

Міністерство освіти і науки України

Державний заклад  
«Південноукраїнський національний педагогічний університет  
імені К. Д. Ушинського»

Фізико-математичний факультет

Кобякова Людмила Миколаївна  
Мазурок Тетяна Леонідівна

Методичні рекомендації  
«Частина 1. Python: Керуючі конструкції»  
для самостійної підготовки і виконання лабораторних робіт  
навчального курсу ОК 10 «Інформатика»  
для здобувачів першого (бакалаврського) рівня вищої освіти  
1 року навчання спеціальності 014.09 Середня освіта (інформатика)

**УДК: 004.42+004.432**

Друкується за рішенням Вченої ради Державного закладу «Південноукраїнський національний педагогічний університет імені К.Д.Ушинського» (протокол № \_\_\_ від « » 2021 року).

Кобякова Л.М., Мазурок Т.Л. **Python: Керуючі конструкції**: Методичні рекомендації для самостійної підготовки і виконання лабораторних робіт навчального курсу ОК 10 «Інформатика» для здобувачів першого (бакалаврського) рівня вищої освіти 1-го року навчання спеціальності 014.09 Середня освіта (інформатика). Одеса: ПНПУ імені К.Д.Ушинського, 2021. 41 с.

**Рецензенти:**

**Брескіна Лада Валентинівна**, кандидат педагогічних наук, доцент, доцент кафедри прикладної математики та інформатики Південноукраїнського національного педагогічного університету імені К. Д. Ушинського.

**Сметаніна Людмила Сергіївна**, кандидат педагогічних наук, доцент, доцент кафедри електронного урядування та інформаційних систем Одеського регіонального інституту державного управління

Поданий у методичних рекомендаціях матеріал охоплює питання тем «Лінійний алгоритм», «Алгоритм розгалуження», «Циклічний алгоритм» мовою програмування Python, що формують уявлення про методику розробки програм, є фундаментом програмування і важливою ланкою професійної освіти для спеціалістів у галузі прикладної математики та інформатики.

Реалізація навчальних прикладів акцентує увагу на принципах побудови ефективних алгоритмів, способах опису вхідних і вихідних даних для вирішення прикладних задач.

## ЗМІСТ

НАВЧАЛЬНІ МАТЕРІАЛИ	5
Творець Python Гвидо ван Россум	5
Історія створення і розвитку мови програмування Python	5
Сайти з навчальними матеріалами. Уроки	5
Відеоматеріали. Відеоуроки	5
Самовчителі. Підручники	5
Довідники	5
Задачники, практикуми	5
Статті	6
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	7
ВСТУП	8
ТЕМА 1: ЛІНІЙНИЙ АЛГОРИТМ	9
ЛАБОРАТОРНА РОБОТА №1	9
<b>1.1.</b> Вступ до написання програм	9
<b>1.2.</b> Типи даних. Змінні	9
<b>1.3.</b> Спосіб розташування змінних у пам'яті	9
<b>1.4.</b> Введення с клавіатури	9
<b>1.5.</b> Модуль <code>math</code> (математичні константи і функції). Округлення.	10
Арифметичні операції й вирази	
<b>1.6.</b> Вивід на екран: форматний рядок	10
<b>1.7.</b> Розв'язання задач	11
<b>1.8.</b> Стандартний алгоритм обміну значеннями змінних	12
<b>1.9.</b> Задачі на написання програмного коду	12
<b>1.10.</b> Задачі на читання програмного коду	13
<b>1.11.</b> Відповіді до задач на читання програмного коду	16
<b>1.12.</b> Домашнє завдання	17
ТЕМА 2. АЛГОРИТМ РОЗГАЛУЖЕННЯ	18
ЛАБОРАТОРНА РОБОТА №2	18
<b>2.1.</b> Умовні вирази. Операції порівняння. Логічні операції	18
<b>2.2.</b> Операції в рядками	188
<b>2.3.</b> Умовні конструкції: умовний оператор, тернарний оператор	18
<b>2.4.</b> Розв'язання задач	18
<b>2.5.</b> Задачі на написання програмного коду	22
<b>2.6.</b> Задачі на читання програмного коду	23
<b>2.7.</b> Відповіді до задач на читання програмного коду	25
<b>2.8.</b> Домашнє завдання	26
ТЕМА 3. ЦИКЛІЧНИЙ АЛГОРИТМ	27
ЛАБОРАТОРНА РОБОТА №3	27

<b>3.1. Оператор циклу for. Оператори continue, break</b>	27
3.1.1. Розв'язання задач	27
3.1.2. Задачі на написання програмного коду	30
3.1.3. Задачі на читання програмного коду	31
3.1.4. Відповіді до задач на читання програмного коду	32
3.1.5. Домашнє завдання	34
ЛАБОРАТОРНА РОБОТА №4	34
<b>3.2. Оператор циклу while</b>	34
3.2.1. Розв'язання задач	34
3.2.2. Стандартний алгоритм Евкліда	36
3.2.3. Задачі на написання програмного коду	37
3.2.4. Задачі на читання програмного коду	38
3.2.5. Відповіді до задач на читання програмного коду	38
3.2.6. Домашнє завдання	39
ЛАБОРАТОРНА РОБОТА №5	39
<b>3.3. Вкладені цикли</b>	39
3.3.1. Розв'язання задач	39
3.3.2. Задачі на написання програмного коду	41
3.3.3. Домашнє завдання	41

## НАВЧАЛЬНІ МАТЕРІАЛИ

Творець Python Гвидо ван Россуме

[https://uk.wikipedia.org/wiki/Гвідо\\_ван\\_Россум](https://uk.wikipedia.org/wiki/Гвідо_ван_Россум)

Історія створення та розвитку мови програмування Python

<https://ru.wikipedia.org/wiki/Python>

**Сайти з навчальними матеріалами. Уроки**

<https://metanit.com/python/tutorial/>

<https://pythonworld.ru/samouchitel-python>

**Відеоматеріали. Відеоуроки**

Канал Тимофія Хір'янова на YouTube

<https://www.youtube.com/playlist?list=PLRDzFCPr95fIDJUvFхvzWxg-V9BmZlMMe>

**Самовчителі. Підручники:**

1. Любанович Билл. Простой Python. Современный стиль программирования. СПб., Питер, 2021. 592 с.
2. Лутц Марк. Изучаем Python. М., Диалектика, 2019-2020.
3. Дауни Аллен. Основы Python. Научитесь думать как программист. М., Манн, Иванов и Фербер, 2021. 304 с.
4. Седжвик Роберт, Уэйн Кевин, Дондеро Роберт. Программирование на языке Python: учебн. курс. СПб.: ООО «Альфа-книга», 2017. 736 с.

**Довідники**

Мартелли Алекс, Рейвенскрофт Анна, Холден Стив. Python. Справочник. Полное описание языка. СПб., Диалектика, 2019. 896 с.

**Задачники, практикуми**

1. Абрамов С.А и др. Задачи по программированию. Вологда, 2000. 596с.  
<https://ideafix.name/wp-content/uploads/stuff/book95.pdf>
2. Абрамян М.Э. 1000 задач по программированию. Ростов-на-Дону, 2004.  
<http://www.ptaskbook.com/ru/>
3. Електронний підручник Рішельєвського ліцею (ЕПРЛ)  
<http://rlwestudy.blogspot.com/p/powerpoint.html>
4. Жуков Р.А. Язык программирования Python: практикум: учеб. пос. М.: ИНФРА-М, 2019. 216с.  
[https://codernet.ru/books/python/yazyk\\_programmirovaniya\\_python\\_praktikum/](https://codernet.ru/books/python/yazyk_programmirovaniya_python_praktikum/)
5. Златопольский Д. 1400 задач по программированию. М., ДМК Пресс, 2020. 192 с.  
<https://coderoll.net/dev-book/200-1400-zadach-po->

[programirovaniyu.html](#)

### **Статті**

1) Компілятор

<https://uk.wikipedia.org/wiki/Компілятор>

2) Інтерпретатор

<https://uk.wikipedia.org/wiki/Інтерпретатор>

3) Поняття алгоритму

[https://ru.wikibooks.org/wiki/Что\\_такое\\_алгоритм](https://ru.wikibooks.org/wiki/Что_такое_алгоритм)

4) Блок-схема

<https://uk.wikipedia.org/wiki/Блок-схема>

5) Нескінченний цикл

[https://ru.wikipedia.org/wiki/Бесконечный\\_цикл](https://ru.wikipedia.org/wiki/Бесконечный_цикл)

6) Що таке goto, використовувати його чи ні?

<https://ru.wikipedia.org/wiki/Goto>

7) Спагетті-код

<https://ru.wikipedia.org/wiki/Спагетти-код>

8) Спагетті-стаття про спагетті-код

<https://habr.com/ru/post/187154/>

9) Онлайн-інтерпретатори та онлайн-компілятори Python

<https://dev-gang.ru/article/7-luczshih-perevodczikov-python-obnovleno-2019-auvtihzcfi/>

## ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

### 1. онлайн-компілятори

1) **ideone**

[ideone.com](https://ideone.com)

обрати Python 3

2) **replit**

<https://replit.com/languages/python3>

### 2. MS Visual Studio

Інструкція зі встановлення

<https://metanit.com/python/tutorial/1.4.php>

### 3. Anaconda

Інструкція зі встановлення

<https://www.machinelearningmastery.ru/guide-to-installing-anaconda-in-windows-10-93880eca89b4/>

<https://pythonru.com/baza-znaniy/kak-ustanovit-anaconda-na-windows>

<https://python.ivan-shamaev.ru/guide-conda-environments-anaconda-python-data-science-platform/>

## ВСТУП

Навчальна дисципліна ОК 10 «Інформатика» для здобувачів першого (бакалаврського) рівня вищої освіти 1-го року навчання спеціальності 014.09 Середня освіта (інформатика) за навчальним планом підготовки фахівця охоплює 12 кредитів загальною кількістю 360 годин. Аудиторні заняття – 180 годин (лекції – 36 годин, лабораторні роботи — 144 години), 180 годин відведено на самостійну роботу студентів. Зміст навчальної дисципліни розділено за програмою на 2 модуля, перший присвячено основам інформатики, другий охоплює питання процедурного програмування мовою програмування C++ та/або Python на вибір студента. Вибір студентом мови програмування дозволяє реалізувати індивідуальний підхід до навчання програмування.

