib.io/p/python-oor>

Онлайн середовища програмування мовою Python

3. https://www.onlinegdb.com/online_python_compiler
4. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Описати клас „фігура,” в якому передбачений метод обчислення площі поверхні фігури. Описати похідні класи: „паралелепіпед”, „тетраедр”, „куля”. В похідних класах підмінити необхідні методи. Площа поверхні: кулі – $S=4\pi r^2$; тетраедра – $S=a^2 \sqrt{3}$.

Виконання.

Текст програми може бути таким:

```
import math
class Fig:
    # визначення базового класу
    def s(self):
    # визначення базового метода
    обчислення площі
        return 0
```

```

class Par(Fig):          # визначення похідних класів
    def __init__(self, a=0, b=0, c=0):
        self.a=a
        self.b=b
        self.c=c
    def s(self):        # підміна батьківського метода
        return self.a*self.b*self.c

class Cir(Fig):
    def __init__(self, r=0):
        self.r=r
    def s(self):
        return math.pi*self.r*self.r

class Tetr(Fig):
    def __init__(self, a=0):
        self.a=a
    def s(self):
        return self.a*self.a*3**(0.5)

f=[                    # Створення списку площ фігур
print("введіть радіус кулі")
r=int(input("r= "))
tmp=Cir(r)
f.append(tmp)
print("Введіть сторони паралелепіпеда:")
a=int(input("a= "))
b=int(input("b= "))
c=int(input("c= "))
tmp=Par(a,b,c)
f.append(tmp)
print("Введіть сторону тетраедра:")
a=int(input("a= "))
tmp=Tetr(a)
f.append(tmp)
print("Площі фігур:")
for i in f:
    print(i.s())

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1b. Описати клас „Викладач”, який характеризується полями Прізвище і Стаж. В класі передбачений метод обчислення базової заробітної плати.

Зарплата обчислюється на основі стажу роботи і мінімальної зарплати за такою формулою:

мінімальна_зарплата+стаж*100

Мінімальною зарплатою вважати 4000.

Описати похідні класи: „Погодинник” – викладач, який отримує зарплату за відпрацьовані години, „Штатник” – викладач, який отримує фіксовану зарплату в залежності від посади, „Викладач з неповною ставкою” – з зарплатою, яка становить певний відсоток від базової зарплати.

В похідних класах підмінити метод обчислення зарплати.

Виконання.

Текст програми може бути таким:

```
# визначення базового класу
class Vykladach:
    def __init__(self, prizm="", stazh=0):
        self.prizm=prizm
        self.stazh=stazh
    # визначення базового метода обчислення зарплати
    def zarpl(self):
        z=self.stazh*100+4000
        return z
# визначення похідних класів
class Pogodynnyk(Vykladach):
    def __init__(self, prizm="", stazh=0, godyn=0):
        Vykladach.__init__(self, prizm, stazh)
        self.godyn=godyn
    # підміна батьківського метода
    def zarpl(self):
        return self.godyn*300

class Shtatnyk(Vykladach):
    def __init__(self, prizm="", stazh=0, posada=""):
        Vykladach.__init__(self, prizm, stazh)
        self.posada=posada
    def zarpl(self):
        z=4000
        if self.posada=="доцент":
            z=6000
        if self.posada=="професор":
            z=8000
        return z
```

```

class Vidsotkova(Vykladach):
    def __init__(self,prizv="",stazh=0,vids=0):
        Vykladach.__init__(self,prizv,stazh)
        self.vids=vids
    def zarpl(self):
        return Vykladach.zarpl(self)*self.vids

print("Відомості про викладача з базовою зарплатою")
pr=input("Введіть прізвище викладача: ")
st=int(input("Введіть стаж= "))
osoba=Vykladach(pr,st)
print("Зарплата = ",osoba.zarpl())
print(25*"")

print("Відомості про викладача з погодинною оплатою")
pr=input("Введіть прізвище викладача: ")
st=int(input("Введіть стаж: "))
god=int(input("Введіть кількість годин: "))
osoba=Pogodynnuk(pr,st,god)
print("Зарплата = ",osoba.zarpl())
print(25*"")

print("Відомості про викладача-штатника")
pr=input("Введіть прізвище викладача: ")
st=int(input("Введіть стаж: "))
pos=input("Введіть посаду (доцент / професор): ")
osoba=Shtatnyk(pr,st,pos)
print("Зарплата = ",osoba.zarpl())
print(25*"")

