Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження рекурсивних алгоритмів»

Варіант 21

Виконав студент ІП-11, Ляля Іван Олександрович

Перевірила Мартинова Оксана Петрівна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2021

**Лабораторна робота 6**

**Дослідження рекурсивних алгоритмів**

Мета – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Задача: Написати рекурсивну функцію для розрахунку ступеня n дійсного числа а (n-натуральне число).

Постановка задачі:

В даній задачі ми вводимо число і натуральний степінь, до якого його треба піднести. Отже, головна функція складатиметься з трьох кроків – ввід цих даних, підпрограма обчислення степеня n числа a, його вивід. Деталізуємо підпрограму обчислення степеня. Це буде функція pow з двома аргументами (формальні параметри x тa y), де фактичними параметрами будуть a i n відповідно. Якщо степінь, до якого треба піднести число дорівнює 1, то x^1=x, а отже pow(x,y) повинна повернути просто z=x. Однак, якщо степінь числа не дорівнює 1, а більше, то z(змінна, що буде в нас повертати значення функції pow) дорівнює x\*pow(x,y-1) за властивістю степенів.

Математична модель:

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Значення** | **Тип** |
| **a** | **Основа степеня, вхідні дані** | **Дійсне число** |
| **n** | **Показник степеня, вхідні дані** | **Натуральне число** |
| **pow(x,y)** | **Функція піднесення числа x до степеня y** | **Функція двох параметрів** |
| **x** | **Основа степеня, формальний аргумент** | **Дійсне число** |
| **y** | **Показник степеня, формальний аргумент** | **Натуральне число** |
| **z** | **Змінна, в якій повертається значення функції, проміжні дані** | **Дійсне число** |
| **step** | **Результат розрахунку a^n, вихідні дані** | **Дійсне число** |

**Крок 1**. Визначимо основні дії

**Крок 2**. Описуємо ввід

**Крок 3**. Деталізуємо роботу підпрограми

**Крок 4**. Описуємо вивід

***Псевдокоди з блок-схемами:***

Крок 1:

**Початок**

Ввести a,n

Обчислити step=a^n рекурсивно

Вивести результат

**Кінець**

Крок 2:

**Початок**

**Input(a,n)**

Обчислити step=a^n рекурсивно

Вивести результат

**Кінець**

Крок 3:

Pow(x,y) //функція

{

**If** (y==1) z=x

**else** z=x\*pow(x, y-1)

**return** z;

}

**Початок**

**Input(a,n)**

step=pow(a,n) *//функція*

Вивести результат

**Кінець**

Крок 4:

Pow(x,y) //*функція*

{

**If** (y==1) z=x

**else** z=x\*pow(x, y-1)

**return** z;

}

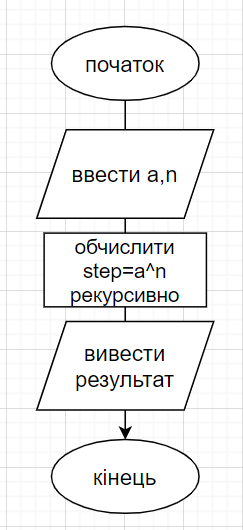
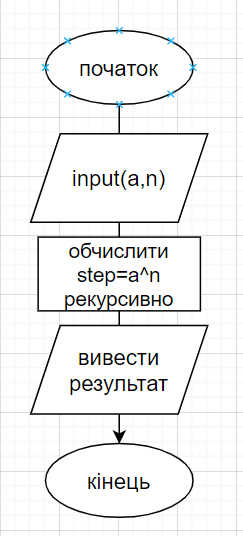
**Початок**