Методичні рекомендації «Python: Керуючі конструкції:» розкривають питання тем «Лінійний алгоритм», «Алгоритм розгалуження», «Циклічний алгоритм» навчальної програми другого модуля навчальної дисципліни ОК 10 «Інформатика». За програмою навчальної дисципліни заняття розраховані на 30 годин аудиторної і 30 годин самостійної роботи.

Зміст не потребує додаткових базових знань, частково спирається на шкільний курс математики.

Розроблені лабораторні роботи є основою для засвоєння процедурного програмування, а в подальшому – об'єктно-орієнтованого програмування та практичним підґрунтям для дисциплін комп'ютерного циклу.



## Тема 1. ЛІНІЙНИЙ АЛГОРИТМ

### ЛАБОРАТОРНА РОБОТА №1

- 1.1. Вступ до написання програм
- 1.2. Типи даних. Змінні
- 1.3. Спосіб розташування змінних у пам'яті
- 1.4. Введення с клавіатури
- 1.5. Модуль `math` (математичні константи і функції). Округлення. Арифметичні операції й вирази
- 1.6. Вивід на екран: форматний рядок
- 1.7. Розв'язання задач
- 1.8. Стандартний алгоритм обміну значеннями змінних
- 1.9. Задачі на написання програмного коду
- 1.10. Задачі на читання програмного коду
- 1.11. Відповіді до задач на читання програмного кода
- 1.12. Домашнє завдання

#### ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

- 1.1. Вступ до написання програм

<https://metanit.com/python/tutorial/2.1.php>

- 1.2. Типи даних. Змінні

<https://metanit.com/python/tutorial/2.2.php>

[Лобанович], Таблиця 2.1, С.57.

- 1.3. Спосіб розташування змінних у пам'яті

[Седжвик], С.31-38

- 1.4. Введення значень з клавіатури

<https://foxford.ru/wiki/informatika/vvod-dannyh-v-python>

<https://younglinux.info/python/input>

#### Приклади

№	Програмний код	На екрані / Після виконання програмного коду
1	<pre>a = input() print( type(a) )</pre>	Програма зупиняється і чекає, що користувач введе з клавіатури значення змінної <code>a</code> Вивід на екран типу змінної <code>a</code> На екрані: <code>&lt;class 'str'&gt;</code>
2	<pre>b = int( input() ) print( type(b) )</pre>	Введене користувачем значення (рядкового типу) перетворюється (якщо це можливо) до цілого типу ( <code>int</code> ) та присвою-

		ється змінній <code>b</code> Вивід на екран типу змінної <code>b</code> На екрані: <code>&lt;class 'int'&gt;</code>
3	<pre>c = float( input() ) print( type(c) )</pre>	Значення, введене з клавіатури, перетворюється до типу <code>float</code> та присвоюється змінній <code>c</code> Вивід на екран типу змінної <code>c</code> На екрані: <code>&lt;class 'float'&gt;</code>
4	<pre>name = input( "Як Вас звати? " )</pre>	На екрані запрошення Як Вас звати? Користувач вводить, наприклад, Вася Змінній <code>name</code> типу <code>str</code> присвоєно значення Вася
5	<pre>d = int(input("Введіть ціле число: "))</pre>	На екрані Введіть ціле число: Користувач вводить, наприклад, 123 (типа <code>str</code> ) Змінній <code>d</code> присвоєно значення 123 типа <code>int</code>
6	<pre>r = float( input("r = " ) )</pre>	На екрані <code>r =</code> Користувачем здійснюється ввід 12.567 Змінна <code>r</code> ініціалізована значенням 12.567 типу <code>float</code>

### 1.5. Модуль `math` (математичні константи та функції). Округлення.

Арифметичні операції, функції та вирази

<https://metanit.com/python/tutorial/2.3.php>

Округлення:

<https://pythonworld.ru/osnovy/okruglenie.html>

Модуль `math`: Математичні функції та константи  $\pi$ ,  $e$  :

[Жуков], С.58-60

<https://metanit.com/python/tutorial/6.2.php>

<https://pythonchik.ru/matematika/modul-math-v-python>

### 1.6. Вивід на екран: форматний рядок

<https://younglinux.info/python/input>

[Жуков], п.2.1.5, С.55-57

### 1.7. Розв'язання задач

1. Дано 2 цілих числа. Знайти їх суму, різницю, добуток, результат ці-

лочисельного ділення, залишок від ділення, частку від звичайного ділення першого числа на друге.

Алгоритм:

- 1) завдання значень змінних  $a$  і  $b$ ;
- 2) змінна  $sum$  – сума чисел  $a$  і  $b$ , вивід значення змінної на екран;
- 3) змінна  $difference$  – різниця чисел  $a$  і  $b$ , вивід на екран;
- 4) змінна  $product$  – добуток чисел  $a$  і  $b$ , вивід на екран;
- 5) змінна  $int\_div$  – частка цілочисельного ділення чисел  $a$  і  $b$ , вивід на екран;
- 6) змінна  $rem\_div$  – залишок від цілочисельного ділення чисел  $a$  і  $b$ , вивід на екран;
- 7) змінна  $division$  – обчислення частки чисел  $a$  і  $b$ , вивід на екран.

	Програмний код
1	<code>a = 100</code>
2	<code>b = 15</code>
3	
4	<code>summa = a + b</code>
5	<code>print("%4d + %4d = %4d" % (a, b, summa) )</code>
6	
7	<code>difference = a - b</code>
8	<code>print("%4d - %4d = %4d" % (a, b, difference) )</code>
9	
10	<code>product = a * b</code>
11	<code>print("%4d * %4d = %4d" % (a, b, product) )</code>
12	
13	<code>int_div = a // b</code>
14	<code>print("%4d // %4d = %4d" % (a, b, int_div) )</code>
15	
16	<code>rem_div = a % b</code>
17	<code>print("%4d %% %4d = %4d" % (a, b, rem_div) )</code>
18	
19	<code>division = a / b</code>
20	<code>print("%4d / %4d = %6.2f" % (a, b, division) )</code>

На екрані:

```

100 +   15 =  115
100 -   15 =   85
100 *   15 = 1500
100 //  15 =    6
100 %   15 =   10
100 /   15 =  6.67

```

**2.** Дано 2 сторони трикутника та кут (у градусах) між ними. Знайти протилежну сторону.

Алгоритм:

- 1) підключити бібліотеку математичних функцій `math`
- 2) ініціалізувати вихідні дані – сторони  $a$ ,  $b$  та  $\gamma$  – кут між ними (у

градусах)

3) перевести кут з градусної міри в радіанну, використати функцію `radians()` бібліотеки `math`

$$\gamma' = \gamma \cdot \frac{\pi}{180^\circ}, \quad \gamma' - \text{кут у радіанах, } \gamma - \text{кут в градусах}$$

`pi` – стандартна константа  $\pi$  з бібліотеки `math`

4) обчислити за теоремою косинусів 3-ю сторону трикутника `c` та вивести її на екран

$$c = \sqrt{a^2 + b^2 - 2ab \cos \gamma}$$

Програмний код	
1	<code>import math</code>
2	
3	<code>a = 10</code>
4	<code>b = 7</code>
5	<code>gamma = 60 # градусів</code>
6	
7	<code>gammaR = math.radians(gamma)</code>
8	<code>c = math.sqrt(a**2 + b**2 - 2*a*b*math.cos(gammaR) )</code>
9	
10	<code>print("c = ", c)</code>

На екрані:

```
c = 8.888194417315589
```

### 1.8. Стандартний алгоритм обміну значеннями змінних

Дано 2 змінних `a = 100` и `b = 1`.

Здійснити обмін значеннями змінних, тобто. в результаті: `a = 1`, `b = 100`.

Невірно		Вірно	
Програмний код	Результат виконання програмного коду	Програмний код	Результат виконання програмного коду
<code>b = a</code> <code>a = b</code>	<code>b = 100</code> <code>a = 100</code>	<u>1 спосіб</u> <code>temp = a</code> <code>a = b</code> <code>b = temp</code>	<code>temp = 100</code> <code>a = 1</code> <code>b = 100</code>
		<u>2 спосіб</u> <code>a, b = b, a</code>	<code>a = 1, b = 100</code>

`temp` – допоміжна змінна (3-я каструлька при обміні вмісту 2 каструлек ☺).

### 1.9. Задачі на написання програмного коду

1. [Абрамян, Вегін8-9]. Дано 3 цілих числа `a`, `b`, `c`. Знайти їх середнє арифметичне та середнє геометричне (квадратний корінь їх добутку).

2. [Абрамян, Вегін11-12]. Дано 3 дійсних числа. Знайти

– різницю модулів та добуток їх квадратів;

– модуль їх суми та квадрат їх частки.

3. [Абрамян, Begin24]. Дано змінні  $a$ ,  $b$ ,  $c$ . Змінити їх значення, перемістивши вміст  $a$  в  $c$ ,  $c$  — в  $b$ ,  $b$  — в  $a$ , та вивести вихідні і нові значення змінних  $a$ ,  $b$ ,  $c$  на екран.

4. [Абрамян, Begin28]. Дано число  $a$ . Обчислити  $a^{15}$ , використовуючи дві допоміжні змінні та 5 операцій множення. Для цього послідовно знаходити  $a^2$ ,  $a^3$ ,  $a^5$ ,  $a^{10}$ ,  $a^{15}$ . Вивести всі знайдені ступені числа  $a$ .

5. [Абрамян, Integer3]. Дано розмір файлу в байтах. Використовуючи операцію ділення націло, знайти кількість повних кілобайтів, які займає даний файл (1 кілобайт = 1024 байта).

6. [Абрамян, Integer11]. Дано тризначне число. Знайти суму та додатак його цифр.

7. [Абрамян, Integer19-23]. З начала доби пройшло  $N$  секунд ( $N$  — ціле). Знайти

- кількість повних хвилин, які минули від начала доби;
- кількість повних годин, які минули від начала доби;
- кількість секунд, які минули від начала останньої хвилини;
- кількість секунд, які минули від начала останньої години;
- кількість повних хвилин, які минули від начала останньої години.

8. [Абрамян, Integer24]. Дні тижня пронумеровані наступним чином: 0 — неділя, 1 — понеділок, 2 — вівторок, ..., 6 — субота. Дано ціле число  $K$ , яке належить діапазону 1–365. Визначити номер дня тижня для  $K$ -го дня року, якщо відомо, що в цьому році 1 січня було днем з номером  $N$ ,  $N$  введіть з клавіатури.

9. Дано дійсні та уявні частини 2-х комплексних чисел. Знайти дійсні та уявні частини їх суми, добутку та частки.

