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\_**20**\_\_

Виконав студент: **ІП-11 Лошак Віктор Іванович**

(шифр, прізвище, ім'я, по батькові)

Перевірив: **Мартинова О.П.**

Київ 2021

**Лабораторна робота**

**Дослідження рекурсивних алгоритмів**

**Мета:**

дослідити особливості роботи рекурсивних алгоритмів та набути практичних

навичок їх використання під час складання програмних специфікацій підпрограм

**Варіант №20**

**Задача:**

Обчислити значення квадрата цілого додатного числа n рекурсивно, якщо відома залежність

**Розв’язок:**

1)Постановка задачі

Результатом розв’язку є число значення якого отримується

із заданого користувачем числа використовуючи процес знаходження рекурсивного розкладу числа. Для обчислення виразу при заданому n підпрограму. Значенням n є число що вводиться користувачем за запитом на початку програми. Для правильного виконання програми робимо перевірку числа на знак, і у випадку якщо число від`ємне під час виводу результату повідомляємо користувача про помилку. Для реалізації рекурсивного алгоритму використаємо логічні оператори.

2)Побудова математичної моделі

Складемо таблицю імен змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Користувацький ввід | цілий | n | Початкове дане/ результат обчислень |

3)Програмні специфікації напишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1:* Визначимо основні дії.

*Крок 2:* Вивід результату та деталізуємо рекурсивний алгоритм знаходження квадарату числа всередині підпрограми

5)Псевдокод

*Крок 1*

**Початок**

-**ввід** n

- вивід результату, реалізація рекурсивного алгоритму

**Кінець**

*Крок 2*

**Початок**

-**ввід** n

- **якщо** n>0 **то:**

**вивід pow2(n)**

**інакше:**

**вивід** помилка

**все якщо**

**Кінець**

*Підпрограма*

-**pow2( n)**

- **якщо** n==1

**то:**

**return** 1

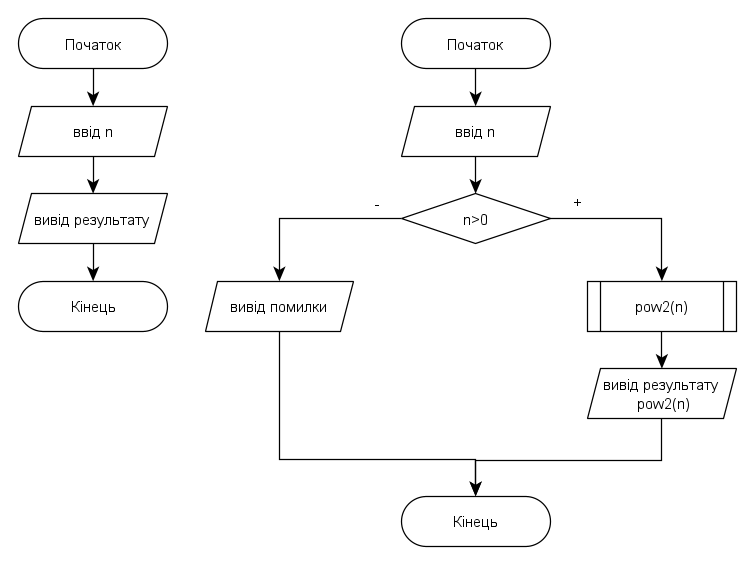
**все якщо**

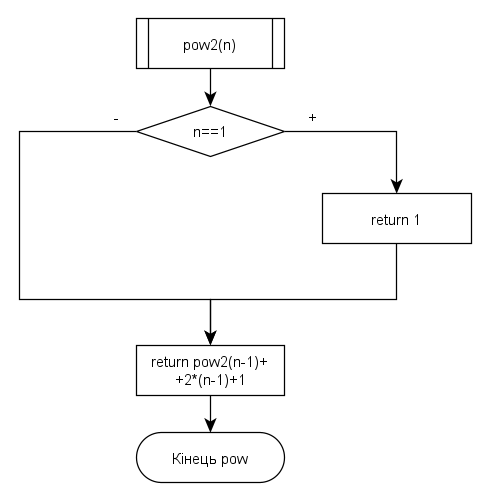
**return pow2(n-1) +** 2(n-1) + 1

- **кінець pow**

**Блок-Схема**

**Крок 1 Крок 2**

** Підпрограма:**

****

**Код програми Python:**

def pow2(a: int):

    if a==1:

        return 1

    return pow2(a-1) +2\*(a-1) +1

n = int(input("Enter the value:"))

print(pow2(n) if n > 0 else "Enter other value")

Перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних:

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Введення n=5; |
| 2 | Виклик pow2(5) => n!=1 => pow2(4) + 2\*4 +1 |
| 3 | Виклик pow2(4) => n!=1 => pow2(3) + 2\*3 + 1+ 2\*4 +1 |
| 4 | Виклик pow2(3) => n!=1 => pow2(2)+2\*2+1 + 2\*3 + 1+ 2\*4 +1 |
| 5 | Виклик pow2(2) => n!=1 => pow2(1)+2\*1+1+2\*2+1 + 2\*3 + 1+ 2\*4 +1 |
| 6 | Виклик pow2(1) => n=1 => 1+2\*1+1+2\*2+1 + 2\*3 + 1+ 2\*4 +1=25 |
| 7 | Виведення 25 |
|  | Кінець |

**Висновок:**

В ході виконання даної лабораторної роботи я навчився працювати з складними циклами, досліджував нелінійні програмні специфікації для визначення і обчислення виразів умова яких залежить від певних параметрів вводу, набув практичних навичок даних використання рекурсивних алгоритмів з глибиною рекурсії 1 під час складання нелінійних програмних специфікацій. Навчився створювати блок-схеми алгоритмів що містять рекурсію та описувати програму за допомогою псевдокоду. Використовуючи отримані знання я успішно обрахував і відобразив на екрані значення заданого рівняння залежно від значення n, на інтерпретованій мові python.