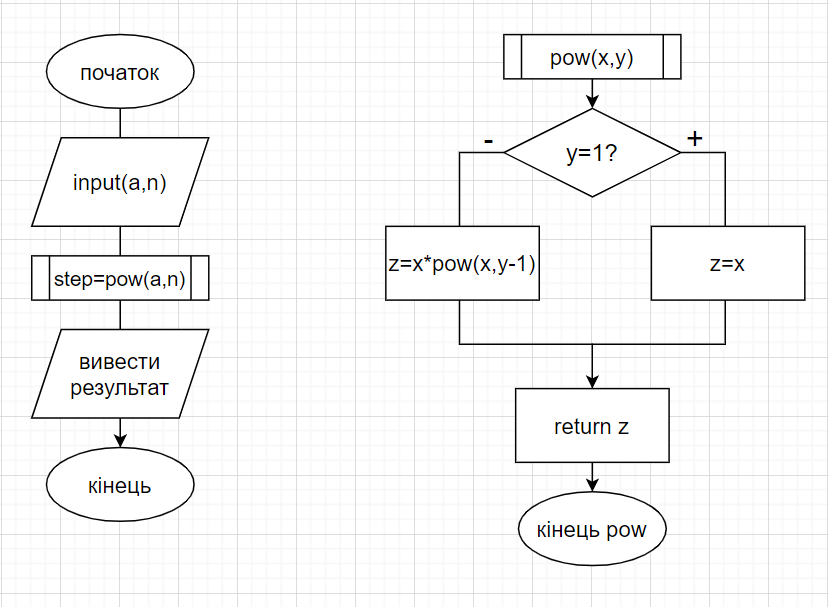
**Input(a,n)**

step=pow(a,n) *//функція*

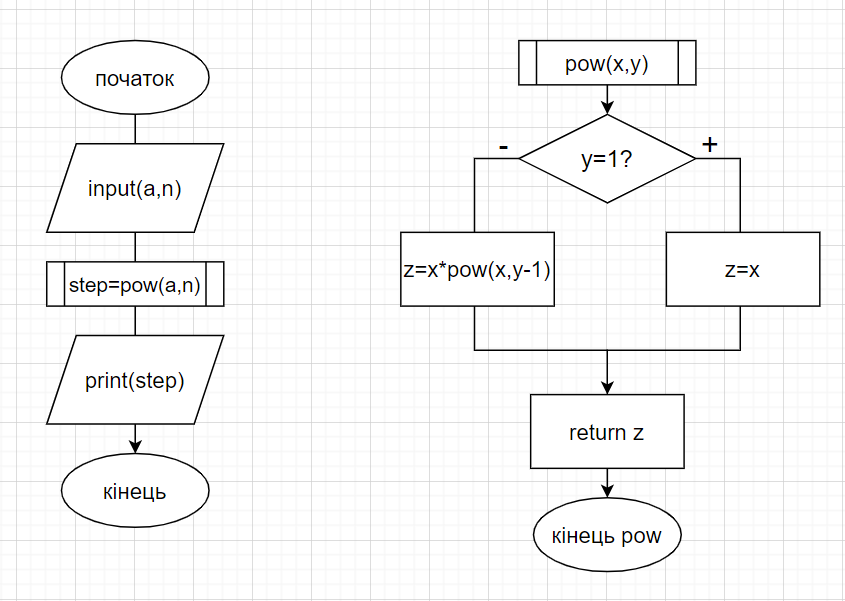
**print**(step)

**Кінець**

**1) 2)**

**3)**

**4)**



**Код і результати роботи програми на Python:**

def poww(x,y):

if y==1:

z=x

else:

z=x\*poww(x,y-1)

