
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-13, Жмайло Дмитро Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірила 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

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

Постановка задачі

Задано натуральне число у двійковій системі числення. Необхідно перевести це число у десяткову систему числення, використовуючи механізм рекурсії підпрограми.

Побудова математичної моделі

Відповідно до умови складемо таблицю змінних:

<i>Змінна</i>	<i>Тип</i>	<i>Назва</i>	<i>Призначення</i>
Початкове число n	Цілий	number	Початкові дані
Проміжне двійкове число	Цілий	bin_n	Проміжні дані
Проміжний результат	Цілий	prev_result	Проміжні дані
Результат обчислень	Цілий	result	Вихідні дані

Для розрахунків введемо допоміжні функції:

- Функція знаходження остачі від ділення числа **a** на **b**: **Mod(a, b)**;
- Функція знаходження цілочисельної частки від ділення числа **a** на **b**:

Div(a, b);

Для переведення числа з двійкової системи числення у десяткову достатньо помножити значення розряду числа на 2 в степені номеру самого розряду, починаючи з 0.

Для того, щоб перевести число з двійкової системи числення у десяткову за допомогою рекурсивної формули, необхідно розробити підпрограму **BinToDec**, яка буде:

1. Перевіряти чи параметр функції (число, яке ми вписуємо в тіло функції) рівний нулю:
 - Якщо параметр функції задовольняє умові, то функція присвоює змінній **prev_result** значення **0** (вихід з рекурсії)
 - Якщо параметр функції **не** задовольняє умові, то присвоюємо змінній **prev_result** значення $\text{Mod}(\text{bin_n}, 10) + 2 * \text{BinToDec}(\text{Div}(\text{bin_n}, 10))$, тим самим викликав рекурсію
2. Виводити значення **prev_result**

Розв’язання:

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії;

Крок 2. Деталізуємо дію виклику підпрограми BinToDec для обчислення результату;

Крок 3. Деталізуємо підпрограму BinToDec.

Псевдокод:

Крок 1

початок

введення number

виклик підпрограми для обчислення результату

виведення result

кінець

Функція BinToDec(bin_n)

реалізація підпрограми

Все функція

Крок 2

початок

введення number

result := BinToDec(number)

виведення result

кінець

Функція BinToDec(bin_n)

реалізація підпрограми

Все функція

Крок 3

початок

введення number

result := BinToDec(number)

виведення result

кінець

функція BinToDec(bin_n)

якщо (bin_n == 0)

то

prev_result := 0

все якщо

інакше

prev_result := Mod(bin_n, 10) + 2 * BinToDec(Div(bin_n, 10))