10. Дано гіпотенузу та гострий кут у градусах прямокутного трикутника. Знайти його катети.

11. Дано катет та його проекція на гіпотенузу. Знайти гіпотенузу, висоту, другий катет та його проекцію на гіпотенузу.

12. Дано перший член  $a_1$ , різниця  $d$  й кількість  $n$  перших членів арифметичної прогресії. Обчисліть  $n$ -ий член  $a_n$  та суму  $n$  перших членів арифметичної прогресії  $s_n$ .

13. Послідовність Фібоначчі визначається формулами:  $a_0 = 0$ ,  $a_1 = 1$ ,  $a_k = a_{k-1} + a_{k-2}$  при  $k \geq 2$ . Обчисліть  $a_2$ ,  $a_3$ ,  $a_4$ .

### 1.10. Задачі на читання програмного коду

<b>1.</b>	<pre># 1. Вихідні дані a = 10.1 b = 5.3  # 2. Обчислювальна частина # 3. Вивід результатів на</pre>	<b>2.</b>	<pre>import math  # 1. Вихідні дані d = 10.  # 2. Обчислювальна частина</pre>
-----------	---	-----------	---

	<pre> екран p = 2 * (a + b) print('p = {0}'.format(p))  s = a * b print('s = {0}'.format(s)) </pre>		<pre> # 3. Вивід результатів на екран c = math.pi * d print("c = %5.2f" % (c) )  r = d / 2 s = math.pi * r**2 print("s = %5.2f" % (s) ) </pre>
<b>3.</b>	<pre> import math  # 1. Вихідні дані a = 3. b = 4.  # 2. Обчислювальна частина c = math.sqrt(a*a + b*b) p = (a + b + c) / 2 s = math.sqrt( p * (p-a) * (p-b) * (p-c) )  # 3. Вивід результатів на екран print("c = ", c) print("s = ", s) </pre>	<b>4.</b>	<pre> import math  # 1. Вихідні дані Ax = 0; Ay = 0 Bx = 0; By = 1 Cx = 1; Cy = 0  # 2. Обчислювальна частина a = math.sqrt( (Bx - Ax)**2 + (By - Ay)**2 ) b = math.sqrt( (Bx - Cx)**2 + (By - Cy)**2 ) c = math.sqrt( (Cx - Ax)**2 + (Cy - Ay)**2 )  # 3. Вивід результатів на екран print("a = ", a) print("b = ", b) print("c = ", c) </pre>
<b>5.</b>	<pre> # 1. Вихідні дані a = 100 b = 1 print("a = ", a, "b = ", b)  # 2. Обчислювальна частина temp = a a = b b = temp  # 3. Вивід результатів на екран print("a = ", a, "b = ", b) </pre>	<b>6.</b>	<pre> # 1. Вихідні дані a = 100 b = 10 c = 1 print("a = ", a, "b = ", b, "c = ", c)  # 2. Обчислювальна частина a, b, c = c, a, b  # 3. Вивід результатів на екран print("a = ", a, "b = ", b, "c = ", c) </pre>
<b>7.</b>	<pre> # 1. Вихідні дані a = 10 print("a = ", a)  # 2. Обчислювальна частина a *= a a *= a a *= a  # 3. Вивід результатів на екран print("a = ", a) </pre>	<b>8.</b>	<pre> # 1. Вихідні дані a = 15 print("a = ", a)  # 2. Обчислювальна частина c1 = a // 10 c2 = a % 10 b = c2 * 10 + c1  # 3. Вивід результатів на екран print("b = ", b) </pre>

9.	<pre># 1. Вихідні дані Ax = 1;  Ay = 2 Cx = 5;  Cy = 6  # 2. Обчислювальна частина Vx = Ax;  Vy = Cy Dx = Cx;  Dy = Ay  # 3. Вивід результатів на екран print("A(",  Ax,  ",",  Ay,       ")") print("B(",  Vx,  ",",  Vy,       ")") print("C(",  Cx,  ",",  Cy,       ")") print("D(",  Dx,  ",",  Dy,       ")")</pre>	10.	<pre>import math  # 1. Вихідні дані alpha = 2*math.pi/3  # 2. Обчислювальна частина s = 2 * math.sin(alpha) * math.cos(alpha) c = 1 - 2 * s**2  # 3. Вивід результатів на екран print("s = ", s) print("c = ", c)</pre>
11.	<pre>import math  # 1. Вихідні дані a = 3. b = 4. c = 5.  # 2. Обчислювальна частина angR = math.acos( (a*a + b*b - c*c) / (2*a*b) ) angG = math.degrees( angR )  # 3. Вивід результатів на екран print("angG = ", angG)</pre>	12.	<pre># 1. Вихідні дані v = 20. v0 = 3. t1 = 2 + 25/60 t2 = 3 + 15/60  # 2. Обчислювальна частина s1 = (v + v0) * t1 s2 = (v - v0) * t2 s = s1 + s2  # 3. Вивід результатів на екран print("s = ", s)</pre>
13.	<pre># 1. Вихідні дані b1 = float(input("input b1: ")) q = float(input("input q : ")) n = int(input("input n : "))  # 2. Обчислювальна частина bn = b1 * q**(n-1) sn = b1 * (1 - q**n)/(1 - q)  # 3. Вивід результатів на екран print("bn = ", bn) print("sn = ", sn)</pre>	14.	<pre># 1. Вихідні дані ch1 = 2; zn1 = 5; ch2 = 4; zn2 = 5;  # 2. Обчислювальна частина ch3 = ch1 * zn2 + ch2 * zn1 zn3 = zn1 * zn2 ch4 = ch1 * ch2 zn4 = zn1 * zn2 ch5 = ch1 * zn2 zn5 = ch2 * zn1  # 3. Вивід результатів на екран print("ch3/zn3 = {0}/{1}".format(ch3, zn3)) print("ch4/zn4 = {0}/{1}".format(ch4, zn4)) print("ch5/zn5 = {0}/{1}".format(ch5, zn5))</pre>
15.	# 1. Обчислювальна частина	16.	import math

	<pre> a = math.sqrt(2) b = math.pow(3, 1/3) c = math.pow(4, 1/4)  # 2. Вивід результатів на екран print("a = ", a) print("b = ", b) print("c = ", c) </pre>		<pre> # 1. Вихідні дані a = 2.5  # 2. Обчислювальна частина a1 = math.floor(a) a2 = math.ceil(a) a3 = round(a) a4 = math.trunc(a)  # 3. Вивід результатів на екран print("a = ", a) print("a1 = ", a1) print("a2 = ", a2) print("a3 = ", a3) print("a4 = ", a4) </pre>
<b>17.</b>	<pre> import math  # 1. Вихідні дані Rez = 1. Imz = 1.  # 2. Обчислювальна частина r = math.sqrt( Rez**2 + Imz**2 ) argz = math.atan( Imz/Rez )  # 3. Вивід результатів на екран print("r = ", r) print("argz = ", argz) </pre>	<b>18.</b>	<pre> # 1. Вихідні дані a = 4.5 b = 4 c = 3  # 2. Обчислювальна частина s1 = a + b s2 = int(a + b) s3 = int(a) + b s4 = b // c s5 = b / c  # 3. Вивід результатів на екран print("s1 = ", s1) print("s2 = ", s2) print("s3 = ", s3) print("s4 = ", s4) print("s5 = ", s5) </pre>
<b>19.</b>	<pre> # 1. Вихідні дані a = 100 b = 1  # 2. Обчислювальна частина a = a + b b = a - b a = a - b  # 3. Вивід результатів на екран print("a = {0}, b = {1}".format(a, b)) </pre>	<b>20.</b>	<pre> # 1. Вихідні дані a = 100 b = 10 c = 1  # 2. Обчислювальна частина s = a + b + c a = s - a b = s - b c = s - c  # 3. Вивід результатів на екран print("a = {0}, b = {1}, c = {2}".format(a, b, c)) </pre>

### 1.11. Відповіді до задач на читання програмного коду

1. Дано сторони прямокутника, знайти його периметр та площину, або



дано 2 числа, знайти їх подвоєну суму та добуток.

2. Дано діаметр кола. Знайти його довжину, радіус і площину.
3. Дано катети прямокутного трикутника. Знайти його гіпотенузу та площину за формулою Герона.
4. Дано координати вершин трикутника. Знайти його сторони.
5. Обмін значеннями змінних із використанням допоміжної змінної.
6. Дано 3 числа  $a$ ,  $b$ ,  $c$ . Здійснити зсув значень вправо, тобто.  

$$a = c, b = a, c = b.$$
7. Дано число  $a$ . За 3 дії знайти  $a^8$ .
8. Дано 2-значне число. Обчислити число, цифри якого розташовані в зворотному порядку. Наприклад,  $a = 12$ ,  $b = 21$ .
9. Дано координати кінців діагоналі прямокутника, сторони якого паралельні осям. Обчислити координати кінців 2-ї діагоналі.
10. Дано кут в радіанах. Обчислити косинус та синус подвоєного кута.
11. Дано сторони трикутника  $a$ ,  $b$ ,  $c$ . Обчислити кут (в градусах), протилежний стороні  $c$ .
12. Дано власна швидкість транспортного засобу, швидкість течії річки, час руху по та проти течії. Знайти пройдений шлях.
13. Дано 1-й член  $b_1$ , знаменник  $q$  та  $n$  – кількість членів геометричної прогресії. Знайти  $n$ -й член та суму  $n$  перших членів геометричної прогресії.
14. Дано чисельники та знаменники 2-х дробів. Знайти їх суму, добуток тв. частку.
15. Знайти  $\sqrt{2}$ ,  $\sqrt[3]{3}$ ,  $\sqrt[4]{4}$ .
16. Дано число. Знайти результати округлення до меншого цілого, більшого цілого, згідно правил банківського округлення та відкидання дробової частини.
17. Дано дійсна та уявна частини комплексного числа. Знайти його модуль та аргумент.
18. Дано 1 дійсне число  $a$  та 2 цілих –  $b$  г  $c$ . Знайти результати наступних дій:  $a + b$ ,  $\text{int}(a + b)$ ,  $\text{int}(a) + b$ ,  $b // c$ ,  $b / c$ .
19. Обмін значеннями змінних без використання допоміжної змінної.
20. Дано числа  $a$ ,  $b$ ,  $c$ . Замінити кожне з них сумою 2-х залишившихся.