print("Відомості про викладача з відсотковою ставкою")
pr=input("Введіть прізвище викладача: ")
st=int(input("Введіть стаж: "))
vid=float(input("Введіть відсоток ставки (0 - 1): "))
osoba=Vidsotkova(pr,st,vid)
print("Зарплата = ",osoba.zarpl())

```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 2а.

Задано категорії: Студент, викладач, персона, завідувач кафедри.

Встановити трирівневу ієрархію класів для наведених категорій.
Для кожного класу визначити поля, які будуть його характеризувати.
Кожен клас має містити принаймні один власний метод, який відсутній у батьківському класі (метод має відповідати предметній галузі).
В класах-нащадках мають підмінятися методи батьківського класу.
Написати демонстраційну програму, в якій створюються і опрацьовуються екземпляри різних класів з використанням змінної лише одного класу.

Виконання.

Задані категорії утворюють таку ієрархію (рис.5):

Рис.5. Ієрархія категорій для Завдання 2а

Батьківський клас Персона (Person) характеризується полями Прізвище (surname) і Вік (age).

Клас Персона містить метод Податок (tax), базовий розмір податку з особи обчислюється за формулою: вік*5.

Клас Студент (Student) є похідним від класу Персона. Він характеризується полями Прізвище (surname), Вік (age), Спеціальність (spec), Курс (course).

Клас Студент містить метод Група (group), в якому залежно від назви спеціальності (інформатика чи математика) і курсу обчислюється номер групи.

У класі Студент підмінено метод Податок батьківського класу. Податок зі студента дорівнює подвоєному базовому податку з особи.

Клас Викладач (Teacher) є похідним від класу Персона. Він характеризується полями Прізвище (surname), Вік (age), Стаж (years).

Клас Викладач містить метод Зарплата (salary), в якому залежно від стажу обчислюється розмір заробітної плати.

У класі Викладач підмінено метод Податок батьківського класу. Податок з викладача дорівнює потроєному базовому податку з особи.

Клас Завідувач кафедри (Director) є похідним від класу Викладач. Він характеризується полями Прізвище (surname), Вік (age), Стаж (years), Кафедра (dep).

У класі Завідувач кафедри підмінено метод Зарплата батьківського класу. Зарплата завідувача на 20% більша, ніж зарплата викладача.

У класі Завідувач кафедри підмінено метод Податок батьківського класу. Податок з завідувача кафедри на 10 грошових одиниць більше, ніж податок з викладача.

Текст програми може бути таким:

```
class Person:
    def __init__(self, surname="", age=0):
        self.surname=surname
        self.age=age
    def tax(self):
        return self.age*5

class Student(Person):
```

```

    def
__init__(self, surname="", age=0, spec="", course=0) :
    Person.__init__(self, surname, age)
    self.spec=spec
    self.course=course
def group(self):
    g=10
    if self.spec=="inf":
        g=(self.course%10)*10+2
    if self.spec=="math":
        g=(self.course%10)*10+3
    return g
def tax(self):
    return Person.tax(self)*2

class Teacher(Person):
    def __init__(self, surname="", age=0, years=0):
        Person.__init__(self, surname, age)
        self.years=years
    def salary(self):
        return self.years*500+5000
    def tax(self):
        return Person.tax(self)*3

class Director(Teacher):
    def __init__(self, surname="", age=0, years=0, dep=""):
        Teacher.__init__(self, surname, age, years)
        self.dep=dep
    def salary(self):
        return Teacher.salary(self)*1.2
    def tax(self):
        return Teacher.tax(self)+10

s=input("Введіть прізвище: ")
a=int(input("Введіть вік: "))
p=Person(s, a)
print("Податок = ", p.tax())

s=input("Введіть прізвище: ")
a=int(input("Введіть вік: "))
sp=input("Введіть спеціальність (inf / math): ")
c=int(input("Введіть курс: "))
p=Student(s, a, sp, c)
print("Група = ", p.group())

```



```
print("Податок = ", p.tax())
```

```
s=input("Введіть прізвище: ")  
a=int(input("Введіть вік: "))  
y=int(input("Введіть стаж: "))  
p=Teacher(s,a,y)  
print("Зарплата = ", p.salary())  
print("Податок = ", p.tax())
```

```
s=input("Введіть прізвище: ")  
a=int(input("Введіть вік: "))  
y=int(input("Введіть стаж: "))  
d=input("Введіть назву кафедри: ")  
p=Director(s,a,y,d)  
print("Зарплата = ", p.salary())  
print("Податок = ", p.tax())
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

3.4. Лабораторна робота №10. Поліморфізм

Зауваження:

1. Інтерфейс програми має бути зрозумілим.
2. При виконанні першого завдання в основній програмі передбачити створення похідних класів та виконання передбачених обчислень.
3. Для другої задачі:
 - Встановити і обґрунтувати ієрархію класів для наведених категорій (термінів), ієрархія має бути тривірневою.
 - Для кожного класу визначити поля, які його характеризують.
 - Написати демонстраційну програму, в якій створюються і опрацьовуються екземпляри різних класів з використанням змінної лише одного класу.
4. Третьою задачею є доповнення другої задачі з урахуванням наступних вимог:
 - Кожен клас має містити принаймні один власний метод, який відсутній у батьківському класі (метод має бути сформульований самостійно і стосуватися предметної області задачі).
 - В класах нащадках мають підмінятися чи перевизначатися методи батьківського класу.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Описати клас „фігура”, в якому передбачений метод обчислення площі поверхні фігури. Описати похідні класи: „паралелепіпед”, „тетраedr”, „куля”. В похідних класах підмінити необхідні методи. Площа поверхні: кулі - $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$; тетраедра -
$$h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2}$$
2. Студент, викладач, персона, завідувач кафедри.

Варіант 2.

1. Описати клас „службовець”, в якому передбачений метод обчислення заробітної плати. Описати похідні класи: „службовець-погодинник” - з зарплатою за відпрацьовані години, „службовець-штатник” - з фіксованою зарплатою, „службовець-з відсотковою ставкою” - з зарплатою, яка становить певну кількість відсотків від загального

прибутку. Метод нарахування заробітної плати для різних класів визначити самостійно.

2. Робітник, кадри, інженер, адміністрація.

Варіант 3.