return prev_result

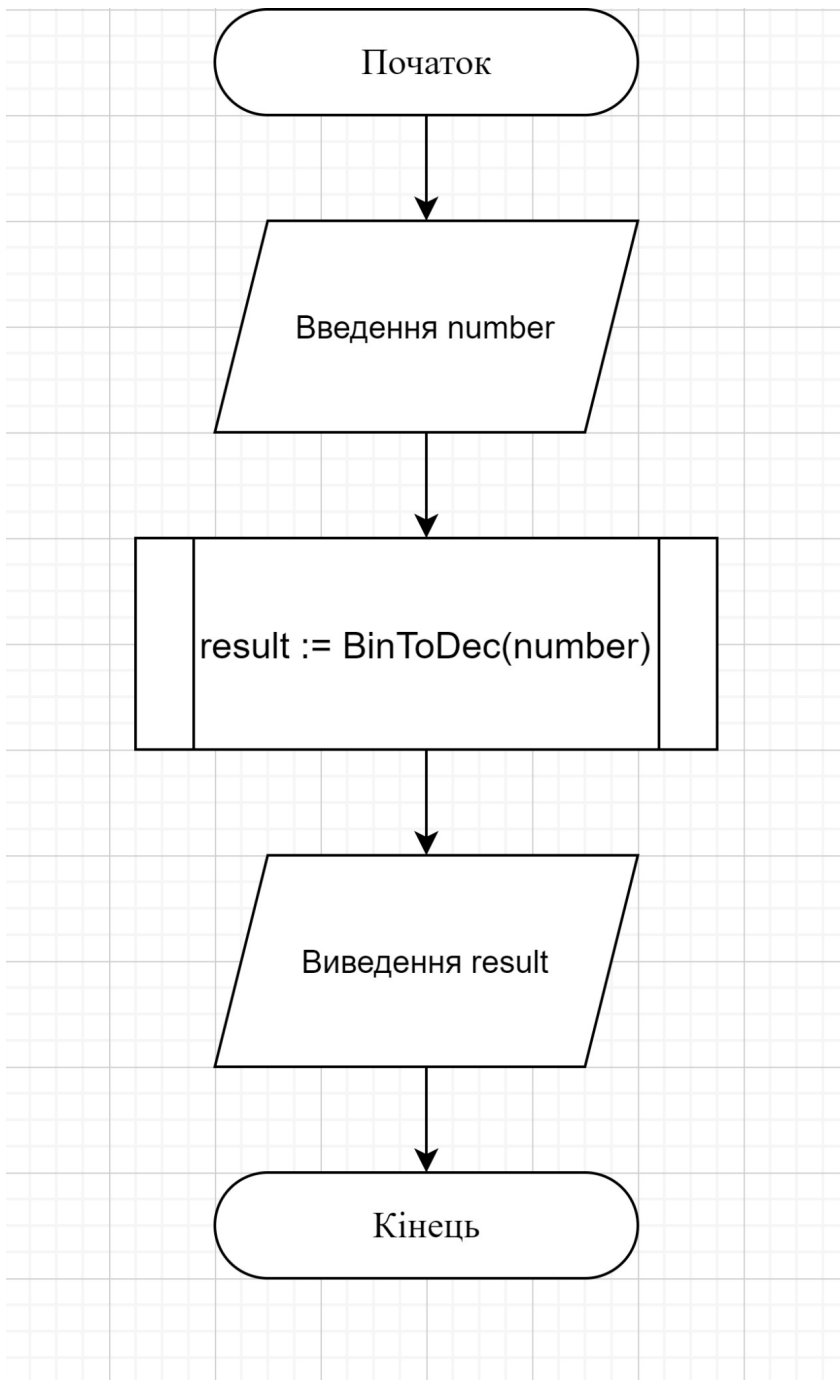
все функція

Блок-схема:

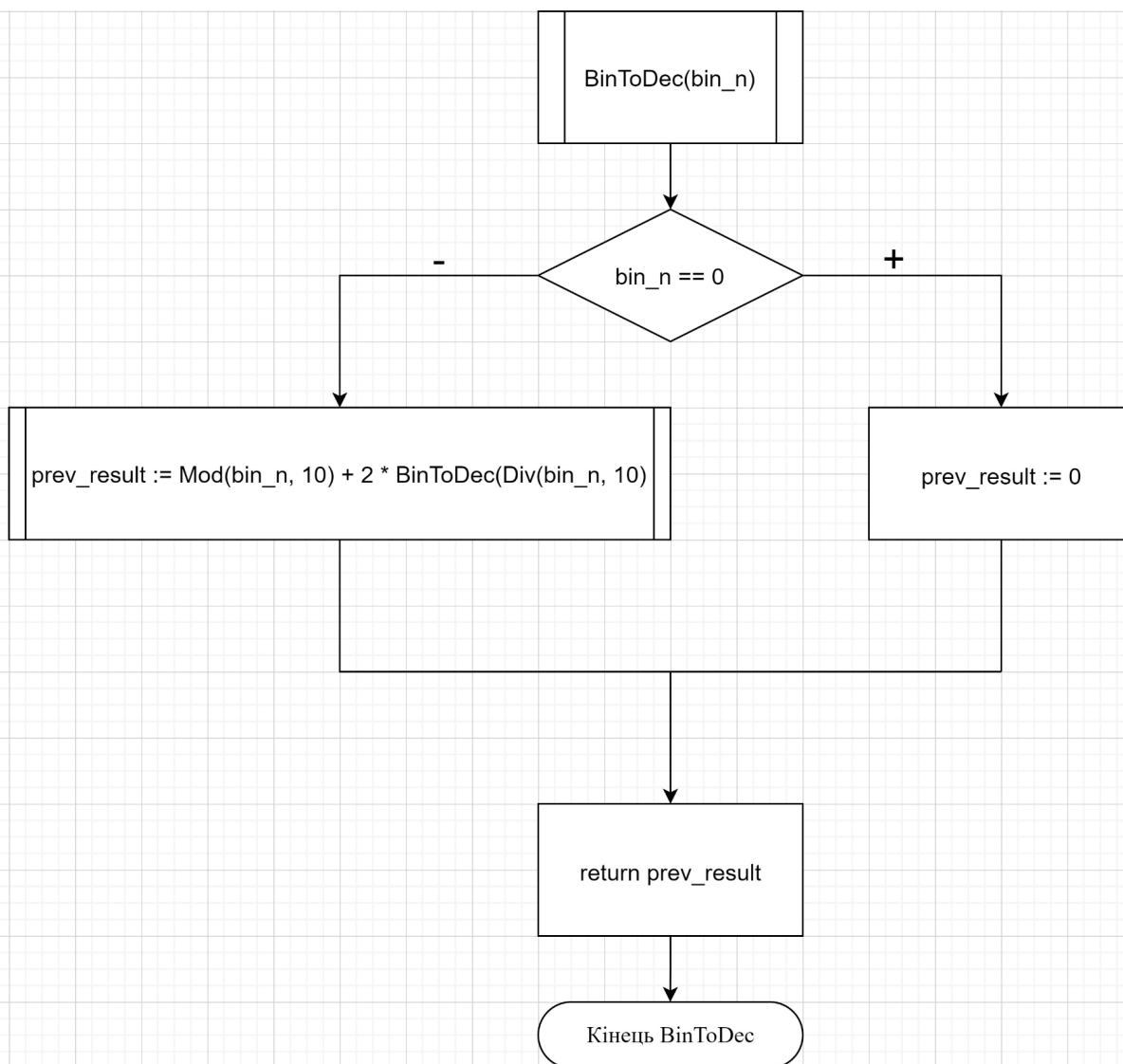
Крок 1



Крок 2



Крок 3



Код програми: (C++)

```
1
2  #include <iostream>
3  using namespace std;
4  int BinToDec(int bin_n);
5  int main()
6  {
7      int number;
8      int result;
9      cout << "Enter the binary value: ";
10     cin >> number;
11     result = BinToDec(number);
12     cout << "Decimal value of binary number is: " << result << endl;
13     return 0;
14 }
15
16 int BinToDec(int bin_n)
17 {
18     if (bin_n == 0)
19         return 0;
20     else
21         return ((bin_n % 10) + 2 * BinToDec(bin_n / 10));
22 }
```

Випробування алгоритму:

Блок	Дія
	Початок
1	number := 101
2	bin_n := 101
3	bin_n == 0 не виконується
4	prev_result := 1 + 2 * BinToDec(10) // Виклик функції
5	bin_n == 0 не виконується
6	prev_result := 0 + 2 * BinToDec(1) // Виклик функції
7	bin_n == 0 не виконується
8	prev_result := 1 + 2 * BinToDec(0) // Виклик функції
9	bin_n == 0 виконується
10	prev_result := 0 return 0
11	prev_result := 1 return 1
12	prev_result := 2 return 2
13	prev_result := 5 return 5
14	result := 5;
15	Виведення 5

Основи програмування – 1. Алгоритми та структури даних

	Кінець
--	--------

Блок	Дія
	Початок
1	number := 11
2	bin_n := 11
3	bin_n == 0 не виконується
4	prev_result := 1 + 2 * BinToDec(1) // Виклик функції
5	bin_n == 0 не виконується
6	prev_result := 1 + 2 * BinToDec(0) // Виклик функції
7	bin_n == 0 виконується
8	prev_result := 0 return 0
9	prev_result := 1 return 1
10	prev_result := 3 return 3
11	result := 3;
12	Виведення 3
	Кінець

Висновок: На цій лабораторній роботі ми дослідили особливості роботи рекурсивних алгоритмів та набули практичних навичок їх використання під час складання програмних специфікацій підпрограм; навчилися складати блок-схеми підпрограм, робити псевдокод рекурсивних алгоритмів. Покроково перевірили виконання алгоритму та за допомогою нього склали програму на мові C++.