

Основи програмування – 1. Алгоритми та структури даних

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 13

Виконав студент

ІП-15 Конденко Іван Ігорович
(шифр, прізвище, ім'я, по батькові)

Перевірів(-ла)

Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

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.

Індивідуальне завдання

Варіант 13

Постановка задачі

13. Перетворення значення у двійковій системі числення в десяткове значення.

Математична модель

Змінна	Тип	Ім'я	Призначення
Число в двійковій системі числення	Цілочисельний	num	Вхідні дані
Число в десятковій системі числення	Цілочисельний	p	Вихідні дані
Функція для перетворення	Функція, вихід цілочисельний	convert(Number)	Проміжні дані
Ітератор	Цілочисельний	i	Проміжні дані
Змінна для перетворення числа	Цілочисельний	t	Проміжні дані

// - цілочисельне ділення. % - знаходження остачі від ділення. $\text{pow}(a,b)$ – взведення в степінь, де a – число яке взводиться в степінь, b – сам степінь. Для того, щоб перевести число з двійкової системи числення до десяткової потрібно кожен цифру помножити на 2 в степіні, що відповідає номеру її розряду і сумувати їх. Для цього використаємо рекурсивну функцію convert. Функція convert приймає в себе змінну Number, що відповідає числу в двійковій СЧ. Для переведення знаходимо останню цифру Number за допомогою остачі від ділення на 10, яке множиться на 2 в степіні, що відповідає номеру її розряду та додати функцію convert, яка приймає змінну Number цілочисельно поділене на 10. Функція повертає перетворене число. Початкове значення $i=-1$

Розв'язання

Крок 1. Визначаємо основні дії

Крок 2. Деталізуємо крок виклику функції convert

Крок 3. Деталізуємо крок виведення перетвореного числа

Псевдокод

Крок 1

Початок

Введення num, p

Виклик функції convert

Виведення перетвореного числа

Кінець

Крок 2

Початок

Введення num, p

p = convert(num)

Виведення перетвореного числа

Кінець

Крок 3

Початок

Введення num, p

p = convert(num)

Виведення p

Кінець

Функція convert(Number)

Введення Number, i, t

t = 0

i = -1

якщо

Number == 0

повернути 0

інакше

i++

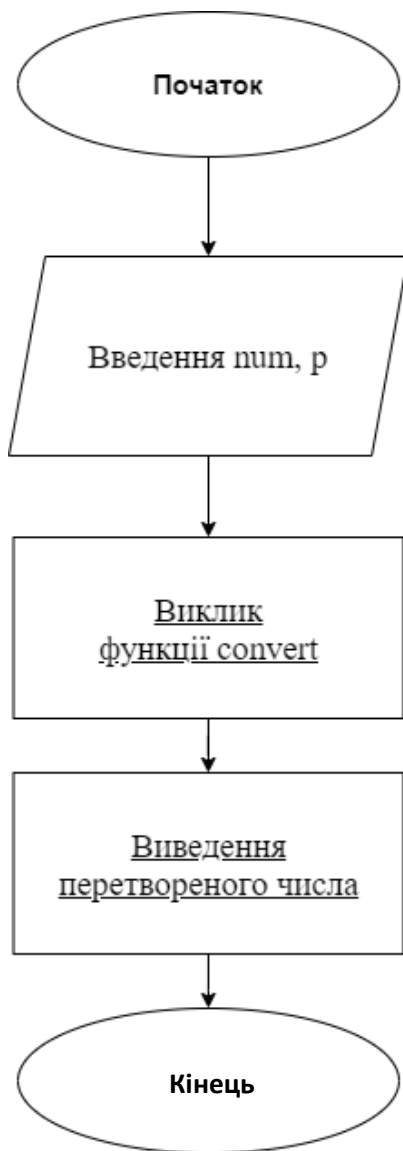
t = (Number % 10)*pow(2, i) + convert(Number // 10)

