

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 7

Виконав студент ІІ-13, Гогіберідзе Торніке Лашаєвич

(шифр, прізвище, ім'я, по батькові)

Перевірила Вєчерковська 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.

Варіант 7

7. Перетворення додатного цілого десяткового значення в значення у вісімковій системі числення

Постановка задачі

Використовуємо рекурсивну функцію, щоб перетворити десяткове число у вісімкову систему числення. Функція приймає додатне ціле число, і якщо воно не менше 8, то виконується рекурсивна гілка, яка обчислює менший розряд числа. Як тільки функції рекурсивно доходить до числа менше 8, виконується термінальна гілка та повертається це ж число. Таким чином значення числа у вісімковій системі числення обчислюється порозрядно.

Результатом розв'язку є перетворення та виведення значення десяткового числа у вісімковій системі числення.

Побудова математичної моделі

Таблиця імен змінних

Змінна	Тип	Ім'я	Призначення
Задане число	Додатне ціле	number	Початкові дані

Функція, що перетворює десяткове число у вісімкове	Функція	toOctal	Початкові дані
Параметр функції	Додатне ціле	n	Початкові дані /Проміжні дані
Змінна, яку повертає функція	Додатне ціле	r	Початкові дані /Проміжні дані
Значення числа у вісімковій системі числення	Додатне ціле	result	Результат

Після введення числа, визиваємо рекурсивну функцію `toOctal`, яка й перетворить наше десяткове число у вісімкове. Ця функція приймає один параметр типу `unsigned int` (додатне число), яким і буде задане число. Термінальна гілка функції виконується тільки, якщо число-параметр менше 8; тоді значення самого числа записується в змінну `r`. В інших випадках виконується рекурсивна гілка, яка записує в змінну `r` вираз: $(n \% 8) + (10 * \text{toOctal}(n / 8))$, де `%` – функція для знаходження залишку від ділення, а `/` – ціличисельне ділення. Таким чином, рекурсивна функція `toOctal` дойде до числа менше 8, що й буде найбільшим розрядом заданого числа у вісімковій системі числення. Потім функція буде додавати кожен розряд помножений на степінь десяти. Перетворення закінчиться, коли функція повернеться до заданого числа, і вираз `n \% 8` буде найменшим розрядом заданого числа у вісімковій системі числення. Вкінці `toOctal` повертає значення `r`, і основна програма записує його в змінну `result` та виводить цю змінну.

Розв'язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Обчислення result за допомогою функції toOctal

Псевдокод

Основна програма:

Крок 1 Крок 2 початок початок

ввід number

ввід number

Обчислення result за **result := toOctal (number)** допомогою функції

toOctal **виведення result**

виведення result **кінець** **кінець**

Підпрограма: toOctal

(n)

якщо n < 8

то

r := n

інакше

r := (n % 8) + (10 * toOctal (n / 8))

повернути r

кінець

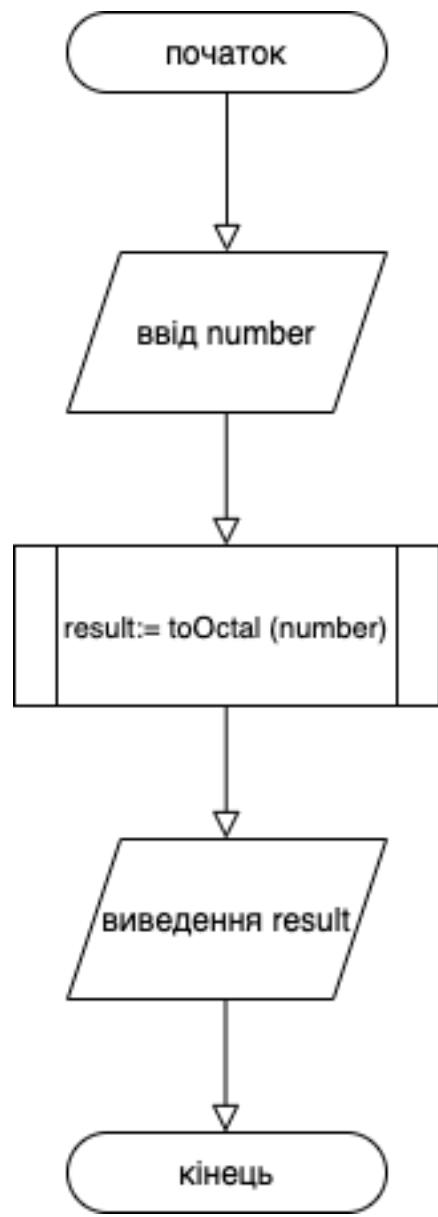
Блок-схема

Основна програма:

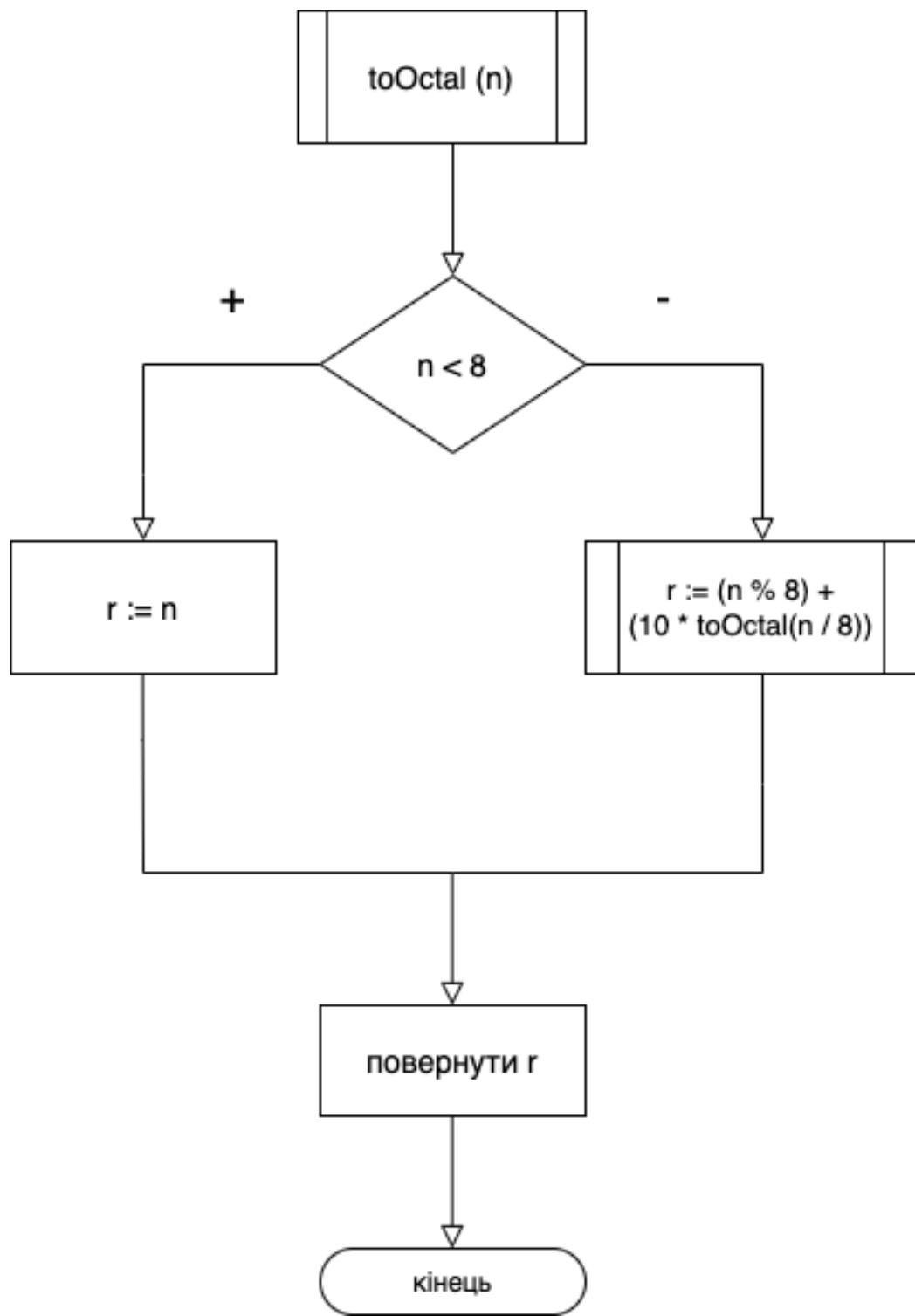
Крок 1.



Крок 2.



Пiдпрограма:



Код:

```
1 #include <iostream>
2 using namespace std;
3
4 unsigned toOctal(unsigned n);
5
6
7 int main()
8 {
9     unsigned number;
10
11    cout << "Enter your number: ";
12    cin >> number;
13
14    unsigned result = toOctal(number);
15    cout << result << endl;
16
17    return 0;
18 }
19
20 unsigned toOctal(unsigned n) {
21
22     unsigned r;
23
24     if (n < 8) {
25         r = n;
26     }
27     else {
28         r = (n % 8) + (10 * toOctal(n / 8));
29     }
30
31     return r;
32 }
```

Тестування

Блок	Дія (основна програма)	Дія (підпрограма)
	Початок	
1	number = 496	
2	result = toOctal(496)	
3		$496 < 8 \rightarrow \text{false}$
4		$r := 496 \% 8 + (10 * \text{toOctal}(496 / 8)) = 0 + 10 * \text{toOctal}(62)$
5		$62 < 8 \rightarrow \text{false}$
6		$r := 62 \% 8 + (10 * \text{toOctal}(62 / 8)) = 6 + 10 * \text{Octal}(7)$
7		$7 < 8 \rightarrow \text{true}$
8		$r := 7$
9		Повернути 7
10		$r := 6 + 10 * 7 = 76$
11		Повернути 76
12		$r := 0 + 10 * 76 = 760$
13		Повернути 760
14	result = 760	
15	Виведення 760	
	Кінець	

Висновки

Протягом шостої лабораторної роботи я дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм. В результат я отримав алгоритм, що використовує рекурсивну підпрограму для переведення числа з десяткової системи числення у вісімку.