1. Описати клас „крива”, в якому передбачений метод обчислення координати кривої у для деякого заданого х та визначення координатної чверті, в якій знаходиться точка з відповідними координатами (х, у). Описати похідні класи: „пряма”, „еліпс”, „гіпербола”. В похідних класах підмінити необхідні методи. Рівняння: прямої - $y=ax+b$; еліпса - $S=h \frac{a+b}{2}$; гіперболи $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$.
2. Організація, страхова компанія, завод, кораблебудівний завод.

Варіант 4.

1. Описати клас „норма”, в якому передбачений метод обчислення норми. Описати похідні класи: „комплексне число”, „вектор” - з десяти елементів, „матриця” - розмірності 2×2 . В похідних класах підмінити необхідні методи. Норма: комплексного числа - $z=\sqrt{a^2+b^2}$, де $z=a+ib$; вектору – корінь квадратний з суми елементів значень по модулю; матриці – максимальне значення по модулю.
2. Журнал, книга, друковане видання, підручник.

Варіант 5.

1. Описати клас „фігура”, в якому передбачений метод обчислення об'єму фігури. Описати похідні класи: „паралелепіпед”, „тетраedr”, „куля”. В похідних класах підмінити необхідні методи. Об'єм: кулі $S=\frac{4}{3} \pi r^3$, тетраедра $S=a^3 \frac{\sqrt{2}}{12}$.
2. Місто, область, мегаполіс, територія.

Варіант 6.

1. Описати клас „Система числення”, в якому передбачений метод переведення числа певної системи числення в десяткову систему числення. Описати похідні класи: „двійкова”, „шістнадцяткова”. В похідних класах підмінити необхідні методи. Для переведення числа в

десяткову систему його необхідно розкласти за ступенями основи системи числення, наприклад для двійкової системи: $10101_2 = 2^4 + 2^2 + 2^0 = 21_{10}$; для шістнадцяткової системи числення: $E1F_{16} = E \cdot 16^2 + 1 \cdot 16^1 + F \cdot 16^0 = 14 \cdot 16^2 + 1 \cdot 16^1 + 15 \cdot 16^0 = 3584 + 16 + 15 = 3615_{10}$.

2. Квитанція, накладна, документ, чек.

Варіант 7.

1. Описати клас „Тейлор”, в якому передбачений метод обчислення значення функції, розкладеної в ряд Тейлора для деякого аргументу з указаною кількістю членів ряду. Описати похідні класи: „sin”, „cos”, „exp”. В похідних класах підмінити необхідні методи. Формули розкладу в ряд Тейлора мають вигляд:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots, \quad \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots,$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

2. Автомобіль, потяг, транспортний засіб, експрес.

Варіант 8.

1. Описати клас „прогресія”, в якому передбачений метод обчислення суми прогресії. Описати похідні класи: „арифметична”, „геометрична”. В похідних класах підмінити необхідні методи. Арифметична прогресія - $an = a1 + (n-1) \cdot d$, (d – різниця прогресії); сума арифметичної прогресії - $S_n = n \cdot \frac{a_1 + a_n}{2}$; геометрична прогресія - $bn = b1 \cdot r^{n-1}$, (r – знаменник прогресії); сума геометричної прогресії - $S_n = \frac{b_1(r^n - 1)}{r - 1}$, при $r \neq 1$.

2. Двигун, двигун внутрішнього згорання, дизель, турбореактивний двигун.

Варіант 9.

1. Описати клас „обмінник”, в якому передбачений метод конвертації грошових одиниць між двома валютами. Описати похідні класи: „грн-\$”, „грн-€”, „грн-руб”. В похідних класах підмінити необхідні методи. Курс переведення взяти самостійно.

Республіка, монархія, королівство, держава.

Варіант 10.

1. Описати клас „сума факторіалів”, в якому передбачений метод обчислення суми виду $S=1!+2!+3!+4!+\dots+n!$ з окремим методом обчислення факторіалу числа. Описати похідні класи: „факторіал цикл” - факторіал обчислити за допомогою циклічного підсумовування (рекурсивно), „факторіал формула” - факторіал обчислити за формулою $n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$. В похідних класах підмінити необхідні методи. Порівняти отримані результати.
2. Ссавець, парнокопитне, птах, тварина.

Варіант 11.

1. Описати клас „чотирикутник”, в якому передбачений метод обчислення площі чотирикутника. Описати похідні класи: „прямокутник”, „паралелограм”, „трапеція”. В похідних класах підмінити необхідні методи.
2. Плоска фігура, коло, куля, еліпс.

Варіант 12.

1. Описати клас „дівчина”, в якому передбачений метод обчислення дівочого щастя. Описати похідні класи: „старшокласниця”, „студентка першого курсу”, „студентка п'ятого курсу”. В похідних класах підмінити необхідні методи. Дівоче щастя: старшокласниці - «кількість поцілунків за тиждень/кількість відвіданих дискотек за тиждень/7*100%»; студентки першого курсу - «середній бал за сесію*кількість хлопців за семестр*100 %»; студентки п'ятого курсу - «50% за одруженість + 50% за відсутність розподілу в село».
2. Аероплан, літак, вертоліт, повітряний транспорт.

3.5. Агрегація

Питання для обговорення.

1. Основні концепції об'єктно-орієнтованого програмування.
2. Поняття агрегації.

