

Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



Звіт

З лабораторної роботи №7

Варіант – 3

З дисципліни: «Кросплатформні засоби програмування»

На тему: «Дослідження базових конструкцій мови Python»

Виконав: ст. гр. КІ-306

Братівник Д. А.

Прийняв:

доцент кафедри ЕОМ

Іванов Ю. С.

Львів 2023

Мета роботи: ознайомитися з базовими конструкціями мови Python

ЗАВДАННЯ

1. Написати та налагодити програму на мові Python згідно варіанту. Програма має задовольняти наступним вимогам:

- програма має розміщуватися в окремому модулі;
- програма має генерувати зубчатий список, який міститиме лише заштриховані

області квадратної матриці згідно варіанту;

- розмір квадратної матриці і символ-заповнювач масиву вводяться з клавіатури;
- при не введенні або введенні кількох символів-заповнювачів відбувається коректне

переривання роботи програми;

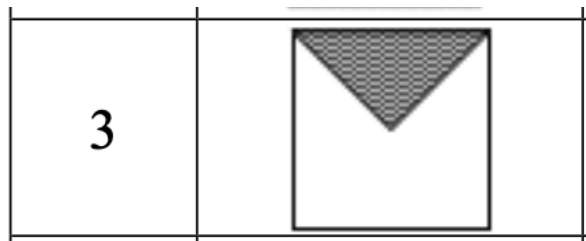
- сформований масив вивести на екран;
- програма має містити коментарі.

2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її

виконання та фрагменту згенерованої документації та завантажити його у ВНС.

4. Дати відповідь на контрольні запитання.

Варіант завдання:



Код програми:

```
import sys
```

```
# Зчитуємо кількість рядків у квадратній матриці від користувача
```

```
rows_num = int(input("Введіть розмір квадратної матриці: "))
```

```
# Зчитуємо символ-заповнювач від користувача
filler = input("Введіть символ-заповнювач: ")

# Перевірка чи введений символ-заповнювач не порожній рядок
if len(filler) == 0:
    print("Не введено символ-заповнювач")
    sys.exit(1)

# Обчислюємо кількість підмасивів у зубчастому масиві
n_subarrays = (rows_num + 1) // 2
newline_count = (rows_num // 2) - 1 # Кількість переходів на нову стрічку
leading_spaces = 0 # Змінна для визначення кількості пробілів перед
кожним рядком

# Ітеруємося через кожен підмасив та його елементи
for i in range(n_subarrays):
    # Додаємо пробіли перед кожним підмасивом
    for _ in range(leading_spaces):
        print(" ", end=" ")

    num_elements = rows_num - 2 * i # Обчислюємо кількість елементів у
поточному підмасиві
    arr = []
    # Заповнюємо підмасив символами-заповнювачами
    for _ in range(num_elements):
        if len(filler) == 1:
            arr.append(ord(filler)) # Конвертуємо символ у його ASCII код
            print(chr(arr[-1]), end=" ") # Виводимо символ за його ASCII кодом
        else:
            print("Забагато символів-заповнювачів")
            sys.exit(1)
```

