

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 1
з навчальної дисципліни
“Скриптові мови програмування (Python)”
АРИФМЕТИЧНІ ВИРАЗИ, УПРАВЛЯЮЧІ КОНСТРУКЦІЇ
ТА МАСИВИ У PYTHON

ВИКОНАВ
студент академічної групи КН-24
Мироненко Я.М

ПЕРЕВІРИВ
асистент кафедри кібербезпеки та
програмного забезпечення
Ткаченко О.С

Тема: Арифметичні вирази, управляючі конструкції та масиви у Python

Мета: навчитися створювати найпростіші програми на Python, використовуючи оператори вибору і циклів, арифметичні вирази та масиви.

Варіант 21

$$1) z = \begin{cases} -\sqrt{x}, & x > 45 \\ \sin 2x, & x \leq 45 \end{cases}$$

Число x вводиться користувачем у консолі Python.

2) Знайти перше число Фібоначчі, що буде більше заданого числа p (у послідовності Фібоначчі кожне наступне число дорівнює сумі двох попередніх: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377. $F_0=1, F_1=1, F_2=1, F_n=F_{n-1}+F_{n-2}, n \geq 2$).

3) Дано одномірний масив, що складається з N дійсних елементів.

- Знайти максимальний елемент.
- Обчислити середнє арифметичне від'ємних елементів масиву.

Вивести масив на екран у зворотному порядку.

Завдання 1

$$z = \begin{cases} -\sqrt{x}, & x > 45 \\ \sin 2x, & x \leq 45 \end{cases}$$

Лістинг програми

```
import math

x = int(input("Введіть ціле число: \n"))
if x > 45:
    z = -math.sqrt(x)
elif x <= 45:
    z = math.sin(2*x)
print(z)
print("Натисніть Enter для виходу...")
input()
print("Программа завершена.")
```

Принцип роботи програми

Користувач вводить число.

Якщо $x > 45 \rightarrow$ обчислюється від'ємний квадратний корінь.

Якщо $x \leq 45 \rightarrow$ обчислюється синус подвоєного значення.

Виводиться результат.

Очікує натискання Enter перед виходом.

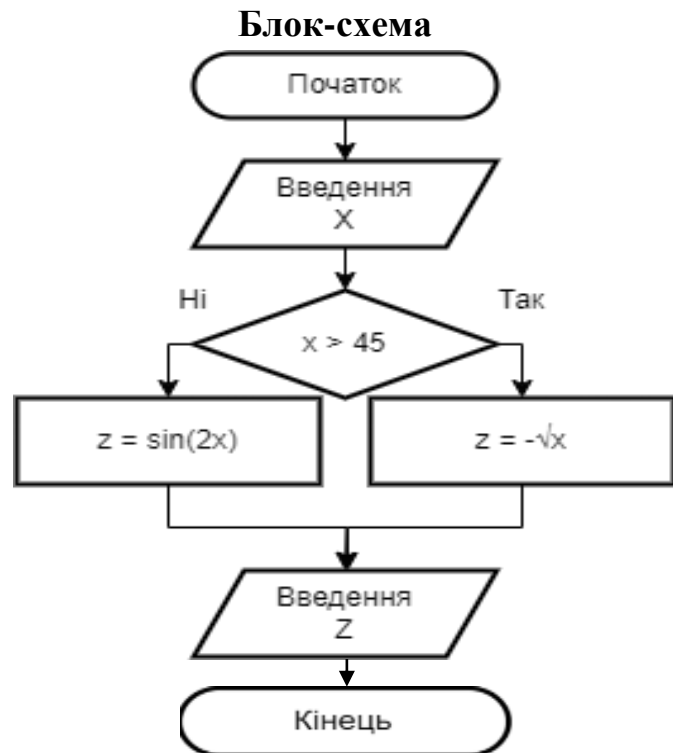
Тестові запуски застосунку

```
Введіть ціле число:  
7  
0.9906073556948704  
Натисніть Enter для виходу...  
Програма завершена.
```

Рисунок 1

```
Введіть ціле число:  
81  
-9.0  
Натисніть Enter для виходу...  
Програма завершена.
```

Рисунок 2



Завдання 2

Знайти перше число Фібоначчі, що буде більше заданого числа p (у послідовності Фібоначчі кожне наступне число дорівнює сумі двох попередніх: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377. $F_0=1, F_1=1, F_2=1, F_n=F_{n-1}+F_{n-2}, n \geq 2$).