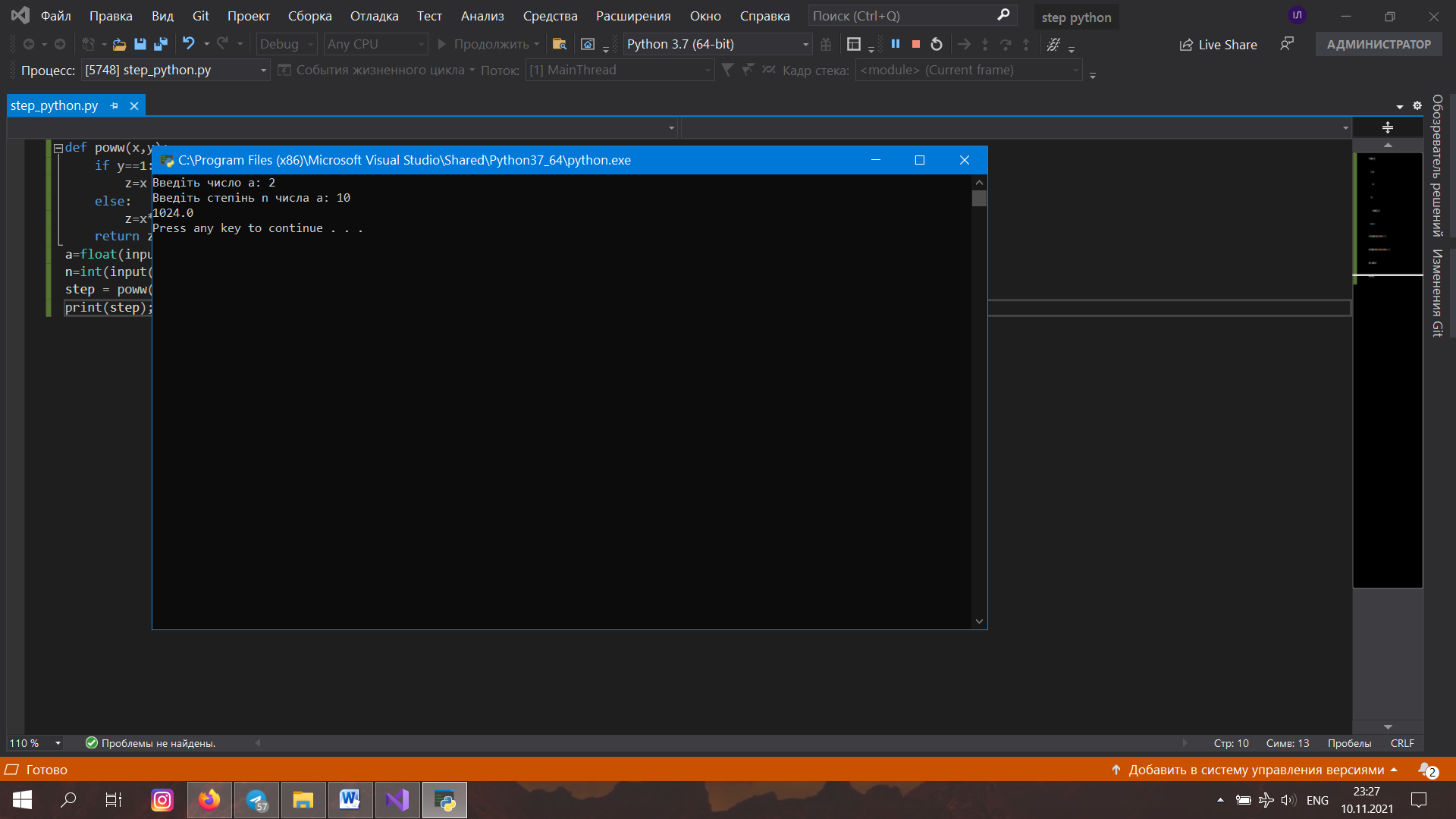
return z

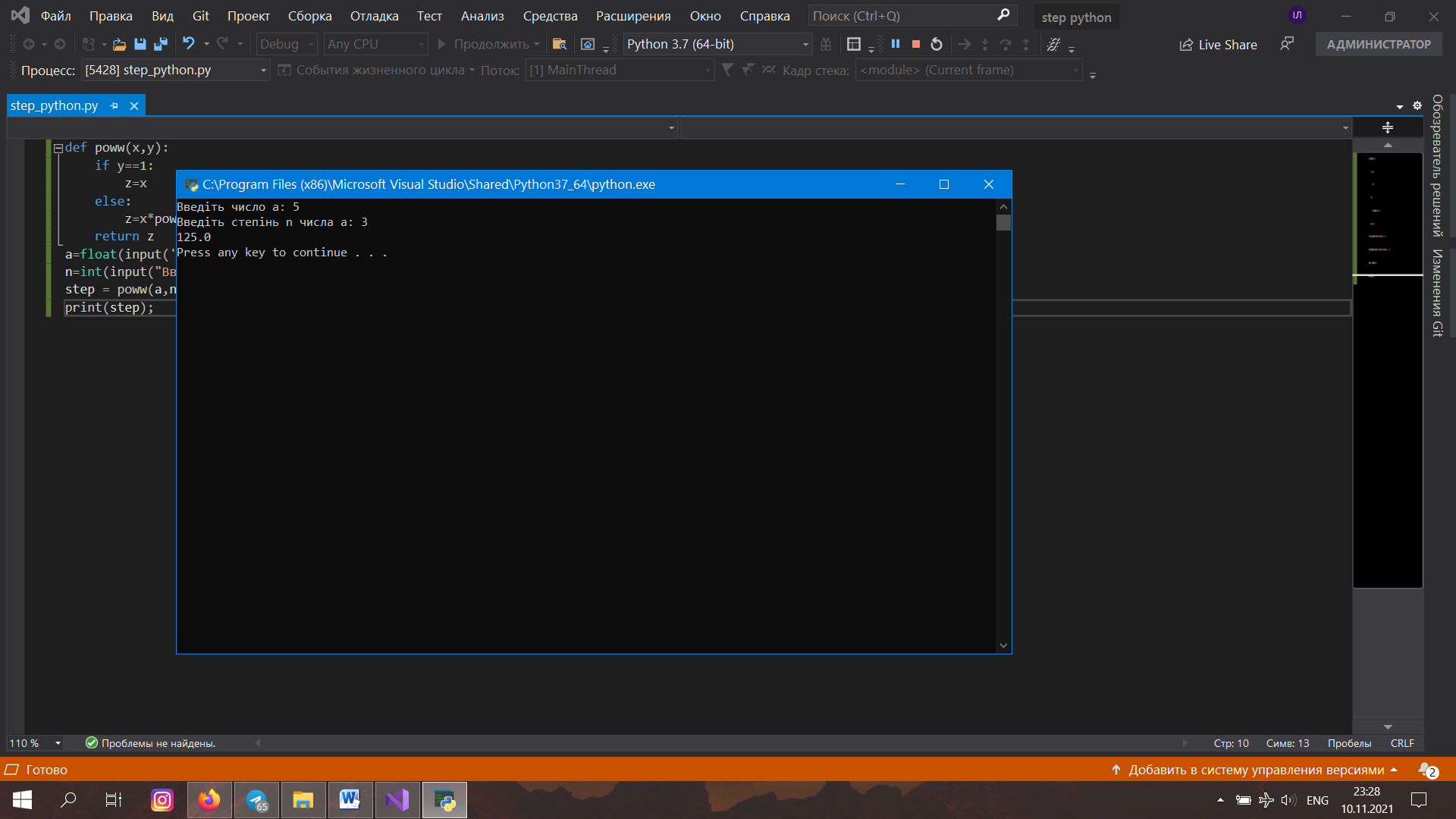
a=float(input('Введіть число а: '))

n=int(input("Введіть степінь n числа а: "))

step = poww(a,n)

print(step);





**Висновок:**

На лабораторній роботі номер 6 я дослідив роботу рекурсії на конкретній задачі, а саме обчислення степеня числа рекурсивно. Весь алгоритм складається з трьох частин – ввід, одна підпрограма і вивід. Підпрограма в свою чергу – один альтернативний умовний блок і операція повернення return. Програма, написана на Python, працює правильно, оскільки 2^10 справді дорівнює 1024, а 5^3 – 125. Глибина рекурсії залежить від введених даних і дорівнює n-1.