Рекомендована література

1. Крєневич А.П. PYTHON у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Режим доступу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/01/python-u-prykladakh-i-zadachakh.-ch2-oor.pdf>
2. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
3. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
4. Васильєв А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
5. Програмування на мові Python (3.x). Початковий курс. Режим доступу: <https://sites.google.com/site/pythonukr/>.
6. ООП на Python: концепции, принципы и примеры реализации. Режим доступу: <https://proglib.io/p/python-oor>

Онлайн середовища програмування мовою Python

5. https://www.onlinegdb.com/online_python_compiler
6. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Авіарейс характеризується: місцем призначення, літаком та кількістю проданих квитків. Місце призначення, як окремий клас, характеризується місцем призначення та кількістю кілометрів до нього. Літак, як окремий клас, характеризується середньою швидкістю польоту та вантажопідйомністю (максимальна вага вантажу, який літак може взяти на борт). Описати клас “Авіарейс”, в якому передбачити метод підрахунку ваги поштового вантажу (в кг), який можна розмістити на борту (враховуючи, що середня вага одного пасажирів разом з багажем становить 150 кг). Використати екземпляр класу „Авіарейс” для визначення, чи буде доставлений певний поштовий вантаж заданої ваги цим літаком до місця призначення і якщо так, то через скільки годин.

Виконання.

Текст програми може бути таким:

```
# тут описується агрегація полів, які є не стандартними класами,
```

```

# а створеними користувачем
# опис класу Місце призначення
class Destination:
    def __init__(self,city="",dist=0):
        self.city=city
        self.dist=dist
# Опис класу Літак
class Plane:
    def __init__(self,speed=0,cargo=0):
        self.speed=speed
        self.cargo=cargo
# Опис класу Авіарейс, полями якого є об'єкти
# попередньо описаних
# класів Місце призначення, Літак і поле Кількість
# квитків
class Fly:
    def
__init__(self,city="",dist=0,speed=0,cargo=0,count=0):
        self.ds=Destination(city,dist) # об'єкт класу
Місце пр.
        self.pl=Plane(speed,cargo) # об'єкт класу Літак
        self.count=count
    def weight(self):
        return self.pl.cargo-self.count*150
# деструктор, де значуються поля, що є об'єктами
def __del__(self):
    del self.ds
    del self.pl

City=input("Місто призначення: ")
Dist=int(input("Відстань до міста: "))
Speed=int(input("Швидкість = "))
Cargo=int(input("Вантажопідйомність = "))
Count=int(input("Продано квитків = "))
fl=Fly(City,Dist,Speed,Cargo,Count)
Weight=int(input("Вага поштового вантажу = "))
if fl.weight()>=Weight:
    print("Вантаж буде доставлено.")
    print("Час доставки = ",fl.ds.dist/fl.pl.speed)
else:
    print("Вантаж не буде доставлено.")
Ввести програму, запустити її на виконання, перевірити
роботу програми на самостійно складених контрольних
прикладках.

```

Завдання 1в. Лікарський засіб характеризується: виробником, складом, вартістю випуску. Виробник, як окремий клас, характеризується назвою і країною розміщення. Склад, як окремий клас, характеризується вагою наявних наркотичних препаратів та містить метод обчислення додаткової вартості, яка обчислюється за формулою „вага_наркотичних_речовин”*10, причому в країні USA додаткова вартість не встановлюється. Описати клас „Лікарські засоби”, в якому передбачити методи обчислення загальної вартості виробництва лікарського засобу. Використати екземпляр класу „Лікарські засоби” для обчислення роздрібної вартості деякого лікарського засобу, на яку встановлюється 30 % надбавка до його вартості виробництва.

Виконання.

