

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 27

Виконав студент ІІ-13 Пархомчук Ілля Вікторович
(шифр, прізвище, ім'я, по батькові)

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

Київ 2021

Лабораторна робота 5

Дослідження рекурсивних алгоритмів

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

Варіант 27.

Перетворення значення у двійковій системі числення в шістнадцятирічне значення.

Постановка задачі

Вхідними даними є число у двійковій системі числення. Потрібно за допомогою алгоритму перевести дане число у десяткову систему числення, а потім перевести з десяткової в шістнадцятирічне.

Побудова математичної моделі

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Число представлене у двійковій системі числення	Ціличесельний	Num	Вихідні дані
Число переведене у десяткову систему числення	Ціличесельний	Num_b10	Проміжні дані
Спеціальне допоміжне число для переведення в десяткову систему	Ціличесельний	b_10	Проміжні дані
Лічильник	Ціличесельний	i	Проміжні дані
Остача від ділення	Ціличесельний	rem	Проміжні дані
Частка від ділення	Ціличесельний	quot	Проміжні дані
Цифра числа	Символьний	c	Вихідні дані

Таблиця аргументів підпрограм

Число в бінарному представленні для переведення в десяткове	Ціличисельний	num_bin	Проміжні дані
Число в десятковому представленні для переведення в шістнадцятиричне	Ціличисельний	num_bin	Проміжні дані
Цифра числа в десятковій системі числення для переводу в шістнадцятиричну	Ціличисельний	dig	Проміжні дані

Таблиця підпрограм

Підпрограма	Оголошення	Синтаксис	Опис
to_DEC	to_DEC(num_bin)	to_DEC(a)	Перетворює та повертає число <i>a</i> з бінарного представлення в десятковому
to_HEX	to_HEX(num_dec)	to_HEX(a)	Перетворює число <i>a</i> в десятковому представленні в шістнадцятиричне
get_digit	get_digit(dig)	get_digit(a)	Переводить та виводить цифру <i>a</i> з десяткового представлення в шістнадцятиричний алфавіт

Таблиця операторів

Оператор	Назва	Синтаксис	Опис
log10	Логарифм за основою 10	log10(a)	Повертає Логарифм за основою 10 числа a
pow	Піднесення до степеню	pow(a,b)	Піднести a в степінь b
mod	Ділення з остачею	a mod b	Остача від ділення a на b
div	Цілочисельне ділення	a div b	Частка від ділення a на b

Вхідними даними є число у бінарному представленні. Спочатку його потрібно перевести у десяткову систему шляхом множення та додавання n-ої цифри на $2^n \cdot a_n$, де k – кількість цифр в бінарному представленні.

Далі потрібно перевести з десяткового представлення в шістнадцятиричне шляхом рекурсивного ділення числа на 16 та зворотного запису остач від ділення.

Оскільки залишки від ділення потрібно записувати в зворотному порядку, а рекурсія завершується при останньому діленні, то вигляд числа буде зразу в нормальному вигляді.

Для представлення чисел від 10 до 15 використовуються спеціальні символи A, B, C, D, E, F відповідно.

Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо переведення з бінарної системи числення в десяткову за допомогою підпрограми з використанням арифметичного оператору повторення.

Крок 3. Деталізуємо переведення з десяткової системи числення в шістнадцятиричну за допомогою підпрограми з використанням умовної форми вибору.

Крок 4. Деталізуємо вивід цифри числа в шістнадцятиричній системі у нормальній формі за допомогою підпрограми з використанням альтернативної форми вибору.

Псевдокод алгоритму

крок 1

Основна програма

початок

ввід Num

Num_b10 = to_DEC(Num)

to_HEX(Num_b10)

кінець

Підпрограми

to_DEC(num_bin)

переведення з бінарної системи числення в десяткову

to_HEX(num_dec)

переведення з десяткової системи числення в шістнадцятирічну

get_digit(dig)

вивід цифри числа в шістнадцятирічній системі у нормальній формі

крок 2

Основна програма

початок

ввід Num

Num_b10 = to_DEC(Num)

to_HEX(Num_b10)

кінець

Підпрограми

to_DEC(num_bin)

b_10 := 0

для i **від** 0 **до** log10(num_bin) + 1, з **кроком** 1 **повторити**

b_10 := b_10 + ((num_bin **div** pow(10, i)) **mod** 10) * pow(2, i)

все повторити

повернути b_10

кінець

to_HEX(num_dec)

переведення з десяткової системи числення в шістнадцятиричну

get_digit(dig)

вивід цифри числа в шістнадцятиричній системі у нормальній формі

крок 3

Основна програма

початок

ввід Num

Num_b10 = to_DEC(Num)

to_HEX(Num_b10)

кінець

Підпрограми

to_DEC(num_bin)

b_10 := 0

для i **від** 0 **до** log10(num_bin) + 1, з **кроком** 1 **повторити**

b_10 := b_10 + ((num_bin **div** pow(10, i)) **mod** 10) * pow(2, i)