### 1.12. Домашнє завдання

[ ЕПРЛ ], Тема 2 «Лінійні програми»

п.2.10. Задачі на цілочисельне ділення

п.2.18. Задачі на відсотки

Варіанти згідно списку групи.

## Тема 2. АЛГОРИТМ РОЗГАЛУЖЕННЯ

### ЛАБОРАТОРНА РОБОТА №2

- 2.1. Умовні вирази. Операції порівняння. Логічні операції
- 2.2. Операції в рядках
- 2.3. Умовні конструкції: умовний оператор, тернарний оператор
- 2.4. Розв'язання задач
- 2.5. Задачі на написання програмного коду
- 2.6. Задачі на читання програмного коду
- 2.7. Відповіді до задач на читання програмного коду
- 2.8. Домашнє завдання

#### ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

- 2.1. Умовні вирази. Операції порівняння. Логічні операції

<https://metanit.com/python/tutorial/2.4.php>

- 2.2. Операції з рядками

<https://metanit.com/python/tutorial/2.5.php>

- 2.3. Умовні конструкції: умовний оператор, тернарний оператор

<https://metanit.com/python/tutorial/2.6.php>

<http://www.rupython.com/python-40-277.html>

<https://codecamp.ru/blog/python-conditionals/>

- 2.4. Розв'язання задач

1. [Абрамян, Boolean34-36]. Дано координати 2-х різних полів шахівниці  $x_1, y_1, x_2, y_2$  (цілі числа, які належать діапазону 1–8). Враховуючи, що ліве нижнє поле шахівниці (1, 1) чорне, перевірити істинність висловлювань:

поле  $(x_1, y_1)$  біле;

поле  $(x_2, y_2)$  чорне;

поля однакового кольору.

тура за один хід може перейти з першого поля на друге.

Алгоритм:

1) вивід на екран повідомлення «Введіть  $x$  та  $y$  координати 1-го поля шахівниці» та введення з клавіатури значень змінних  $x_1, y_1$ ;

2) вивід на екран повідомлення «Введіть  $x$  та  $y$  координати 2-го поля шахівниці» та введення з клавіатури значень змінних  $x_2, y_2$ ;

3) обчислення значень булевих змінних `white1` та `black2`

Клітка біла, якщо  $x$ -ва та  $y$ -ва координати клітки різної парності, чорна, якщо однакової.

Якщо число  $a$  парне, залишок від цілочисельного ділення на 2 дорівнює 0 ( $a \% 2 == 0$ ), якщо непарне – 1 ( $a \% 2 == 1$ ),

4) обчислення булевої змінної `colors_equals`

`colors_equals = true`, якщо змінні `white1` та `black2` мають різні логічні значення (одна з них `True`, друга – `False`);

#### 5) обчислення булевої змінної `tura`

Тура ходить вертикально, тоді `x`-ві координати кліток співпадають (`x1 == x2`), або горизонтально, тоді співпадають `y`-ві (`y1 == y2`).

	Програмний код
1	<code>print("Введіть x та y координати 1-го поля шахівниці")</code>
2	<code>x1 = int( input() )</code>
3	<code>y1 = int( input() )</code>
4	
5	<code>print("Введіть x та y координати 2-го поля шахівниці")</code>
6	<code>x2 = int( input() )</code>
7	<code>y2 = int( input() )</code>
8	
9	<code>white1 = (x1 % 2 == 0 and y1 % 2 == 1) or (x1 % 2 == 1 and y1</code>
10	<code>% 2 == 0)</code>
11	<code>print("1-а клітка біла? ", white1 )</code>
12	<code>black2 = (x2 % 2 == 0 and y2 % 2 == 0) or (x2 % 2 == 1 and y2</code>
13	<code>% 2 == 1)</code>
14	<code>print("2-а клітка чорна? ", black2 )</code>
15	
16	<code>color_equals = (white1 != black2)</code>
17	<code>print("Кольори 2-х кліток співпадають? ", color_equals)</code>
18	
19	<code>tura = (x1 == x2 or y1 == y2)</code>
20	<code>print("Чи може тура за один хід дійти із 1-ої клітки до 2-ї?</code>
21	<code>", tura)</code>

На екрані:

```

Введіть x та y координати 1-го поля шахівниці
1 2
Введіть x та y координати 2-го поля шахівниці
1 8
0 - False, 1 - True
1-я клітка біла?      1
2-я клітка чорна?    0
Кольори 2-х кліток співпадають?      1
Чи може тура за один хід дійти з 1-ої клітки у 2-у?      1

```

## 2. Дано 2 дійсних числа `a`, `b`. Знайти їх максимум

### Алгоритм 1 – повний умовний оператор

- 1) Ініціалізувати дійсні змінні `a`, `b`
- 2) Використовуючи повний умовний оператор, записати умову  
якщо `a > b`,  
то `max = a`,  
інакше `max = b`;

- 3) Вивести на екран значення змінної `max`

### Алгоритм 2 – неповний умовний оператор

- 1) Задати значення дійсних змінних  $a$ ,  $b$
- 2) Присвоїти дійсній змінній  $max$  значення  $a$ ;
- 3) Знаходження максимуму 2-х чисел. Використовуючи неповний умовний оператор, записати умову

якщо  $max < b$ ,  
то  $max = b$

- 4) Вивести на екран значення змінної  $max$

### Алгоритм 3 – тернарний оператор

- 1) Задати значення дійсних змінних  $a$ ,  $b$
- 2) Застосовуючи тернарний оператор, обчислити значення дійсної змінної  $max$ , тобто: у випадку виконання умови ( $a > b$ ) покласти її значення рівним  $a$ , невиконання –  $b$ .
- 3) Вивести на екран значення змінної  $max$

Алгоритм 1 – Програмний код повний умовний оператор	
1	<code>import math</code>
2	
3	<code># 1. Вихідні дані</code>
4	<code>a = 100</code>
5	<code>b = 10</code>
6	
7	<code># 2. Обчислювальна частина</code>
8	<code>if a &gt; b :</code>
9	<code>    max = a</code>
10	<code>else :</code>
11	<code>    max = b</code>
12	
13	<code># 3. Вивід результатів на екран</code>
14	<code>print("a = {0}, b = {1}, max = {2}".format(a, b, max))</code>

Алгоритм 2 – Програмний код неповний умовний оператор	
1	<code># 1. Вихідні дані</code>
2	<code>a = 100</code>
3	<code>b = 10</code>
4	
5	<code># 2. Обчислювальна частина</code>
6	<code>max = a</code>
7	<code>if a &lt; b :</code>
8	<code>    max = b</code>
9	
10	<code># 3. Вивід результатів на екран</code>
11	<code>print("a = {0}, b = {1}, max = {2}".format(a, b, max))</code>

Алгоритм 3 – Програмний код тернарний оператор	
	<code># 1. Вихідні дані</code>

<pre> a = 100 b = 1  # 2. Обчислювальна частина <b>max = a if a &gt; b else b</b>  # 3. Вивід результатів на екран print("a = {0}, b = {1}, max = {2}".format(a, b, max)) </pre>
--

На екрані:

```
max = 100
```

**3.** Ввести з клавіатури номер місяця (ціле число). Вивести на екран назву пори року. Якщо введене число не може бути номером місяця, вивести на екран повідомлення «Введено неприпустиме значення змінної month».

#### Алгоритм

1) Вивести на екран запрошення «Введіть номер місяця»

Ввести з клавіатури значення змінної month

2) Використовуючи умовний оператор, записати

якщо month < 1 або month > 12,

то вивести на екран повідомлення «Введено неприпустиме значення змінної month»,

інакше якщо month = 1 або month = 2 або month = 12,

то вивести на екран повідомлення «Зима»,

інакше якщо month >= 3 та month <= 5,

то вивести на екран повідомлення «Весна»,

інакше якщо month >= 6 та month <= 8,

то вивести на екран повідомлення «Літо»,

інакше вивести на екран повідомлення «Осінь»,

Алгоритм – Програмний код	
1	# 1. Вихідні дані
2	month = int( input("Введіть номер місяця") )
3	
4	# 2. Обчислювальна частина та Вивід результатів на екран
5	<b>if</b> month < 1 or month > 12 :
6	print("\n Введено неприпустиме значення змінної month")
7	<b>elif</b> month == 1 or month == 2 or month == 12 :
8	print("\n Зима")
9	<b>elif</b> month >= 3 and month <= 5 :
10	print("\n Весна")
11	<b>elif</b> month >= 6 and month <= 8 :
12	print("\n Літо")
13	<b>else :</b>
14	print("\n Осінь")

На екрані:

```
Введіть номер місяця
```

```
2
```

Зима

## 2.5. Задачі на написання програмного коду

### 2.5.1. Неповний умовний оператор

1. [Абрамян, If1]. Дано ціле число. Якщо воно додаткове, то додати до нього 1; в іншому випадку не змінювати його. Ввести отримане число на екран.

2. [Абрамян, If4]. Дано три цілих числа. Знайти кількість додаткових чисел у вихідному наборі.

3. [Абрамян, If12]. Дано три числа. Знайти найменше з них.

4. [Абрамян, If20]. На числовій осі розташовані три точки: А, В, С. З'ясувати, яка з двох останніх точок (В або С) розташовані ближче до А, й вивести цю точку й відстань від неї до точки А.

### 2.5.2. Повний умовний оператор

1. [Абрамян, If2]. Дано ціле число. Якщо воно додаткове, то додати до нього 1; в іншому випадку – відняти від нього 2. Вивести отримане число.

2. [Абрамян, If8]. Дано два нерівних числа. Вивести першим більше з них, а потім – менше.

3. [Абрамян, If17]. Дано три змінні дійсного типу:  $a, b, c$ . Якщо їх значення упорядковані за зростання або за спаданням, подвоїти їх; в іншому випадку замінити значення кожної змінної на протилежне. Вивести нові значення змінних  $a, b, c$ .

4. [Абрамян, If9]. Дано дві змінні дійсного типу:  $a, b$  ( $a \neq b$ ). Перерозподілити значення даних змінних так, щоб в  $a$  виявилось менше із значень, а в  $b$  — більше. Вивести нові значення змінних  $a, b$ .

### 2.5.3. Умовний оператор з перевіркою багатьох умов

1. [Абрамян, If3]. Дано ціле число. Якщо воно додатна, додати до нього 1; якщо від'ємне відняти від нього 2; якщо нульове, замінити його на 10. Вивести отримане число

2. [Абрамян, If28]. Дано номер року (додатна ціле число). З'ясувати кількість днів у цьому році, враховуючи, що звичайний рік налічує 365 днів, а високосний — 366 днів. Високосним вважається рік, який ділиться на 4, за винятком тих років, які діляться на 100 та не діляться на 400 (наприклад, роки 300, 1300 та 1900 не є високосними, а 1200 та 2000 — високосні).