Текст програми може бути таким:

```
class Producer:
    def __init__(self, name="", state=""):
        self.name=name
        self.state=state

class Composition:
    def __init__(self, drug=0):
        self.drug=drug
    def add_cost(self, state):
        add=0
        if state!="USA":
            add=self.drug*10
        return add

class Medicine:
    def __init__(self, name="", state="", drug=0, cost=0):
        self.pr=Producer(name, state) # полями є об_екти
попередньо описаних класів
        self.cm=Composition(drug)
        self.cost=cost
    def fullcost(self):
        return
self.cost+self.cm.add_cost(self.pr.state)
    def __del__(self):
        del self.pr
        del self.cm

Name=input("Виробник препарату: ")
State=input("Країна = ")
Drug=int(input("Вага наркотичних речовин, грам = "))
```



```
Cost=int(input("Вартість випуску = "))
med=Medicine(Name,State,Drug,Cost)
price=med.fullcost()*1.3
print("Роздрібна ціна в ",State," = ",price)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

3.6. Лабораторна робота № 11. Агрегація

Зауваження:

1. *В задачах, в яких наявна необхідність роботи з датами (місяцями та днями), для спрощення, можна вважати кількість днів в місяці фіксованою і рівною 30. Місяць може задаватися як його порядковий номер. Окрім того, вважати, що переходу між роками при роботі програми не буде.*

Варіанти завдань для самостійного виконання

Варіант 1.

1. Авіарейс характеризується: місцем призначення, літаком та кількістю проданих квитків. Місце призначення, як окремий клас, характеризується: містом призначення та кількістю кілометрів до нього. Літак, як окремий клас, характеризується середньою швидкістю польоту та вантажопідйомністю (максимальна вага вантажу, який літак може взяти на борт). Описати клас “Авіарейс”, в якому передбачити метод підрахунку ваги поштового вантажу (в кг), який можна розмістити на борту (враховуючи, що середня вага одного пасажира разом з багажем рівна 150 кг). Використати екземпляр класу „Авіарейс” для визначення, чи буде доставлений певний поштовий вантаж заданої ваги цим літаком до місця призначення, і якщо так, то за скільки годин.

Варіант 2.

1. Вибираючи туристичну путівку, клієнт вибирає: місце відпочинку, період відпочинку, місце проживання та транспортний засіб, яким він зможе дістатися до місця відпочинку. Місце проживання, як окремий клас, характеризується назвою типу місця проживання (готель, турбаза, кемпінг, палата) та містить метод підрахунку вартості проживання за добу в залежності від вибраного місця проживання (відповідне співвідношення придумати самостійно). Транспортний засіб, як окремий клас характеризується назвою типу транспортного засобу (літак, поїзд, автобус, корабель) та містить метод підрахунку вартості проїзду в один кінець в залежності від вибраного транспортного засобу (відповідне співвідношення придумати самостійно). Описати клас „Туристична путівка”, в якому передбачити метод підрахунку загальної вартості путівки, що складається з вартості проживання, вартості дороги в двох напрямках та 30% від загальної суми за послуги туристичної фірми. Використати екземпляр класу „Туристична путівка” для обчислення вартості деякої туристичної

пугівки.

Варіант 3.

1. Овочева культура характеризується: назвою, сортом, параметрами дозрівання, тим, що висаджується на певній площі (в Га) і має врожайність. Параметри дозрівання, як окремий клас, характеризується числом і місяцем посадки, кількістю місяців і днів дозрівання та містить метод обчислення дати дозрівання. Врожайність, як окремий клас, характеризується вагою отриманого врожаю з 1 Га та можливим відсотком псування врожаю. Описати клас „Овочева культура”, в якому передбачити метод обчислення загальної врожайності з засіяної площі з урахуванням можливих втрат від псування врожаю. Використати екземпляр класу „Овочева культура” для визначення дати дозрівання та врожайності деякої культури.

Варіант 4.

1. Поштова бандероль характеризується: адресою відправника, адресою одержувача, датою відправлення та додатковими параметрами. Адреса відправника та одержувача, як окремий клас, характеризується вулицею, номером будинку та квартири. Додаткові параметри, як окремий клас, характеризується вагою (кг) та цінністю (грн) вмісту бандеролі та містить метод обчислення вартості її доставки в залежності від ваги і цінності (відповідне співвідношення придумати самостійно). Дата відправлення, як окремий клас, характеризується днем і місяцем відправлення та містить метод визначення дати доставки бандеролі (бандероль доходить до одержувача через 7 днів після відправлення). Описати клас „Бандероль”, в якому передбачити метод обчислення виплати за затримку бандеролі в розмірі 10% від її цінності за кожен день затримки. Використати екземпляр класу „Бандероль” для визначення, чи була доставлена деяка бандероль вчасно (день і місяць реальної доставки має вказати користувач) і в разі її затримки обчислити розмір повернення коштів.

Варіант 5.

1. Власник квартири характеризується: прізвищем, адресою та комунальними витратами. Адреса, як окремий клас, характеризується: назвою вулиці, номером будинку, номером квартири, розміром місячної оплати за прибирання прибудинкової території та містить метод обрахунку поверху розташування квартири (в місті побудовані будинки з трьома квартирами на поверсі та з одним під'їздом).

Комунальні послуги, як окремий клас, характеризується: назвою звітного місяця, показниками лічильника на початок і кінець місяця за електроенергію (кВт*год) і воду (м³) та містить методи підрахунку плати за використану електроенергію та воду (окремо для кожного ресурсу), враховуючи, що на поверхні вище 4-го вода не завжди доходить через поганий тиск, мешканці цих поверхів за воду платять на 50% менше (тарифи на електроенергію та воду придумати самостійно). Описати клас „Власник квартири”, в якому передбачити метод обчислення загальної плати за комунальні послуги. Використати екземпляр класу „Власник квартири” для визначення залишків коштів після оплати комунальних послуг при вказаних місячних доходах.

Варіант 6.

1. Метеорологічні спостереження характеризуються: датою та параметрами вимірювання погодних умов. Дата, як окремий клас, характеризується місяцем, днем, кількістю денних годин, годиною, яка вважається початком дня та містить метод який перевіряє, чи входить вказана година в денні години. Параметри вимірювання погодних умов, як окремий клас, характеризується вимірами температури кожні 6 годин (о 1, 7, 13, 19 годинах) та містить метод обчислення середньодобової температури. Описати клас “Метеорологічні спостереження”, в якому передбачити методи обчислення середньоденної та середньо-нічної температури. Використати екземпляр класу „Метеорологічні спостереження” для визначення середньодобової, середньоденної та середньо-нічної температури деякий день.