Лістинг програми

```
while True:
    p = int(input("\nВведіть число p: "))
    F0, F1 = 1, 1

    while F1 <= p:
        F0, F1 = F1, F0 + F1
    print(f"Перше число Фібоначчі, більше за p({p}): {F1}")

    choice = input("Хочете продовжити? (Введіть так): ")
    if choice not in ("так", "yes", "y", "т"):
        print("Програма завершена.")
        break
```

Принцип роботи програми

Користувач вводить число p . Програма генерує числа Фібоначчі до тих пір, поки число не перевищить p . Виводиться перше число Фібоначчі, яке більше за p . Користувач запитує, чи хоче він продовжити.

Якщо відповідь позитивна (наприклад, "так"), цикл повторюється.

Якщо відповідь негативна, програма завершується.

Тестові запуски застосунку

```
Введіть число p: 7
Перше число Фібоначі, більше за p(7): 8
Хочете продовжити? (Введіть так):
```

Рисунок 3

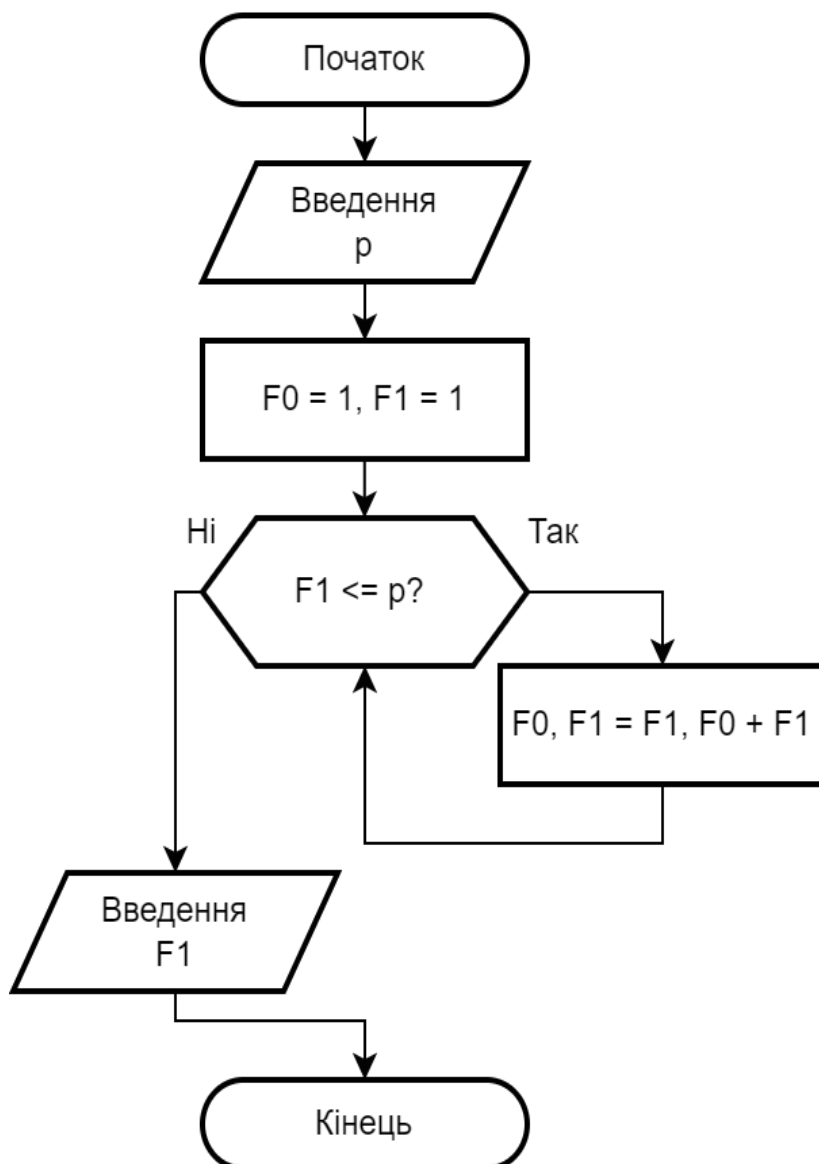
```
Введіть число p: 123
Перше число Фібоначі, більше за p(123): 144
Хочете продовжити? (Введіть так):
```