3. [Абрамян, If29]. Дано ціле число. Вивести його рядок-опис виду «від'ємне парне число», «нульове число», «додатна непарне число» і т. ін.

4. [Абрамян, If30]. Дано ціле число, яке лежить в діапазоні 1–999. Вивести його рядок-опис виду «парне двозначне число», «непарне тризначне число»

### 2.5.4. Розбір варіантів

1. [Абрамян, Case2]. Дано ціле число  $K$ . Вивести рядок-опис оцінки, що відповідає числу  $K$  (1 — «погано», 2 — «незадовільно», 3 — «задовільно», 4 —

«добре», 5 — «відмінно»). Якщо K не лежить в діапазоні 1–5, то вивести повідомлення «помилка».

**2.** [Абрамян, Case4].. Дано номер місяця — ціле число в діапазоні 1–12 (1 — січень, 2 — лютий і т.ін.). Визначити кількість днів в цьому місяці для невисокосного року.

**3.** [Абрамян, Case5].. Арифметичні операції над числами пронумеровані наступним: 1 — складання, 2 — віднімання, 3 — множення, 4 — цілочисельне ділення, 5 — залишок від цілочисельного ділення, 6 — ділення. Дано номер операції N (ціле число в діапазоні 1–6) та цілі числа A і B (B не дорівнює 0). Виконати над числами вказану операцію й вивести на екран результат

**4.** [Абрамян, Case8-9].. Дано два цілих числа: D (день) і M (місяць), які визначають правильну дату невисокосного року. Вивести значення D і M для дати, яка передеує, та наступної за вказаною.

**5.** Ввести з клавіатури номер дня тижня (ціле число). Вивести на екран назву дня. Якщо введене число не може бути номером дня, вивести на екран повідомлення «Неприпустиме значення змінної».

## 2.6. Задачі на читання програмного коду

<b>1</b>	<pre># 1. Вихідні дані a = 100 b = 1 print("a = {0}, b = {1}".format(a, b))  # 2. Обчислювальна частина if a &lt; b :     m1 = a     m2 = b else :     m1 = b     m2 = a  # 3. Вивід результатів на екран print("m2 = {0}, m1 = {1}".format(m2, m1))</pre>	<b>2</b>	<pre># 1. Вихідні дані a = 100 b = -10 c = 1  # 2. Обчислювальна частина m = a if b &lt; m :     m = b if c &lt; m :     m = c  # 3. Вивід результатів на екран print("m = {0}".format(m))</pre>
<b>3</b>	<pre># 1. Вихідні дані a = -100 b = 10 c = 1  # 2. Обчислювальна частина s = a + b + c p = a * b * c m = s if s &gt; p else p  # 3. Вивід результатів на екран print("m = ", m)</pre>	<b>4</b>	<pre># 1. Вихідні дані a = -100 b = 10 c = 1 print("{0} {1} {2}".format(a, b, c))  # 2. Обчислювальна частина if a &gt;= b and b &gt;= c :     a *= 2     b *= 2     c *= 2 else :</pre>

			<pre> a = abs(a) b = abs(b) c = abs(c)  # 3. Вивід результатів на екран print("{0} {1} {2}".format(a, b, c)) </pre>
<b>5</b>	<pre> # 1. Вихідні дані x = 20. y = -11.5 print("x = {0}, y = {1}".format(x, y))  # 2. Обчислювальна частина z = x-y if x &gt; y else y- x+1  # 3. Вивід результатів на екран print("z = {0}".format(z)) </pre>	<b>6</b>	<pre> # 1. Вихідні дані x = 20. y = -11.5  # 2. Вивід результатів на екран if x &gt; y :     print("x = {0}".format(x)) else :     print("x = {0}, y = {1}".format(x, y)) </pre>
<b>7</b>	<pre> # 1. Вихідні дані x = 20. y = 11. z = -12 print("x = {0}, y = {1}, z = {2}".format(x, y, z))  # 2. Обчислювальна частина if x+y+z &lt; 1 :     if x &lt; y and x &lt; z :         x = (y+z)/2     if y &lt; x and y &lt; z :         y = (x+z)/2     if z &lt; x and z &lt; y :         z = (x+y)/2 else :     if x &lt; y :         x = (y+z)/2     else :         y = (x+z)/2  # 3. Вивід результатів на екран print("x = {0}, y = {1}, z = {2}".format(x, y, z)) </pre>	<b>8</b>	<pre> # 1. Вихідні дані a = 12. b = 36. c = 4 d = 51 print("a = {0}, b = {1}, c = {2}, d = {3}".format(a, b, c, d))  # 2. Обчислювальна частина if a &lt;= b and b &lt;= c and c &lt;= d :     m = a     if m &lt; b :         m = b     if m &lt; c :         m = c     if m &lt; d :         m = d     a = b = c = d elif not(a &gt; b and b &gt; c and c &gt; d) :     a *= a     b *= b     c *= c     d *= d  # 3. Вивід результатів на екран print("a = {0}, b = {1}, c = {2}, d = {3}".format(a, b, c, d)) </pre>
<b>9</b>	<pre> # 1. Вихідні дані </pre>	<b>10</b>	<pre> # 1. Вихідні дані </pre>



	<pre>x = 0.1 y = 2. print("x = {0}, y = {1}".format(x, y))  # 2. Обчислювальна частина if x &lt; 0 and y &lt; 0 :     x = abs(x)     y = abs(y) elif (x &gt; 0 and y &lt; 0) or (x &lt; 0 and y &gt; 0) :     x += 0.5     y += 0.5 elif x &gt;= 0 and y &gt;= 0 and not(x &gt;= 0.5 and x &lt;= 2.) and not(y &gt;= 0.5 and y &lt;= 2.) :     x /= 10     y /= 10  # 3. Вивід результатів на екран print("x = {0}, y = {1}".format(x, y))</pre>		<pre>x = 2.1  # 2. Обчислювальна частина if x &lt;= 0 :     fx = -x elif x &lt;= 1 :     fx = x elif x &lt;= 2 :     fx = 1 else :     fx = 5 - 2*x  # 3. Вивід результатів на екран print("x = {0}, fx = {1}".format(x, fx))</pre>
11	<pre># 1. Вихідні дані k = int(input("Оцінка : "))  # 2. Вивід результатів на екран if k == 1 :     print("погано") elif k == 2 :     print("незадовільно") elif k == 3 :     print("задовільно") elif k == 4 :     print("добре") elif k == 5 :     print("відмінно") else :     print("Помилка")</pre>	12	<pre># 1. Вихідні дані n = 2 a = 0.6 b = 1.2  # 2. Обчислювальна частина # 3. Вивід результатів на екран if n == 1 :     s = a + b     print("{0} + {1} = {2}".format(a, b, s)) elif n == 2 :     r = a - b     print("{0} - {1} = {2}".format(a, b, r)) elif n == 3 :     p = a * b     print("{0} * {1} = {2}".format(a, b, p)) elif n == 4 :     d = a / b     print("{0} / {1} = {2}".format(a, b, d)) else :     print("Помилка")</pre>

## 2.7. Відповіді до задач на читання програмного коду

1. Дано 2 числа. Знайти їх максимум та мінімум.

2. Дано 3 числа. Знайти їх мінімум.
3. Дано 3 числа. Знайти максимум їх суми та добутку.
4. Дано дійсні числа  $a$ ,  $b$  і  $c$ . Подвоїти ці числа, якщо виконуються нерівності  $a \geq b \geq c$ ; замінити їх абсолютними значеннями, якщо це не так.
5. Дано дійсні числа  $x$ ,  $y$ . Обчислити  

$$z = \begin{cases} x - y, & \text{якщо } x > y; \\ y - x + 1, & \text{в іншому випадку.} \end{cases}$$
6. Дано 2 дійсних числа. Вивести на екран перше число, якщо воно більше другого, та обидва числа, якщо це не так.
7. Якщо сума 3 попарно різних дійсних чисел  $x$ ,  $y$ ,  $z$  менше 1, замінити найменше з них півсумою 2-х других, в іншому випадку замінити менше з  $x$  і  $y$  півсумою 2-х значень, які залишилися.
8. Дано дійсні числа  $a$ ,  $b$ ,  $c$ ,  $d$ . Якщо  $a \leq b \leq c \leq d$ , то кожне число замінити найбільшим з них; якщо  $a > b > c > d$ , то залишити незмінними; в іншому випадку – всі числа замінити їх квадратами.
9. Дано дійсні числа  $x$  і  $y$ . Якщо  $x$  і  $y$  від'ємні, то кожне з них замінити його модулем; якщо від'ємне тільки одно з них, то обидва значення збільшити на 0.5; якщо обидва значення невід'ємні та жодне з них не належить  $[0.5; 2.0]$ , обидва значення зменшити в 10 разів; в інших випадках – залишити значення незмінними.
10. Дано дійсне значення  $x$ . Обчислить  $f(x)$ , якщо  

для недодатніх	$f(x) = -x;$
$0 < x \leq 1,$	$f(x) = x;$
$1 < x \leq 2,$	$f(x) = 1;$
$2 < x,$	$f(x) = 5 - 2x$
11. Дано ціле число  $K$ . Вивести рядок-опис оцінки  $K$  (1 — «погано», 2 — «незадовільно», 3 — «задовільно», 4 — «добре», 5 — «відмінно»). Якщо  $K$  не належить діапазону 1–5, вивести повідомлення «помилка».
12. Арифметичні дії над числами пронумеровані наступним чином: 1 — складання, 2 — віднімання, 3 — множення, 4 — ділення. Дано номер дії  $N$  (ціле число в діапазоні 1–4) та дійсні числа  $A$  і  $B$  ( $B$  не дорівнює 0). Виконайте над числами вказану дію й виведіть результат.