Варіант 7.

1. Робітник характеризується: прізвищем, освітою (середня, незавершена вища, вища), місцем проживання, заробітною платнею. Місце проживання, як окремий клас, характеризується: назвою вулиці, відстанню до підприємства (в км) та містить метод обчислення компенсації за проїзд на роботу, яка становить 0,1 % від заробітної плати за 1 км. Заробітна платня, як окремий клас, характеризується фіксованим розміром платні та містить метод обчислення надбавки за наявність освіти (розмір надбавки придумати самостійно). Описати клас “Робітник”, в якому передбачити методи підрахунку загальної заробітної плати та компенсації за проїзд. Використати екземпляр класу „Робітник” для підрахунку загальних грошових надходжень деякого робітника.

Варіант 8.

1. Замовлення на товар характеризується: замовником та товаром. Замовник, як окремий клас, характеризується назвою фірми, сумою коштів попередньо зроблених покупок, від якої залежить % знижки та методом підрахунку знижки (відповідне співвідношення придумати самостійно). Товар, як окремий клас, характеризується назвою товару, кількістю замовлених одиниць, ціною за одну одиницю та містить метод підрахунку загальної вартості, на яку замовлено товар. Описати клас “Замовлення”, в якому передбачити метод підрахунку суми, яку потрібно сплатити постачальнику, та метод збільшення суми коштів попередньо зроблених покупок на суму поточної покупки. Використати екземпляр класу “Замовлення” для моделювання ситуації покупки деякого товару.

Варіант 9.

1. Студент характеризується: прізвищем, місцем проживання та результатами навчання. Місце проживання, як окремий клас, характеризується відміткою, чи проживає студент в гуртожитку, та містить метод встановлення місячної плати за гуртожиток (розмір плати за гуртожиток придумати самостійно). Результати навчання, як окремий клас, характеризується оцінками з чотирьох іспитів, які отримав студент в останню сесію та методом обчислення середнього балу та методом встановлення розміру стипендії в залежності від середнього балу. Описати клас “Студент”, в якому передбачити метод підрахунку суми коштів, які студент отримуватиме на руки. Використати екземпляр класу “Студент” для визначення отриманих коштів студентом чи заборгованість на кінець місяця.

Варіант 10.

1. Читач бібліотеки характеризується: прізвищем, адресою, назвою книги, її ціною, датою її видачі та відміткою, чи була повернена попередня книга. Адреса, як окремий клас, характеризується назвою вулиці, відстанню до бібліотеки (в км) та методом обчислення вартості доставки читачу повідомлень, яка залежить від відстані до бібліотеки (відповідне співвідношення придумати самостійно). Дата видачі книги, як окремий клас, характеризується місяцем і числом видачі книги та містить метод підрахунку місяця та числа, що відповідає останньому дню повернення книги (книга може бути надана читачу не більше ніж

на 10 днів). Описати клас “Читач”, в якому передбачити метод обчислення розміру можливого штрафу за невчасне повернення книги в розмірі 50 % від вартості книги. Використати екземпляр класу “Читач” для проведення операції взяття деякої книги (за умови, що попередня книга була повернута) та виведення сумарної вартості виплат у випадку невчасного повернення книги, яка складається з вартості доставки повідомлення про заборгованість та накладеного штрафу.

Варіант 11.

1. Пакет акцій характеризується: прізвищем власника акцій, підприємством, яке випустило акції, кількістю акцій в пакеті, додатковими характеристиками. Підприємство, яке випустило акції, як окремий клас, характеризується назвою, статутним капіталом (млн. грн.) та містить метод обчислення рентабельності підприємства (в %), який залежить від розміру статутного капіталу (відповідне співвідношення придумати самостійно). Додаткові характеристики, як окремий клас, характеризуються номінальною вартістю однієї акції та містить метод обчислення вартості пакету акцій. Описати клас “Пакет акцій”, в якому передбачити метод обчислення можливих річних дивідендів за формулою (вартість пакету акцій * рентабельність / 100%). Використати екземпляр класу “Пакет акцій” для обчислення дивідендів певного пакету акцій.

Варіант 12.

1. Лікарські засоби характеризуються: виробником, складом, вартістю випуску. Виробник, як окремий клас, характеризується назвою та континентом розміщення. Склад, як окремий клас, характеризується вагою наявних наркотичних препаратів та містить метод обчислення додаткової вартості, яка залежить від ваги наркотичних препаратів (відповідне співвідношення придумати самостійно). В країні USA додаткова вартість не встановлюється. Описати клас „Лікарські засоби”, в якому передбачити метод обчислення загальної вартості виробництва лікарського засобу. Використати екземпляр класу „Лікарські засоби” для обчислення роздрібної вартості деякого лікарського засобу, на яку встановлюється 30 % надбавка до його вартості виробництва.

3.7. Основи об'єктно-орієнтованого програмування. Захист даних

Питання для обговорення.

1. Основні поняття об'єктно-орієнтованого програмування.
2. Поняття інкапсуляції.
3. Особливості реалізації інкапсуляції у мові Python. Захист даних.
4. Властивості. Геттери і сеттери.

Рекомендована література

1. Крєневич А.П. PYTHON у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Режим доступу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/01/python-u-prykladakh-i-zadachakh.-ch2-oor.pdf>
2. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. 180 с.
3. Python 3.8.0 documentation. URL: <https://docs.python.org/3/>.
4. Васильєв А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
5. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/>.
6. ООП на Python: концепции, принципы и примеры реализации. Режим доступу: <https://proglib.io/p/python-oor>

Онлайн середовища програмування мовою Python

1. https://www.onlinegdb.com/online_python_compiler
2. <https://www.programiz.com/python-programming/online-compiler/>

Практичні завдання

Приклади розв'язування задач

Завдання 1а. Описати клас „Добуток”, в якому передбачено обчислення добутку двох цілих чисел. Використати екземпляр даного класу для: а) обчислення добутку двох цілих чисел; б) обчислення факторіалу деякого числа.

Виконання.

Текст програми може бути таким:

```
class Mult:
    def __init__(self, x=0, y=0):
        self.__x=x # __ - префікс, який вказує, що
        поле захищене
        self.__y=y
```

```
# геттер - це метод, який повертає значення захищеного
поля (метод для зчитування)
# сеттер - метод, який перевіряє правильність даних,
# і у випадку правильності присвоює їх значення
захищеному полю (метод для запису)
```

```
# після опису геттера і сеттера із захищеним полем
# можна працювати як із звичайним,
# але програма буде взаємодіяти з ним через геттер та
сеттер
```

```
@property
def x(self):
    return self.__x

@x.setter
def x(self, x):
    self.__x = x

@property
def y(self):
    return self.__y

@y.setter
def y(self, y):
    self.__y = y

def mul(self):
    return self.x*self.y
```

```
x=int(input("Введіть перший множник: "))
y=int(input("Введіть другий множник: "))
a=Mult(x,y)
print("Добуток дорівнює ",a.mul())
n=int(input("Введіть число, факторіал якого треба
знайти: "))
a.x=1
for i in range(1,n+1):
    a.y=i
    a.x=a.mul()
print("Факторіал дорівнює",a.x)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

Завдання 1б. Описати клас „Мінімум”, в якому передбачено знаходження мінімального значення з двох чисел. Використати екземпляр даного класу для: а) знаходження мінімального значення з двох чисел; б) знаходження мінімального значення у п’ятиелементному векторі.

Виконання.

Текст програми може бути таким:

```
import random
class Min:
    def __init__(self, x=0):
        self.__min=x
    @property
    def min(self):
        return self.__min
# змінна a захищена від надання значення, більшого, ніж
# поточне
    @min.setter
    def min(self, a):
        if a<self.__min:
            self.__min=a

x=int(input("Введіть перше число: "))
a=Min(x)
y=int(input("Введіть друге число: "))
a.min=y
print(a.min)
print("Вектор: ")
b=[random.randint(-20,20) for i in range(5)]
print(b)
# b=[2,3,-1,1,8]
a=Min(b[0])
for i in b:
    a.min=i
print("Мінімальний елемент вектора: ",a.min)
```

Ввести програму, запустити її на виконання, перевірити роботу програми на самостійно складених контрольних прикладах.

3.8. Лабораторна робота № 12. Захист даних

Зауваження:

1. Інтерфейс програми має бути зрозумілим.
2. Кожне завдання має бути розв'язане в межах окремої програми.
3. При розв'язуванні задач повторювані блоки операторів оформити в вигляді підпрограм
4. При розв'язуванні обох задач описані класи мають бути інкапсульовані.
5. В класі першої задач, результати обчислень мають бути атрибутами.
6. В першій задачі розв'язання підзадач "а" та "б" мають бути здійсненні в межах різних програм.
7. В основній програмі розв'язання підзадачі "б" першої задачі екземпляр описаного класу має бути лише один (конструктор класу має викликатися лише один раз).
8. Другою задачею є доопрацювання першої задачі свого варіанту лабораторної роботи № 10 (Поліморфізм) або задачі свого варіанту лабораторної роботи „Реалізація відношення агрегації”, а саме інкапсульовання описаних класів.

Варіанти завдань для самостійного виконання

Варіант 1.

1. Описати клас „Добуток”, в якому передбачено обчислення добутку двох цілих чисел. Використати екземпляр цього класу для:
а) обчислення добутку двох цілих чисел; б) обчислення факторіалу деякого числа.

Варіант 2.

1. Описати клас „Функція”, в якому передбачено обчислення значення функції $f(x)=x+x^3+x^5+x^7$. Використати екземпляр цього класу для:
а) обчислення значення функції для деякого аргументу; б) обчислення значення виразу $\frac{1.7f(0.25)+2f(1+y)}{6-f(y^2-1)}$.

Варіант 3.

1. Описати клас „Скалярний добуток”, в якому передбачено обчислення скалярного добутку двох двохелементних векторів. Використати екземпляр цього класу для: а) обчислення скалярного добутку двох двохелементних векторів; б) обчислення $(a,b)+(b,c)$, де (p,q) – скалярний добуток векторів p і q .

Варіант 4.

1. Описати клас „Відрізок”, в якому передбачено обчислення відстані між двома точками, заданими координатами на декартовій площині. Використати екземпляр цього класу для: а) обчислення відстані між двома точками, заданими координатами на декартовій площині; б) обчислення периметра трикутника.

Варіант 5.

1. Описати клас „Максимум”, в якому передбачено знаходження максимального з двох чисел. Використати екземпляр цього класу для: а) знаходження максимального з двох чисел; б) знаходження найбільшого числа з заданих трьох.

Варіант 6.

1. Описати клас “Норма”, в якому передбачено обчислення норми радіус-вектора. Використати екземпляр цього класу для: а) обчислення норми радіус-вектора; б) знаходження суми довжин трьох радіус-векторів.

Варіант 7.

1. Описати клас „НСД”, в якому передбачено обчислення НСД (найбільшого спільного дільника) двох чисел. Використати екземпляр цього класу для: а) обчислення НСД двох чисел; б) обчислення НСД для трьох чисел $НСД(a, b, c)=НСД(НСД(a, b), c)$ та НСК (найбільше спільне кратне).

Варіант 8.

1. Описати клас „Функція”, в якому передбачено обчислення значення функції $h(a,b) = \frac{a}{1+b^2} + \frac{b}{1+a^2}$. Використати екземпляр цього класу для: а) обчислення значення функції для деякого аргументу;

б) обчислення значення виразу $h(s,t) + h(s-t,s \cdot t) + h(1,1)$.