все якщо

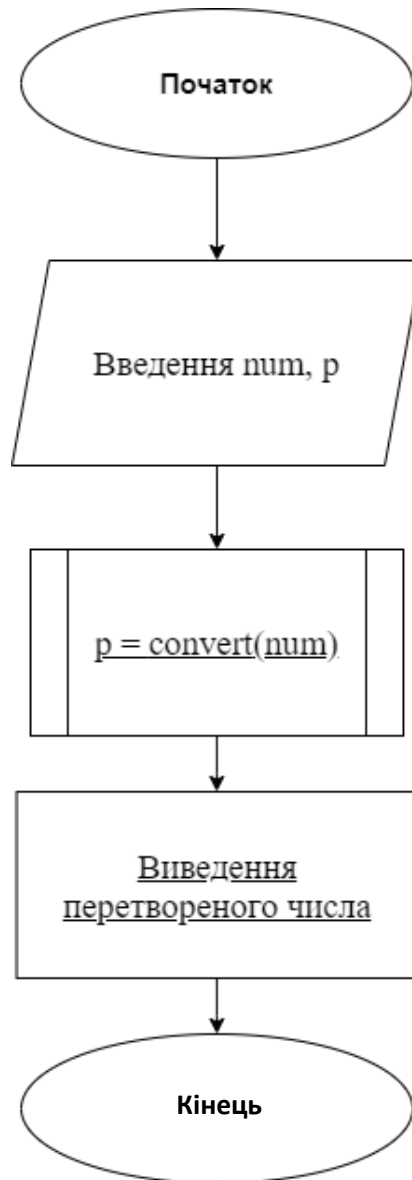
Повернути t

Блок схема

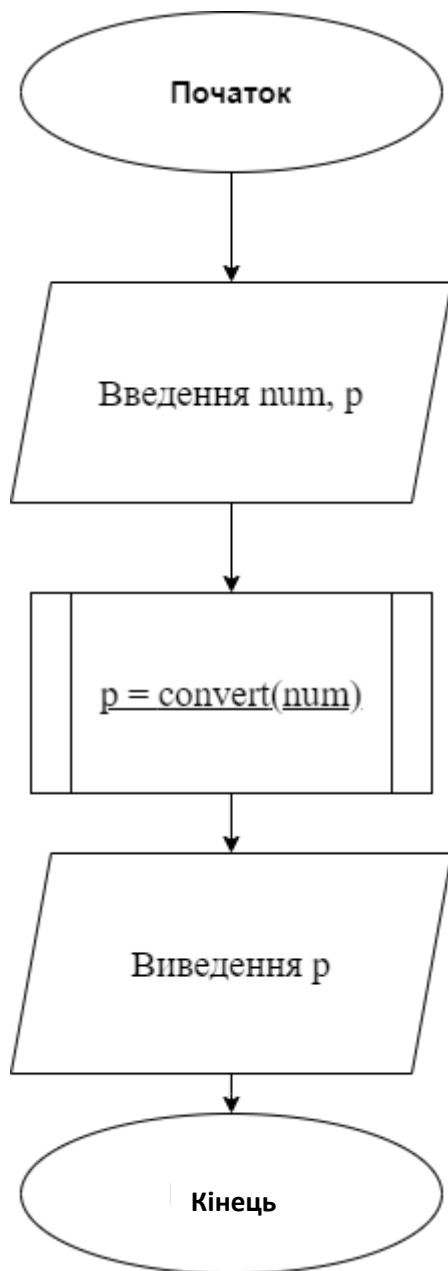
Крок 1



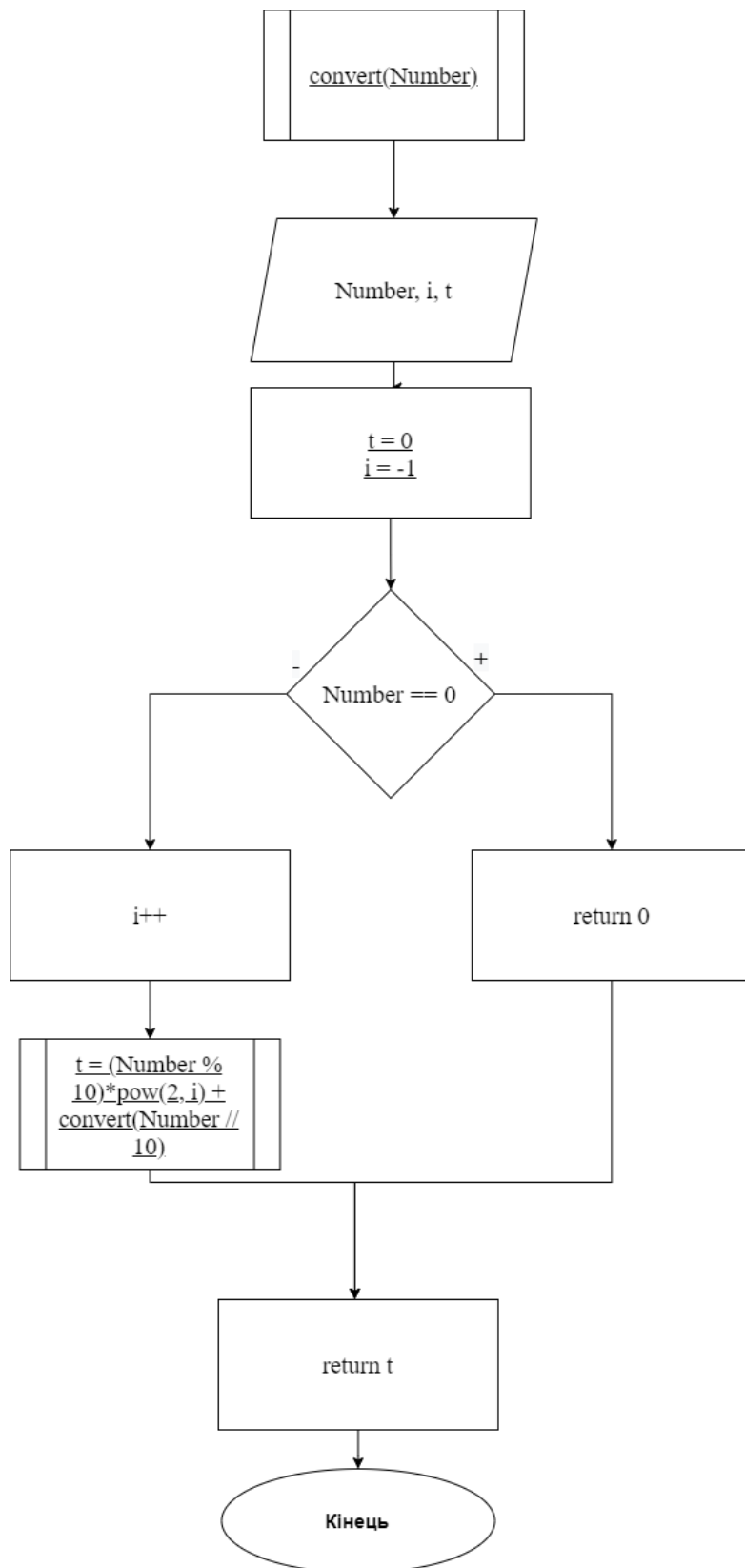
Крок 2



Крок 3



Блок-схема функції `convert(Number)`



Код програми

```
#include <iostream>
#include <cmath>

using namespace std;
int i = -1;

int convert(int Number)
{
    int t = 0;
    if (Number == 0)
        return 0;
    else
    {
        i++;
        t = (Number % 10) * pow(2, i) + convert(Number / 10);
    }
    return t;
}

int main()
{
    int num;
    int p;
    cin >> num;
    p = convert(num);

    cout << p;
}
```

Випробування

Блок	Дія
	Початок
1.	<u>num = 1111</u>
2.	<u>Number = 1111</u>
3.	<u>$t = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 = 1 + 2 + 4 + 8 = 15$</u>
4.	<u>p = 15</u>
	Кінець

Випробування програми

```
1111
15
```

Висновки

Ми дослідили особливості роботи рекурсивних алгоритмів та набули практичних навичок їх використання під час складання програмних специфікацій підпрограм.