## 2.8. Домашнє завдання

[ ЕПРЛ ], Тема 3 «Програми з розгалуженням»

п.3.6. Задачі з повним умовним оператором з складеною умовою

п.3.9. Задачі з неповним умовним оператором

п.3.11. Задачі з вкладеними умовними операторами

п.3.15. Задачі з оператором вибору `switch`

Варіанти згідно списку групи

### Тема 3. ЦИКЛІЧНИЙ АЛГОРИТМ

#### ЛАБОРАТОРНА РОБОТА №3

#### 3.1. Оператор циклу `for`. Оператори `continue`, `break`

3.1.1. Розв'язання задач

3.1.2. Задачі на написання програмного коду

3.1.3. Задачі на читання програмного коду

3.1.4. Відповіді до задач на читання програмного коду

3.1.5. Домашнє завдання

#### ЛАБОРАТОРНА РОБОТА №4

#### 3.2. Оператор циклу `while`

3.2.1. Розв'язання задач

3.2.2. Стандартний алгоритм Евкліда

3.2.3. Задачі на написання програмного коду

3.2.4. Задачі на читання програмного коду

3.2.5. Відповіді до задач на читання програмного коду

3.2.6. Домашнє завдання

#### ЛАБОРАТОРНА РОБОТА №5

#### 3.3. Вкладені цикли

3.3.1. Розв'язання задач

3.3.2. Задачі на написання програмного коду

3.3.3. Домашнє завдання

#### ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

<https://metanit.com/python/tutorial/2.7.php>

#### ЛАБОРАТОРНА РОБОТА №3

#### 3.1. Оператор циклу `for`. Оператори `continue`, `break`

3.1.1. Розв'язання задач

1. [Абрамов, №77 г)]. Дано число  $n$ . Обчислити

$$\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \dots + \sin n}$$

Результат – сума дробів.

Знаменник – сума синусів. На  $k$ -му кроці циклу до знаменника додається  $\sin k$ .

У знаменнику останнього дроби  $n$  доданків ( $\sin 1 + \dots + \sin n$ ), отже, цикл виконується  $n$  разів. Оскільки кількість кроків циклу відома заздалегідь, застосовується цикл `for`.

#### Алгоритм:

1) Підключення бібліотеки `math`

2) Введення з клавіатури значення змінної цілого типу  $n$

- 3) Змінним дійсного типу `summa_drobej` й `znamenatel` присвоюються нульові значення (0 не впливає на результат підсумовування)
- 4) Обчислення значення змінної `n_plus_1`
- 5) Цикл `for` з параметром  $k = \overline{1, n}$  з кроком зміни параметру, рівним 1. На кожній ітерації (витку цикла, кроці)) повторюються дії:
  - змінна `znamenatel` збільшується на  $\sin k$
  - до змінної `summa_drobej` додається  $1 / \text{znamenatel}$
- 6) Вивід на екран результату – значення змінної `summa_drobej`

Програмний код оператор циклу <code>for</code> з параметром	
1	<code>import math</code>
2	
3	<code># 1. Вихідні дані</code>
4	<code>n = int( input("n = ") )</code>
5	
6	<code># 2. Обчислювальна частина</code>
7	<code>summa_drobej = 0. # 0. - 0 дійсного типу</code>
8	<code>znamenatel = 0.</code>
9	<code>n_plus_1 = n + 1</code>
10	<code>for k in range(1, n_plus_1) :</code>
11	<code>    znamenatel += math.sin(k)</code>
12	<code>    summa_drobej += 1 / znamenatel</code>
13	
14	<code># 3. Вивід результатів на екран</code>
15	<code>print("{0}".format(summa_drobej))</code>

На екрані:

```
n = 10
summa_drobej = 2.8349232565889864
```

2. [Абрамов, №77 ж)] Дано число  $n$ . Обчислити

$$\sqrt{3 + \sqrt{6 + \dots + \sqrt{3(n-1) + \sqrt{3n}}}}$$

На  $k$  (кожній) ітерації підкореневий вираз збільшується на  $\sqrt{3 * k}$ ,  $k = \overline{1, n}$ .

Алгоритм:

- 1) Введення з клавіатури значення цілочисельної змінної  $n$ ;
- 2) Початкове значення дійсної змінної  $s$  дорівнює 0
- 3) Цикл `for` з параметром  $k = \overline{n, 1}$  з кроком зміни параметру, рівним  $-1$ .

На кожній ітерації циклу повторюються дії:

- до змінної  $s$  додається  $3 * k$
- обчислюється квадратний корінь змінної  $s$

## 4) Вивід на екран значення результуючої змінної s

Програмний код оператор циклу for (з параметром)	
1	import math
2	
3	# 1. Вихідні дані
4	n = int( input("n = ") )
5	
6	# 2. Обчислювальна частина
7	s = 0.
8	for k in range(n, 0, -1) :
9	s += 3*k
10	s = math.sqrt(s)
11	
12	# 3. Вивід результатів на екран
13	print("{0}".format(s))

На екрані:

```
n = 10
s = 2.46993
```

3. [Абрамов, №80] Дано дійсне число  $x$ . Обчисліть

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}$$

Алгоритм:

- 1) Значення цілої змінної n встановлюється рівним 13
- 2) Для дійсної змінної x вводиться значення з клавіатури
- 3) Початкове значення дійсної змінної summa дорівнює x
- 4) Ціла змінна знак дорівнює 1
- 5) Обчислення значення змінної n\_plus\_1
- 6) Цикл for з параметром  $k = \overline{3, n}$  з кроком зміни параметру, рівним 2.

На кожній ітерації повторюються дії:

– змінна знак змінює значення на протилежне (1 на -1, -1 на 1)

– змінна summa збільшується на  $\text{знак} * x^k / k!$

Факторіал обчислюється за допомогою функції factorial(k) з модулю math

- 7) Вивід на екран значення результуючої змінної summa

Програмний код оператор циклу for (з параметром)	
1	import math
2	
3	# 1. Вхідні дані
4	n = 13
5	x = float( input("x = ") )
6	
7	# 2. Обчислювальна частина

```

8   summa = x
9   znak = 1
10  n_plus_1 = n + 1
11  for k in range(3, n_plus_1, 2) :
12      znak = -znak
13      summa += znak * math.pow(x,k) / math.factorial(k)
14
15  # 3. Вивід результатів на екран
16  print("{0}".format(summa))

```

На екрані:

```

x = 2
summa = 0.9092974515196738

```

### 3.1.2. Задачі на написання програмного коду

1. [Абрамян, For10]. Дано ціле число  $N (> 0)$ . Знайти суму

$$1 + 1/2 + 1/3 + \dots + 1/N$$

2. [Абрамян, For13]. Дано ціле число  $N (> 0)$ . Знайти значення виразу  $1.1 - 1.2 + 1.3 \cdot \dots$  ( $N$  доданків). Умовний оператор не використовувати.

3. [Абрамян, For33]. Дано ціле число  $N (> 1)$ . Послідовність чисел Фібоначчі  $F_K$  (цілого типу) визначається наступним чином:

$$F_1 = 1, F_2 = 1, F_K = F_{K-2} + F_{K-1}, K = 3, 4, \dots$$

Вивести елементи  $F_1, F_2, \dots, F_N$ .

У наступних задачах перед розв'язанням задачі розібрати приклад, наведений на початку параграфа.

4. [ЕПРЛ, п.4.3, Вар.17, №1]. Надрукуйте всі двохзначні та трьохзначні числа, які складаються з різних цифр.

5. [ЕПРЛ, п.4.3, Вар.17, №2]. Знайти всі трьохзначні числа, що дорівнюють квадрату якогось двозначного числа.

6. [ЕПРЛ, п.4.4, Вар.17, №1]. Знайдіть кількість та середнє арифметичне натуральних 5-значних чисел, у яких перша цифра дорівнює останній, а друга – передостанній (наприклад, 12421, 47174).

7. [ЕПРЛ, п.4.4, Вар.17, №2]. Ввести з клавіатури  $n$  будь-яких чисел. Знайти середнє арифметичне всіх чисел, з парними порядковими номерами

8. [ЕПРЛ, п.4.5, Вар.17, №1]. Дано  $n$  цілих чисел. Чи всі вони належать інтервалу  $[x, y]$ ?

9. [ЕПРЛ, п.4.5, Вар.17, №2] Дано  $n$  цілих чисел. Чи є в цій послідовності тризначні додатні числа, які складаються з різних цифр?

10. [ЕПРЛ, п.4.5, Вар.17]. Дано ціле число  $t$ . Надрукувати з 3 знаками після крапки  $n$  чисел

$$a_i = \frac{\sqrt{5i + 8}}{i^2 + 1}, \quad i = \overline{1, n}$$

11. [ЕПРЛ, п.4.8, Вар.17]. Знайдіть значення функції на вказаному інтервалі з вказаним кроком:

$$f(x) = \begin{cases} \frac{\sqrt{x} + 1}{x^2 + 2}, & 0 \leq x \leq 10; \\ \frac{|2 + x| + 1}{3x}, & \text{у ост. вип.} \end{cases}$$

$x \in [-10, 10]$ ,  $h = 2,5$ .

**13.** [ЕПРЛ, п.4.9, Вар.17, №1]. Дано  $n$  цілих чисел. Знайти серед цих чисел перше двозначне число, яке складається з різних цифр та його порядковий номер. Якщо таких чисел немає, то вивести NO.

**14.** [ЕПРЛ, п.4.9, Вар.17, №2]. Дано  $n$  цілих чисел. Знайти серед цих чисел останнє число, яке належить інтервалу  $[x, y]$  та кратне  $b$  та його порядковий номер. Якщо таких чисел немає, то вивести NO.

**15.** [ЕПРЛ, п.4.10, Вар.17, №1]. Ввести з клавіатури  $n$  будь-яких чисел та число  $k$ . Знайдіть серед них число (його порядковий номер і значення) найближче до  $k$ . Якщо таких чисел декілька, то знайдіть перше.

**16.** [ЕПРЛ, п.4.10, Вар.17, №2]. Ввести з клавіатури  $n$  будь-яких чисел. Знайти серед них найменше тризначне число та найбільше двозначне число. Якщо є декілька таких чисел, то визначте порядкові номери останніх з них. Якщо немає таких чисел, то вивести повідомлення.

