

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 29

Виконав студент ПІ-12 Скорик Родіон Олегович
(шифр, прізвище, ім'я, по батькові)

Перевірів _____
(прізвище, ім'я, по батькові)

Лабораторна робота 6

Дослідження рекурсивних алгоритмів

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

Варіант 29

Індивідуальне завдання. Перетворення натурального числа з десяткової системи числення у двійкову.

Розв'язання

Постановка задачі

Вхідні данні – натуральне число у 10-вій системі числення. Для обчислення інших даних не потрібно, необхідно використати рекурсивний алгоритм. Результат обчислень – те саме число, представлене у двійковій системі числення.

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Натуральне число	Ціле	n	Початкове дане
Остача від ділення на 2	Ціле	ost	Проміжні розрахунки
Знаходження остачі від ділення	Оператор, операнди – цілі числа	%	Проміжні розрахунки
Цілочисельне ділення	Оператор, операнди – цілі числа	/	Проміжні розрахунки
Знаходження бінарного представлення числа	Функція, аргумент – ціле число	find_bin_number()	Знаходження та вивід результату

Побудова математичної моделі

За основу візьмемо спосіб переаєднення цілого числа методом ділення на основу. Необхідно ділити число на основу нової системи числення, доки воно більше за неї, та виводити остачу від ділення у зворотньому порядку. Рекурсивно реалізуємо це так: якщо $n > 1$ (тобто $n \geq 2$), викликаємо функцію від $n/2$, де / - цілочисельне ділення. Далі виводимо та знаходимо остачу від ділення n на 2, без пробілів. Таким чином, кожен наступний виклик знаходитиме остачку від ділення n , поділеного на 2. Оскільки першим відбудеться виведення від останнього

виклику, то остачі будуть одразу виведені у зворотньому порядку. У основній програмі достатньо реалізувати виклик рекурсивної функції, яка прийме як аргумент уведене число.

Псевдокод

Крок 1

початок

ввід n

знаходження двійкового представлення числа

кінець

Крок 2

початок

ввід n

find_bin_number(n)

кінець

Підпрограма find_bin_number():

find_bin_number(n)

якщо $n > 1$

find_bin_number($n/2$)

ost = $n \% 2$

вивід ost

Блок-схеми алгоритму

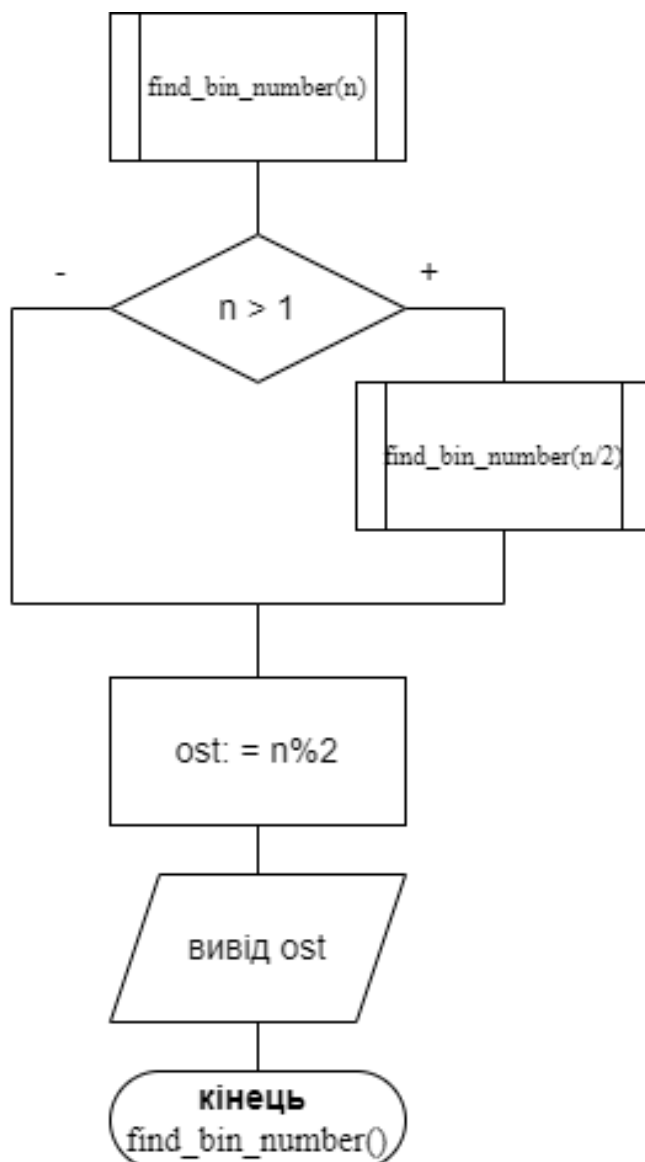
Крок 1



Крок 2



Підпрограма find_bin_number()



Перевірка

Блок	Дія	
	Початок	Початок
1	Ведення 4	Ведення 3
2	find_bin_number(4)	find_bin_number(3)
3	$4 > 1$ - істина	$3 > 1$ - істина
4	find_bin_number(2)	find_bin_number(1)
6	$2 > 1$ – істина	$1 > 1$ – хиба
6	find_bin_number(1)	$ost = 1 \% 2 = 1$
7	$1 > 1$ – хиба	Виведення: 1
8	$ost = 1 \% 2 = 1$	$ost = 3 \% 2 = 1$
9	Виведення: 1	Виведення: “1 ”
10	$ost = 2 \% 2 = 0$	Кінець
11	Виведення: 0	
12	$ost = 4 \% 2 = 1$	
13	Виведення: 1	
	Кінець	

Код програми

```
#include <iostream>
using namespace std;

void find_bin_number(long long n)
{
    if (n > 1)
        find_bin_number(n / 2);
    int ost = n % 2;
    cout << ost;
}

int main()
{
    long long n;
    cout << "n = ";
    cin >> n;
    cout << "n in binary numeral system: ";
    find_bin_number(n);
}
```

Випробування програми

```
n = 3
n in binary numeral system: 11
C:\Users\ROS\source\repos\ASD labs\Debug\lab6.exe (process 7100) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debug
le when debugging stops.
Press any key to close this window . . .
```

```
n = 0
n in binary numeral system: 0
C:\Users\ROS\source\repos\ASD labs\Debug\lab6.exe (process 12992)
To automatically close the console when debugging stops, enable T
le when debugging stops.
Press any key to close this window . . .
```

```
n = 1024
n in binary numeral system: 10000000000
C:\Users\ROS\source\repos\ASD labs\Debug\lab6.exe (process 7532) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debug
le when debugging stops.
Press any key to close this window . . .
```

Висновок

Під час виконання роботи було досліджено особливості роботи рекурсивних алгоритмів та набуто практичних навичок їх використання під час складання програмних специфікацій підпрограм. Особливістю реалізації знаходження

бінврного представлення числа через просту рекурсію є зокрема можливість виведення без використання додаткових структур даних, наприклад рядків, та простота запису.