`print()` # Переводимо курсор на новий рядок для виводу наступного підмасиву

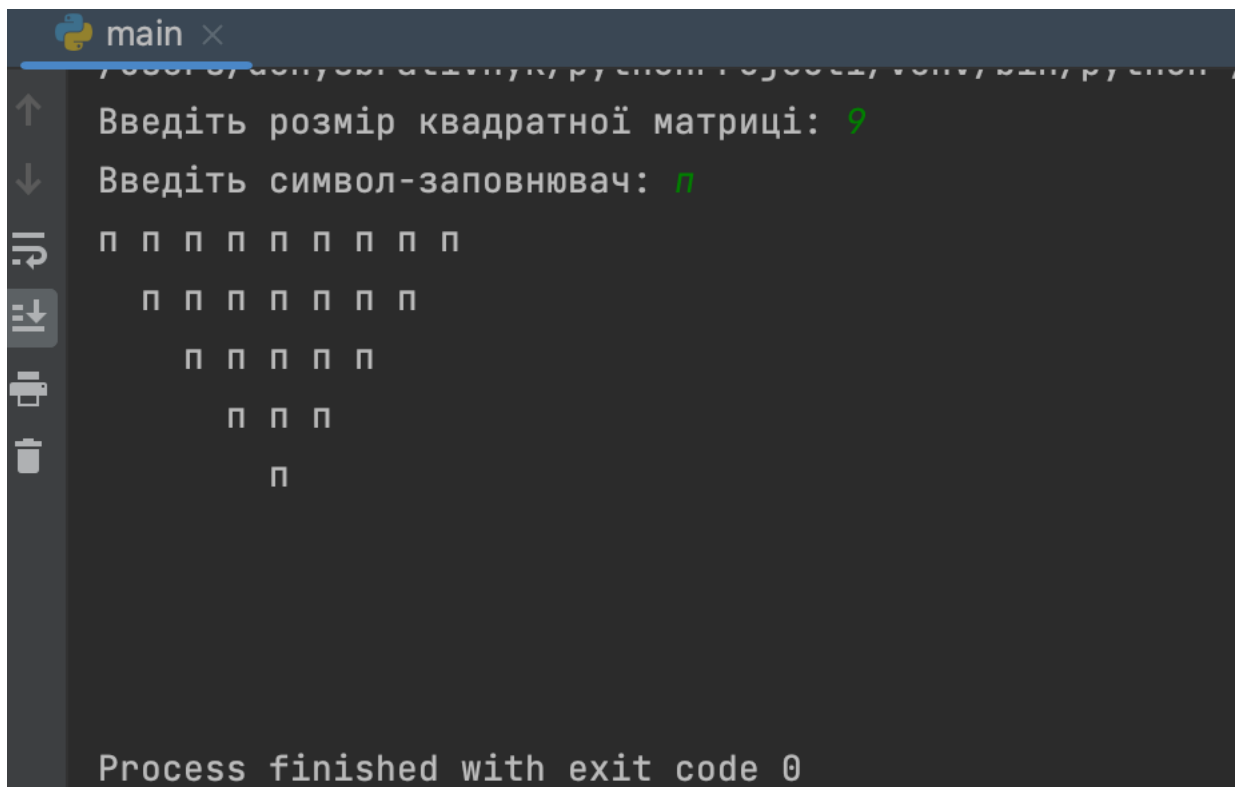
`leading_spaces += 1` # Збільшуємо кількість пробілів перед наступним підмасивом

Додаємо переходи на нову стрічку в кінці виводу

`for _ in range(newline_count):`

`print()`

Результати роботи програми:



```
main x
Введіть розмір квадратної матриці: 9
Введіть символ-заповнювач: p
п п п п п п п п п
  п п п п п п п
    п п п п п
      п п п
        п п
          п

Process finished with exit code 0
```

Відповіді на контрольні запитання

1. Який вигляд має програма мовою Python?
- має вигляд послідовності інструкцій у текстовому файлі з розширенням ".py".
2. Як запустити на виконання програму мовою Python? - "python ім'я_файлу.py" у командному рядку.
3. Які коментарі підтримує Python?
- однорядкові коментарі з символом "#" та багаторядкові коментарі, обмежені потрійними лапками (" або """).

4. Які типи даних підтримує Python?
 - числа (цілі, дійсні, комплексні), рядки, списки, кортежі, множини, словники та інші.
5. Як оголосити змінну?
 - присвоївши їй значення, наприклад, "змінна = значення".
6. Які керуючі конструкції підтримує Python?
 - умовні конструкції (if-elif-else), цикли (for, while), та інструкції для контролю потоку виконання програми.
7. Які операції підтримує Python?
 - додавання, віднімання, множення, ділення, цілочисельного ділення, залишок від ділення, порівняння та інші.
8. Як здійснити ввід з консолі? - "input()".
9. Як здійснити вивід у консоль? - "print()".
10. Як здійснити приведення типів?
 - Приведення типів можна виконати за допомогою функцій, наприклад, "int()", "float()", "str()".

Висновок

Під час вивчення базових конструкцій мови Python, було освоєно ключові аспекти програмування. Це включає в себе вигляд програми, запуск її виконання, роботу з коментарями та типами даних, оголошення змінних, керуючі конструкції, операції, а також навички вводу та виводу даних та приведення типів.