### 3.1.3. Задачі на читання програмного коду

<b>1</b>	<pre># 1. Вивід результатів на екран for i in range(10, 100) :     c1 = i // 10     c2 = i % 10     if c1 != c2 :         print(i, end = " ") print()</pre>	<b>2</b>	<pre># 1. Вивід результатів на екран for i in range(20, 90) :     c1 = i // 10     if c1 % 2 == 0 :         print(i, end = " ") print()</pre>
<b>3</b>	<pre># 1. Обчислювальна частина s = 0 n = 0 for k in range(1000,10000) :     c1 = k // 1000     c2 = (k - c1*1000) // 100     c3 = (k - c1*1000 - c2*100) // 10     c4 = k % 10     if c1==c4 and c2 == c3 :         s += k         n += 1 sa = s / n  # 2. Вивід результатів на екран print("{0}".format(sa))</pre>	<b>4</b>	<pre># 1. Вхідні дані n = 10  # 2. Обчислювальна частина p = 0 for k in range(n) :     a = int( input() )     if 100 &lt;= a and a &lt;= 999 :         p += 1  # 3. Вивід результатів на екран print("p = ", p)</pre>
<b>5</b>	<pre># 1. Вхідні дані n = 10</pre>	<b>6</b>	<pre>import math</pre>

	<pre># 2. Обчислювальна частина for k in range(n) :     a = int( input() )     ap = a &gt; 0     if ap == False :         break  # 3. Вивід результатів на екран if ap == True :     print("yes") else :     print("no")</pre>	<pre># 1. Вхідні дані n = 10  # 2. Обчислювальна частина # 3. Вивід результатів на екран n1 = n + 1 for k in range(1,n1) :     a = 3. / (k*k + 1) + math.pow(0.15*k, 1/k)     print("%5.3f" % a)</pre>
<b>7</b>	<pre>import math  # 1. Вхідні дані a = -11. b = 11. h = 2.75  # 2. Обчислювальна частина # 3. Вивід результатів на екран n = int( (b-a)/h + 1) for i in range(n) :     x = a + i*h     if -7. &lt;=x and x &lt;= 7. :         fx = 25./math.sqrt(x+9)     else :         fx = (4*x*x + 1)/ abs(x)     print("( %5.2f, %5.3f)" % (x, fx))</pre>	<b>8</b> <pre># 1. Вхідні дані n = 5 x = -11. y = 15.  # 2. Обчислювальна частина b = False for i in range(n) :     a = int( input() )     b = (x &lt;= a and a &lt;= y)     if b == True :         break  # 3. Вивід результатів на екран if b == True :     print("yes") else :     print("no")</pre>
<b>9</b>	<pre>import random  # 1. Вхідні дані n = 10 a = -100 b = 500  # 2. Обчислювальна частина bb = False for i in range(n) :     x = random.randint(a, b)     bb = bb or ( 100 &lt;= x and x &lt;= 999 and x % 11 == 0 )      if bb == True :         v = x         vn = i  # 3. Вивід результатів на екран if bb == True :     print("v = {0}, vn =</pre>	<b>10</b> <pre># 1. Вхідні дані n = 50 k = 2.8  # 2. Обчислювальна частина a0 = float( input() ) r0 = abs(k - a0) for i in range(1, n) :     a = float( input() )     r = abs(k - a)     if r &lt;= r0 :         r0 = r         a0 = a  # 3. Вивід результатів на екран print("a0 = {0}, r0 = {1}".format(a0, r0))</pre>



	<pre>{1}").format(v, vn)) else :     print("no")</pre>		
<b>11</b>	<pre>import random  # 1. Вхідні дані n = 5 a = -10. b = 20.  # 2. Обчислювальна частина k = 0 for i in range(n) :     a = random.uniform(a, b)     if a == int(a) :         k += 1  # 3. Вивід результатів на екран print("k = ", k)</pre>	<b>12</b>	<pre># 1. Вхідні дані n = 10  # 2. Обчислювальна частина b1 = float( input() ) b2 = float( input() ) q0 = b2 / b1 b1 = b2 for i in range(2,n) :     b2 = float( input() )     q = b2 / b1     t = (q == q0)     if t == False :         break     b1 = b2  # 3. Вивід результатів на екран if t == True :     print("yes") else :     print("no")</pre>

### 3.1.4. Відповіді до задач на читання програмного коду

- Надрукуйте всі 2-значні числа, які складаються з різних цифр
- Надрукуйте всі 2-значні непарні числа, перша цифра яких парна.
- Знайдіть середнє арифметичне 4-значних чисел, 1-а цифра яких дорівнює 4-й, 2-а – 3-й.
- З клавіатури введені  $n$  цілих чисел. Знайдіть кількість 3-значних чисел серед них.
- З клавіатури введені  $n$  цілих чисел. Чи є всі вони додатними?
- Дано ціле число  $n$ . Вивести на екран з 3-мя знаками після коми  $n$  членів послідовності, заданої формулою:

$$a_k = \frac{3}{k^2+1} + \sqrt[k]{0,15k}, \quad k = \overline{1, n}$$

- Протабулювати на відрізку  $[-11, 11]$  з кроком  $h = 2.75$  функцію

$$f(x) = \frac{25}{\sqrt{x+9}}, \quad -7 \leq x \leq 7;$$

$$\frac{4x^2 + 1}{|x|} \text{ . в ост. вип.}$$

- Дано  $n$  цілих чисел та дійсні числа  $x, y$ . Знайти серед них 1-е число, яке належить відрізку  $[x, y]$  та його порядковий номер. Якщо таких чисел немає, вивести на екран "no".

9. Дано цілі числа  $a$ ,  $b$  ( $b \geq 1000$ ) та  $n$  – кількість випадкових чисел з  $\{a, a+1, \dots, b\}$ . Знайти останнє 3-значне число, яке ділиться на 11, та його номер. Якщо таких чисел немає, вивести на екран “no”.

10. Дано ціле число  $n$ , дійсне число  $k$ . З клавіатури ввести  $n$  дійсних чисел, знайти серед них найближче до  $k$  число та його порядковий номер. Якщо таких чисел декілька, знайти останнє.

11. Дано  $n$  випадкових дійсних чисел з відрізка  $[-10, 20]$ . Знайти кількість цілих чисел серед них.

12. Ввести з клавіатури  $n$  дійсних чисел. Чи утворюють вони геометричну прогресію?

### 3.1.5. Домашнє завдання

[ЕПРЛ], Тема 4 «Цикли з параметром»

п.4.3. Задачі на пошук в інтервалі чисел, які задовольняють деякій умові

п.4.4. Задачі на знаходження кількості, суми, добутку, середнього арифметичного

п.4.5. Задачі з використання логічних змінних

п.4.7. Задачі на обчислення елементів послідовності за їх порядковим номером

п.4.8. Задачі на створення таблиць значень функцій

п.4.9. Задачі на пошук першого та останнього чисел з заданою властивістю

п.4.10. Задачі на пошук максимального та мінімального

п.4.12. Задачі з випадковими числами

п.4.13. Задачі з використанням 2-х змінних для вводу послідовності чисел

п.4.14. Задачі на обчислення елементів послідовності через попередні елементи

Варіанти згідно списку групи

## ЛАБОРАТОРНА РОБОТА №4

### 3.2. Оператор циклу `while`

#### 3.2.1. Розв'язання задач

1. [Абрамян, While3]. Дано цілі додатні числа  $N$  і  $K$ . Використовуючи тільки операції складання та віднімання, знайти частку від цілочисельного ділення  $N$  на  $K$ , а також залишок від цього ділення.

#### Алгоритм

1) Ініціалізувати цілі змінні  $n$  і  $k$

2) Цілу змінну  $q$  (частку) та  $r$  (залишок від ділення) покласти рівними  $0$  і  $n$ , відповідно.

3) Ідея рішення: оскільки  $n = k \cdot q + r$ , то  $r = n - k - \dots - k, k$

віднімається від  $n - q$  разів). Процес віднімання зупиняється, коли  $r$  стає меншим  $k$ . Кількість віднімань  $q$  невідомо заздалегідь, тому для розв'язання задачі використовується цикл з передумовою  $r \geq k$ .

Дії, які повторюються в циклі:

- а) зменшення залишку від ділення  $r$  на  $k$
- б) збільшення частки  $q$  на 1
- 4) Вивід на екран  $q$  і  $r$

Програмний код оператор циклу з передумовою while	
1	# 1. Вхідні дані
2	$n = 100$
3	$k = 12$
4	
5	# 2. Обчислювальна частина
6	$q = 0$
7	$r = n$
8	while $r \geq k$ :
9	$r -= k$
10	$q += 1$
11	
12	# 3. Вивід на екран результатів
13	<code>print("{0} = {1} * {2} + {3}".format(n, k, q, r))</code>

На екрані:

$$100 = 12 * 8 + 4$$

**2.** [Абрамов, №866)]. Дано натуральне число  $n$ . Знайти суму його цифр.

Алгоритм:

- 1) Задати значення цілій змінній  $n$
- 2) Цілі змінні  $nn$ ,  $sum\_digits$  покласти рівними  $n$  і  $0$ , відповідно
- 3) Кількість цифр в числі заздалегідь невідома, тому організувати цикл while з передумовою ( $nn \neq 0$ )

Дії, які повторюються в циклі:

- а) знайти останню цифру числа  $nn$ , дорівнює залишку від ділення на 10
- б) додати знайдену цифру до суми цифр
- в) розділити  $nn$  на 10
- 4) Вивід значення змінної  $sum\_digits$  на екран

Програмний код оператор циклу з передумовою while	
1	# 1. Вхідні дані
2	$n = 7651$
3	
4	# 2. Обчислювальна частина
5	$nn = n$
6	$sum\_digits = 0$
7	while $nn \neq 0$ :
8	$digit = nn \% 10$
9	$sum\_digits += digit$

10	nn // = 10
11	
12	# 3. Вивід на екран результатів
13	print("{0}, сума цифр = {1}".format(n, sum digits))

На екрані:

n = 7651 сума цифр = 19

**3. Знайти номер співмножника, на якому добуток**

$$1 \cdot 3 \cdot 5 \cdot 7 \cdot \dots$$

перевищить  $10^{11}$ .