Варіант 9.

1. Описати клас „Кратність”, в якому передбачено здійснення перевірки кратності одного числа іншому та обчислення залишку їх цілочисельного ділення. Використати екземпляр цього класу для:
а) перевірки кратності одного числа іншому та обчислення залишку їх цілочисельного ділення; б) знаходження кількості чисел, кратних 3, в заданому інтервалі та для знаходження добутків залишків від ділення на 3 чисел цього ж інтервалу, що не кратні 3.

Варіант 10.

1. Описати клас „Сума”, в якому передбачено обчислення суми двох комплексних чисел. Використати екземпляр цього класу для:
а) обчислення суми двох комплексних чисел; б) знаходження суми трьох комплексних чисел.

Варіант 11.

1. Описати клас „Мінімум”, в якому передбачено знаходження мінімального значення з двох чисел. Використати екземпляр цього класу для:
а) знаходження мінімального значення з двох чисел;
б) знаходження мінімального значення в 5-и елементному векторі.

Варіант 12.

1. Описати клас „Рівнобічна трапеція”, в якому передбачено обчислення площі трапеції за формулою $S = h \frac{a+b}{2}$, де a та b – основи трапеції, h – висота трапеції. Використати екземпляр цього класу для:
а) обчислення площі трапеції; б) обчислення загальної площі трьох трапецій з однаковими основами та трьома різними висотами.

Список використаних джерел

1. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Балакіна С.М., 2020. 180 с.
2. Python 3.8.0 documentation. URL: <https://docs.python.org/3/> (дата звернення: 12.11.2019).
3. Васильев А.Н. Python на примерах. Практический курс по программированию. 3-е издание. Москва: Наука и Техника, 2019. 432с.
4. Горошко Ю.В., Костюченко А.О., Теорія і методика розробки педагогічних програмних засобів. – Ч.: Виготовлення Єрмоленко О.М., 2011. - 144 с.
5. Дудик М.В., Рамський Ю.С., Цибко Г.Ю. Основи програмування: Навчальний посібник для студентів вищих навчальних закладів фізико-математичних та індустріально-педагогічних спеціальностей. – Умань: АЛМІ, 2009. – 162 с.
6. Кольцов Д.М. Python. Создаем программы и игры. Москва: Наука и Техника, 2017. 400с.
7. Кренивич А.П. Python у прикладах і задачах. Частина 1. Структурне програмування. Режим доступу: <http://176.102.48.88/ftp/disk-olymp/school-olymp/school-olymp2018-2019/Books/python/python2017.pdf>
8. Кренивич А.П. PYTHON у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Режим доступу: <http://www.mechmat.univ.kiev.ua/wp-content/uploads/2021/01/python-u-prykladakh-i-zadachakh.-ch2-ooop.pdf>
9. Майкл МакГрат. Программирование на Python для начинающих: [перевод с англ. Райтман М.А.]. Москва: Эксмо, 2015. 192 с.
10. Матвійчук С.В., Жуковський С.С. Практикум програмування Python / C++ на e-olymp.com (збірник задач з рекомендаціями до їх розв'язання). Житомир: Вид-во ЖДУ ім. І. Франка, 2019. 232 с.
11. Програмування на мові Python (3.x). Початковий курс. URL: <https://sites.google.com/site/pythonukr/> (дата звернення: 12.11.2019).
12. Роберт Седжвик, Кевин Уэйн, Роберт Дондеро. Программирование на языке Python. Учебный курс: [перевод с англ. Коваленко В.А.]. Москва: Эксмо, 2017. 736 с.
13. Томашевский П. Привет, Python! Моя первая книга по программированию. Москва: Наука и Техника, 2018. 256с.
14. Федоров Д.Ю. Основы программирования на примере языка Python : учеб. пособие. СПб.: Наука и Техника, 2016. 176 с.
15. Эл Свейгарт. Учим Python, делая крутые игры: [перевод с англ. Райтман М.А.]. Москва: Эксмо, 2018. 416 с.

Для нотаток

Навчальне видання

Програмування у Python. Практичний курс

Цибко Ганна Юхимівна – кандидат педагогічних наук, доцент, доцент кафедри інформатики і обчислювальної техніки Національного університету «Чернігівський колегіум» імені Т.Г.Шевченка

Горошко Юрій Васильович – доктор педагогічних наук, професор, професор кафедри інформатики і обчислювальної техніки Національного університету «Чернігівський колегіум» імені Т.Г.Шевченка

Костюченко Андрій Олександрович – кандидат педагогічних наук, старший викладач кафедри інформатики і обчислювальної техніки Національного університету «Чернігівський колегіум» імені Т.Г.Шевченка

Рецензенти:

Ткач Юлія Миколаївна - доктор педагогічних наук, професор, завідувач кафедри кібербезпеки та математичного моделювання Національного університету «Чернігівська політехніка»

Горчинський Сергій Володимирович - кандидат педагогічних наук, доцент кафедри технологічної освіти та інформатики Національного університету «Чернігівський колегіум» імені Т.Г.Шевченка

Підписано до друку __.__.2022р. Формат 60x84/16
Папір офсетний. Гарнітура Таймс. Друк на Ризографі.

Ум.друк.арк. __.

Тираж __ прим. Зам. № __.

Віддруковано в авторській редакції