Рисунок 4

```
Введіть число p: -34
Перше число Фібоначі, більше за p(-34): 1
Хочете продовжити? (Введіть так):
```

Рисунок 5

Блок-Схема



Завдання 3

Дано одномірний масив, що складається з N дійсних елементів.

- Знайти максимальний елемент.
- Обчислити середнє арифметичне від'ємних елементів масиву.

Вивести масив на екран у зворотному порядку.

Лістинг

```
import random
while True:
    n = int(input("Введіть кількість елементів масиву: "))

    array = [round(random.uniform(-10, 10), 2) for _ in range(n)]
    negative_num = [num for num in array if num < 0]

    print(f"Початковий масив\n{array}")
    print(f"\nМаксимальний елемент: {max(array)}\n")
    print("Середнє арифметичне від'ємних елементів масиву:", sum(negative_num) /
len(negative_num) if negative_num else "0\n")
    print(f"\nМасив у зворотньому порядку: \n{array[::-1]}\n")

    choice = input("Хочете продовжити? (Введіть так): ")
    if choice not in ("так", "yes", "y", "t"):
        print("Програма завершена.")
        break
```

Принцип роботи програми

Введення кількості елементів: Користувач вводить кількість елементів масиву (n).

Генерація масиву: Програма створює масив з n випадкових дійсних чисел у діапазоні від -10 до 10, округлених до двох знаків після коми.

Пошук від'ємних чисел: Програма знаходить усі від'ємні числа у масиві.

Виведення результатів:

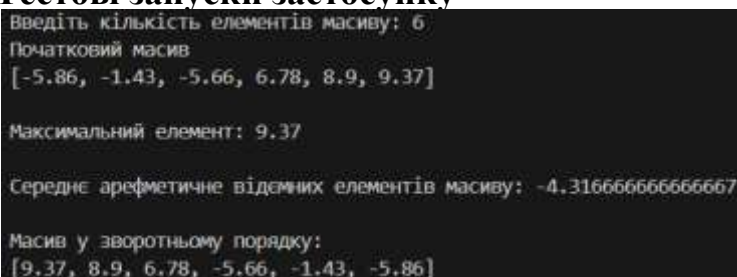
Виводить початковий масив.

Знаходить і виводить максимальний елемент масиву.

Обчислює і виводить середнє арифметичне від'ємних чисел (якщо вони є, інакше виводить "0").

Виводить масив у зворотньому порядку.

Тестові запуски застосунку



```
Введіть кількість елементів масиву: 6
Початковий масив
[-5.86, -1.43, -5.66, 6.78, 8.9, 9.37]

Максимальний елемент: 9.37

Середнє арифметичне від'ємних елементів масиву: -4.316666666666667

Масив у зворотньому порядку:
[9.37, 8.9, 6.78, -5.66, -1.43, -5.86]
```

