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 ІП-11 Тарасьонок Дмитро Євгенович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

( 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 29**

Перетворення натурального числа з десяткової системи числення у двійкову

1. **Постановка задачі:**

Постановка задачі зводиться до того, що нам необхідно організувати підпрограму, в якій поступово обчислюватиметься кожна наступна цифра двійкового представлення числа.

# Побудова математичної моделі. Складемо таблицю імен змінних:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Натуральне число | Цілий | num | Вхідні дані |
| Остача від ділення на 2 | Цілий | last | Проміжні дані |

Для вирішення даної задачі необхідно організувати підпрограму, в як яку буде передаватися спочатку задане число, а потім, поки воно не буде рівним нулю, - щоразу нове число, отримане діленням числа на 2 й обраховуватиметься остача від ділення на 2. Коли ж число дорівнюватиме нулю, рекурсія закінчиться й почне поступово виходити з неї, виводячи на екран кожну цифру, що отримана на кожному етапі рекурсії, починаючи з останнього, оскільки при отриманні остачі від ділення на 2 цифри двійкового виду числа записуються у зворотному напрямку

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію переведення числа у двійкову систему

1. **Псевдокод:**

*Крок 1*

# Початок

Введення num

Переведення num у двійкову систему

**Кінець**

*Крок 2*

# Початок

Введення num

binary(num)

**Кінець**

**Початок binary(num)**

last := num mod 2

**якщо num не дорівнює 0**

**то**

binary(num / 2)

виведення last

**інакше**

повернути

**все якщо**

**Кінець binary(num)**

**4. Блок-схема:**

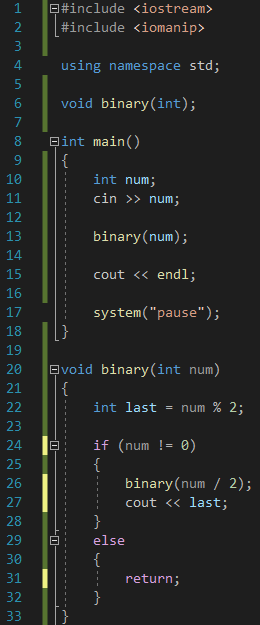


# 5. Випробування алгоритму:

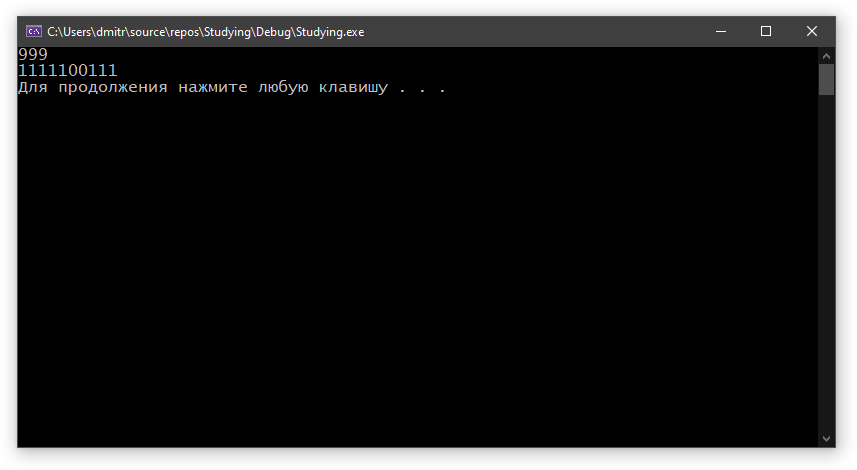
|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | **Початок** |
| **1** | Введення num = 999 |
| **2.1** | Виведення 1 |
| **2.2** | Виведення 1 |
| **2.3** | Виведення 1 |
| **2.4** | Виведення 1 |
| **2.5** | Виведення 1 |
| **2.6** | Виведення 0 |
| **2.7** | Виведення 0 |
| **2.8** | Виведення 1 |
| **2.9** | Виведення 1 |
| **2.10** | Виведення 1 |
|  | **Кінець** |

|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | **Початок** |
| **1** | Введення num = 255 |
| **2.1** | Виведення 1 |
| **2.2** | Виведення 1 |
| **2.3** | Виведення 1 |
| **2.4** | Виведення 1 |
| **2.5** | Виведення 1 |
| **2.6** | Виведення 1 |
| **2.7** | Виведення 1 |
| **2.8** | Виведення 1 |
|  | **Кінець** |

1. **Код програми мовою C++:**



1. **Виконання програми мовою C++:**



# 

# 8. Висновки:

За допомогою математичної моделі та підпрограм можна запрограмувати перетворення чисел з десяткової системи до двійкової. Для виконання поставленої задачі я використав рекурсивну функцію з однією рекурсивною та однією термінальною гілками: якщо число дорівнює 0, рекурсія закінчується, інакше - продовжується. Я здобув достатніх навичок для створення відповідних програм та навчився чітко розрізняти, де необхідно використовувати рекурсію та визначати умови термінальних гілок.