$2 \cdot k - 1$ ,  $k = 1, 2, 3, \dots$  – непарні числа

Алгоритм:

- 1) Змінну MAX\_PRODUCT покласти рівною  $10^{11}$ .
- 2) Для змінної product початковим значенням є 1.
- 3) Цілочисельна змінна k – номер співмножника – дорівнює 0
- 4) здійснити перевірку умови продовження роботи циклу:  
product <= MAX\_PRODUCT
- 5) Дії, які повторюються в циклі:
  - збільшити на 1 значення змінної k
  - помножити product на k-е непарне число ( $2k-1$ )
- 6) вивід на екран значення змінної k

Програмний код оператор циклу while	
1	# 1. Вхідні дані
2	MAX_PRODUCT = 10**11
3	
4	# 2. Обчислювальна частина
5	k = 0
6	product = 1
7	while product <= MAX_PRODUCT:
8	k += 1
9	product *= (2*k - 1)
10	
11	# 3. Вивід на екран результатів
12	print("k = {0}, product = {1}".format(k, product))

На екрані:

k = 12, product = 316234143225

**3.2.2. Алгоритм Евкліда обчислення НСД(a, b)**

[https://uk.wikipedia.org/wiki/Алгоритм\\_Евкліда](https://uk.wikipedia.org/wiki/Алгоритм_Евкліда)

Програмний код оператор циклу з передумовою while	
1	# 1. Вхідні дані
2	a = 900
3	b = 280
4	

```

5 # 2. Обчислювальна частина - Алгоритм Евкліда
6 m = a
7 n = b
8 while ( m != n ) :
9     if (m > n) :
10         m -= n;
11     else :
12         n -= m
13
14 # 3. Вивід на екран результатів
15 print("НСД({0}, {1}) = {2}".format(a, b, m))

```

На екрані:

```
НСД(900,280) = 20
```

### 3.2.3. Задачі на написання програмного коду

1. [Абрамов, №88б)]. Дано натуральне число  $n$ . Знайти число, яке має зворотний порядок цифр.

2. Дано натуральне число  $n$ . Вивести на екран цифри числа  $n$  в прямому порядку.

3. [Абрамов, №88в)]. Дано натуральне число  $n$ . Переставити першу та останню цифри числа.

4. [Абрамов, №88г)]. Дано натуральне число  $n$ . Приписати 1 в початок та кінець числа.

5. [Абрамян, While24]. Дано натуральне число  $n$ . Перевірити, чи є  $n$  членом послідовності Фібоначчі.

6. [Абрамян, While15]. Початковий вклад у банку дорівнює 1000 грн. В кінці кожного місяця розмір вкладу збільшується на  $p$  процентів від наявної суми ( $p$  — дійсне число,  $0 < p < 25$ ). За даним  $p$  визначити, через скільки місяців розмір вкладу перевищить 1100 грн., вивести знайдену кількість місяців  $k$  (ціле число) та підсумковий розмір вкладу  $s$  (дійсне число).

7. [Абрамян, While28]. Дано дійсне число  $\varepsilon (> 0)$ . Послідовність дійсних чисел  $a_k$  визначається наступним чином:  $a_1 = 2$ ,  $a_k = 2 + 1/a_{k-1}$ ,  $k = 2, 3, \dots$ . Знайти перший з номерів  $k$ , для яких виконується умова  $|a_k - a_{k-1}| < \varepsilon$ , та вивести цей номер, а також числа  $a_{k-1}$  и  $a_k$ .

8. Знайти всі правильні нескоротні дроби, знаменник яких не перевищує 10.

9. Дано натуральні числа  $m$  і  $n$ . Знайти в множині  $\{1, \dots, n\}$  всі числа, взаємно прості з  $m$ .

10. [Абрамян, While28]. Дано ціле число  $n (> 1)$ . Вивести найменше з цілих чисел  $k$ , для яких сума  $1 + 1/2 + \dots + 1/k$  буде більше або рівна  $n$ , та саму цю суму.

### 3.2.4. Задачі на читання програмного коду

<b>1</b>	<pre># 1. Вхідні дані n = 1000  # 2. Обчислювальна частина k = 0 while 3**(k+1) &lt; n :     k += 1  # 3. Вивід на екран результатів print("k = ", k)</pre>	<b>2</b>	<pre># 1. Вхідні дані n = 729  # 2. Обчислювальна частина nn = n k = 0 while nn % 3 == 0 :     nn /= 3  # 3. Вивід на екран результатів if nn == 1 :     print("true") else :     print("false")</pre>
<b>3</b>	<pre># 1. Вхідні дані w = 1.8  # 2. Обчислювальна частина k = 1 s = 0 while s + k/(k+1) &lt;= w:     k += 1     s += (-1)**k * k/(k+1)  # 3. Вивід на екран результатів print("k = ", k)</pre>	<b>4</b>	<pre>k = int( input() ) while k % 2 == 0:     print("Помилковий ввід")     k = int( input() )  print("k = ", k)</pre>
<b>5</b>	<pre># 1. Вхідні дані n = 7625  # 2. Обчислювальна частина nn = n d = nn % 10 nn //= 10 m1 = d m2 = d while ( nn != 0 ) :     d = nn % 10     nn //= 10     if d &gt; m1 :         m1 = d     elif d &lt; m2 :         m2 = d  # 3. Вивід на екран результатів print("m1 = %1i, m2 = %1i" % (m1,m2))</pre>	<b>6</b>	<pre># 1. Вхідні дані n = 77777  # 2. Обчислювальна частина nn = n d = nn % 10 nn //= 10 dd = d while ( nn != 0 ) :     d = nn % 10     nn //= 10     e = (d == dd)     if e == False :         break  # 3. Вивід на екран результатів print(e)</pre>

### 3.2.5. Відповіді до задач на читання програмного коду

1. Дано натуральне число  $n$ . Знайти найбільше ціле число  $k$ , при якому

виконується нерівність  $3^k < n$ .

2. Дано натуральне число  $n$ . Якщо воно є ступенем числа 3, вивести `true`, якщо ні — вивести `false`.

3. Дано додатна дійсне число  $w$  ( $w \leq 1.8$ ). Знайти номер доданка, при якому сума знакозмінного ряду  $-1/2 + 2/3 - 3/4 + \dots$  перевищує  $w$ .

4. Користувач повинен ввести непарне число. У випадку вводу парного числа на екран повинно виводитися повідомлення про помилку, після чого дії повинні повторюватися до введення правильного значення.

5. Дано ціле число. Знайти його найбільшу й найменшу цифри.

6. Якщо задане число складається з однакових цифр, вивести на екран `true`, з різних – `false`.

### 3.2.6. Домашнє завдання

[ ЕПРЛ ], Тема 5 «Цикли з умовою»

п.5.2. Пошук декількох чисел, що задовольняють деякій умові

п.5.3. Задачі на перебір цифр натурального числа

п.5.4. Задачі на зміну натурального числа

п.5.5. Задачі на алгоритм Евкліда

п.5.6. Задачі на обчислення елементів послідовності за заданою точністю

Варіанти згідно списку групи

## ЛАБОРАТОРНА РОБОТА №5

### 3.3. Вкладені цикли

#### 3.3.1. Розв'язання задач

1. Намалювати символом «зірочка» (\*) прямокутник 3 x 5 (3 рядка по 5 зірочок у кожному)

Алгоритм:

1) Задати значення цілим змінним  $m = 3$  (кількість рядків) і  $n = 5$  (кількість зірочок в рядку)

2) Зовнішній цикл за номером рядка  $i$ , внутрішній – по номеру  $j$  зірочки в рядку. Після виводу на екран рядка зірочок, перевід рядку

Програмний код Вкладені цикли	
	<pre> m = 3 n = 5  for i in range(m) :     for j in range(n) :         print("*", end = "")     print()</pre>

На екрані:

```

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

## 2. Намалювати зірочками нижній правий трикутник

### Алгоритм

- 1) Задати значення цілої змінної  $n = 3$  (кількість рядків)
- 2) Зовнішній цикл – по номеру рядка
- 3) 1-й внутрішній цикл виводить на екран  $n-i-1$  пробілів, 2-й –  $i+1$  зірочку, після чого здійснюється перевід рядку.

Програмний код Вкладені цикли	
1	<code>n = 5</code>
2	
3	<code>for i in range(n) :</code>
4	<code>    for j in range(n-i-1) :</code>
5	<code>        print(" ", end = "")</code>
6	<code>    for j in range(i+1) :</code>
7	<code>        print("*", end = "")</code>
8	<code>    print()</code>

На екрані:

```

      *
     **
    ***
   ****
  *****

```

3. Дано натуральні числа  $a, b$ . Виведіть на екран числа з множини  $\{a, a+1, \dots, b\}$ , які мають 4 дільника.

### Алгоритм:

- 1) Задати значення цілих змінних  $a, b$
  - 2)  $d$  – кількість дільників числа,  $p$  – половина числа
- Зовнішній цикл `for` з параметром  $i$ , який змінюється від  $a$  до  $b$ .  
 $p$  дорівнює числу, цілочисельно поділеному на 2.

$d$  дорівнює 2, тому що 2 дільника (1 й само число), у заданого числа завжди є.

Оскільки всі дільники числа, окрім 1 та самого числа, належать множині  $\{2, \dots, p\}$ , у внутрішньому циклі, який обчислює кількість дільників числа  $i$ , параметр  $j = 2, \dots, p$ .

Перевірка: якщо кількість дільників числа дорівнює 4, то число  $i$  виводиться на екран.

Програмний код Вкладені цикли	
1	<code>a = 10</code>
2	<code>b = 25</code>
3	
4	<code>for i in range(a, b+1) :</code>
5	<code>    p = i // 2</code>
6	<code>    d = 2;</code>
7	<code>    for j in range(2, p+1) :</code>



8	if (i % j == 0) :
9	d += 1
10	if (d == 4) :
11	print(i, end = " ")

На екрані:

10 14 15 21 22

### 3.3.2. Задачі на написання програмного коду

#### 1. Намалювати зірочками фігури:

<pre> ***** ***** ***** </pre>	<pre> ***** ***** ***** </pre>	<pre> * ** *** **** ***** </pre>	<pre> ***** **** *** ** * </pre>
<pre> ***** **** *** ** * </pre>	<pre> * *** ***** *** * </pre>	<pre> ***** *** * * *** ***** </pre>	<pre> *      * *      * *      * *      * *      * *      * </pre>

2. Дано натуральні числа  $a$ ,  $b$ . Вивести на екран числа з множини  $\{a, a+1, \dots, b\}$ , які є квадратами простих чисел.

3. Дано натуральні числа  $a$ ,  $b$ . Знайти число з найбільшою кількістю дільників. Якщо таких чисел декілька, вивести на екран найменше з них

4. Дано число  $n$ . З  $n$  членів послідовності Фібоначчі вивести на екран прості числа.

#### 3.3.3. Домашнє завдання

[ ЕПРЛ ], Тема 4 «Цикли з параметром»

п.4.15. Задачі на вивід чисел у вигляді таблиці

п.4.16. Задачі на перебір варіантів

п.4.17. Задачі на знаходження дільників натуральних чисел

[ ЕПРЛ ], Тема 5 «Цикли з умовою»

п.5.7. Задачі з вкладеними циклами

Варіанти згідно списку групи