Рисунок 6

```
Введіть кількість елементів масиву: 1
Початковий масив
[6.31]

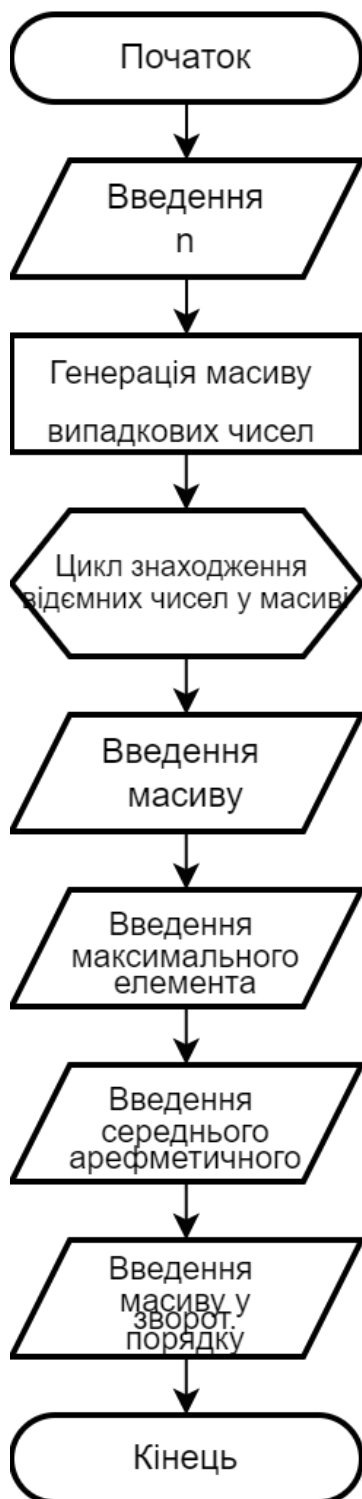
Максимальний елемент: 6.31

Середнє арифметичне від'ємних елементів масиву: 0

Масив у зворотньому порядку:
[6.31]
```

Рисунок 7

Блок-схема



Контрольні питання:

1. Які особливості та переваги мови Python Ви знаєте?

Простота та читабельність: Синтаксис близький до природної мови, що робить код зрозумілим.

Динамічна типізація: Змінні не потребують явного оголошення типу.

Крос-платформеність: Працює на Windows, Linux, macOS тощо.

Велика стандартна бібліотека: Модулі для роботи з мережею, файлами, математикою, тощо.

Підтримка парадигм програмування: ООП, функціональне, процедурне програмування.

Спільнота та ресурси: Безкоштовні курси, фреймворки (Django, Flask), бібліотеки (NumPy, Pandas).

Універсальність: Використовується у веб-розробці, Data Science, AI, автоматизації тощо.

2. Назвіть основні принципи синтаксису мови Python.

Відступи (indentation): Блоки коду виділяються пробілами/табуляцією (не фігурними дужками).

Кінець інструкції: Крапка з комою не обов'язкова; нова строка = нова інструкція.

Регістрочутливість: Змінні name та Name — різні.

Двокрапка після умов/циклів

if x > 0:

 print("Так")

Коментарі: Починаються з символу #.

3. Як здійснюється введення/виведення даних у мові Python?

name = input("Введіть ім'я: ") # Повертає рядок

number = int(input("Введіть число: ")) # Конвертація в ціле число

4. Який синтаксис циклу for у мові Python?

Ітерація через послідовності (списки, рядки, словники тощо):

for item in [1, 2, 3]:

 print(item)

Функція range():

for i in range(5): # 0,1,2,3,4

Паралельна ітерація:

```
print(i)
```

for key, value in dictionary.items():

```
print(key, value)
```

5. Назвіть принципи роботи зі списками у мові Python напиши відповіді розумні

Створення списку:

```
my_list = [1, "два", 3.0] # Гетерогенні елементи
```

Індексація та зрізи (slicing):

```
print(my_list[0]) # Перший елемент
```

```
print(my_list[-1]) # Останній елемент
```

```
print(my_list[1:3]) # Елементи з індексами 1 та 2
```

Методи:

append() — додати елемент,

remove() — видалити елемент,

sort() — сортувати список.

Генератори списків:

```
squares = [x**2 for x in range(10)]
```