все повторити

повернути b_10

кінець

to_HEX(num_dec)

rem = num_hex **mod** 16;

quot = num_hex **div** 16;

якщо quot != 0

то

to_HEX(num_hex / 16);

все якщо

get_digit(rem);

кінець

get_digit(dig)

вивід цифри числа в шістнадцятирічній системі у нормальній формі

крок 4
Основна програма

початок

ввід Num

Num_b10 := to_DEC(Num)

To_HEX(Num_b10)

кінець

Підпрограми

to_DEC(num_bin)

b_10 := 0

для i **від** 0 **до** log10(num_bin) + 1, з **кроком** 1 **повторити**

b_10 := b_10 + ((num_bin **div** pow(10, i)) **mod** 10) * pow(2, i)

все повторити

повернути b_10

кінець

to_HEX(num_dec)

rem := num_hex **mod** 16

quot := num_hex **div** 16

якщо quot != 0

то

to_HEX(quot)

все якщо

get_digit(rem);

кінець

get_digit(dig)

якщо $dig < 10$

то

вивід dig

інакше

якщо $dig == 10$

то

$c := 'A'$

інакше якщо $dig == 11$

то

$c := 'B'$

інакше якщо $dig == 12$

то

$c := 'C'$

інакше якщо $dig == 13$

то

$c := 'D'$

інакше якщо $dig == 14$

то

$c := 'E'$

інакше якщо $dig == 15$

то

$c := 'F'$

все якщо

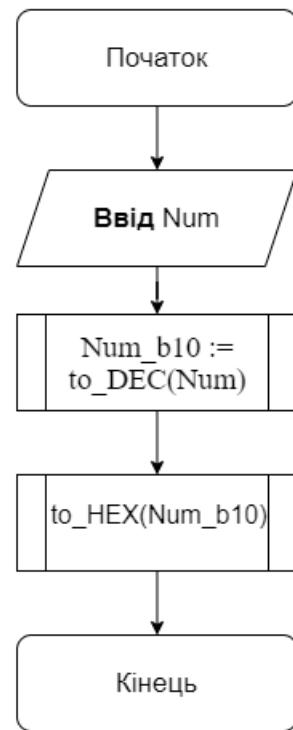
вивід c

все якщо

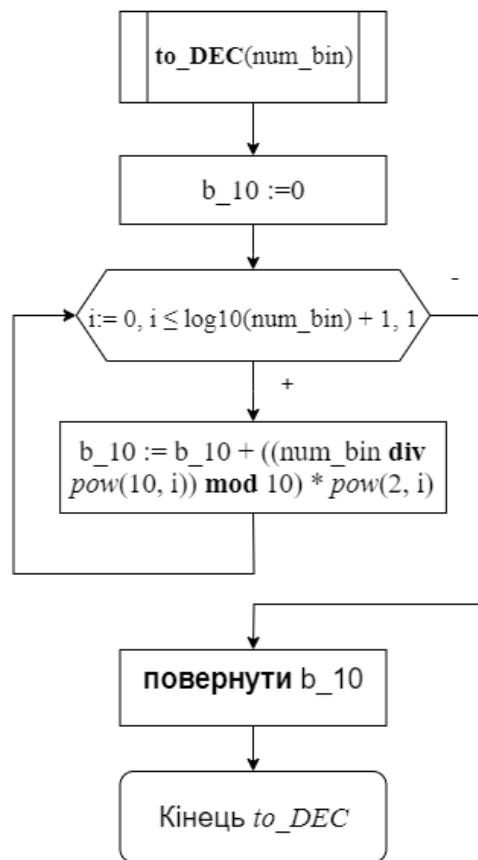
кінець

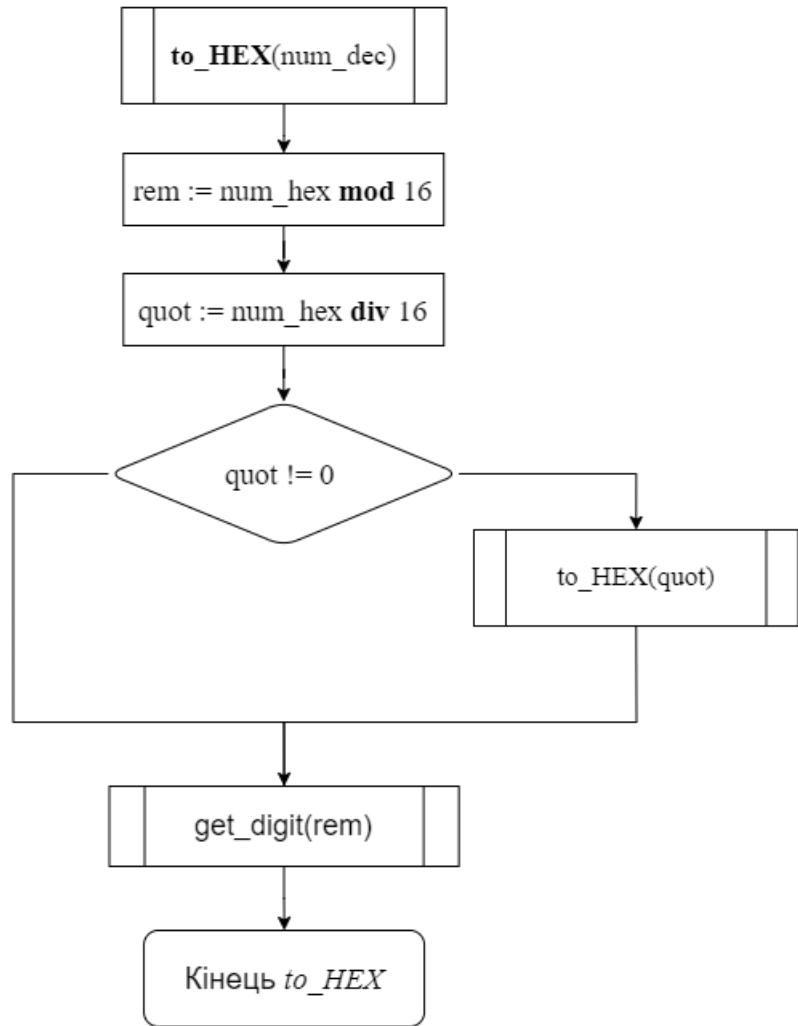
Блок-схема

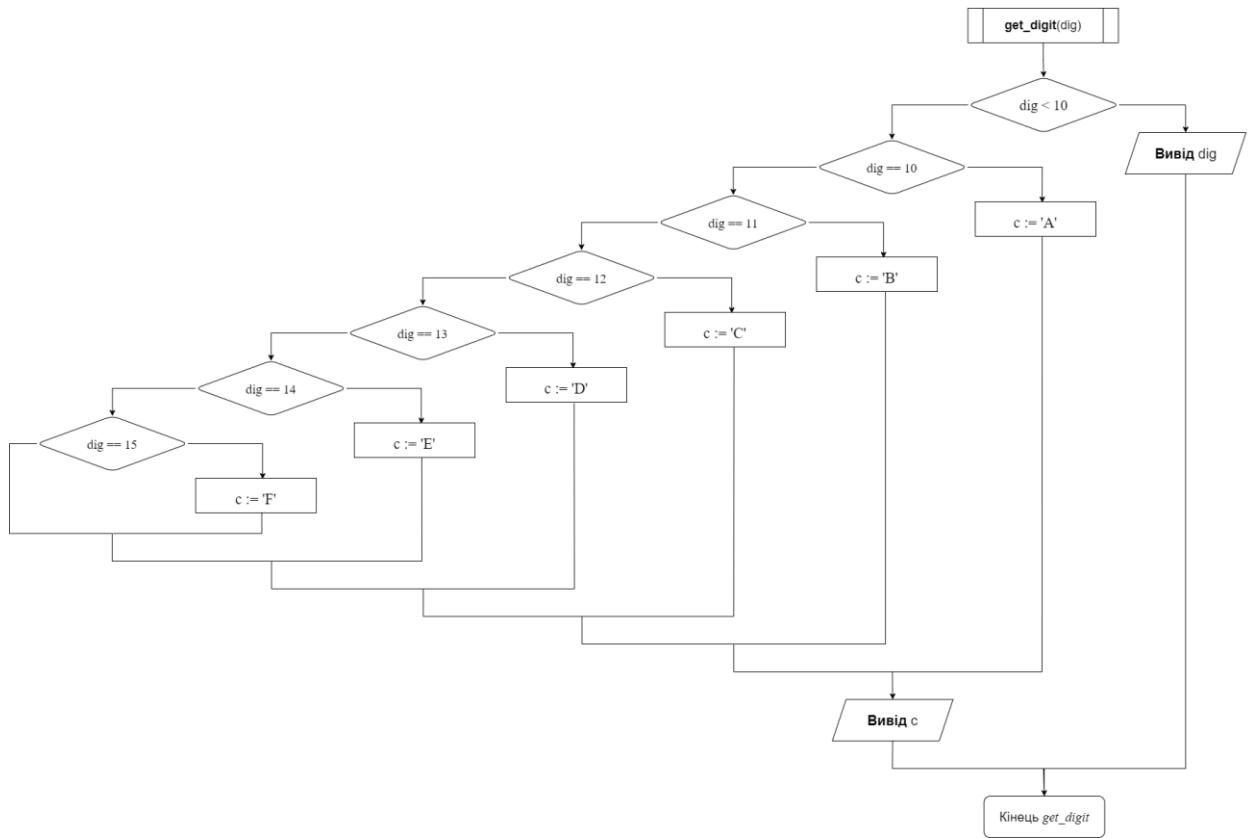
Основна програма:



Підпрограми:







Код программы

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

using std::cout;
using std::cin;
using std::endl;

int to_DEC(long long num_bin);
void to_HEX(int num_hex);
void get_digit(int dig);

int main()
{
    long long Num;
    cout << "Enter Num in binary: ";
    cin >> Num;
    long long Num_b10;
    Num_b10 = to_DEC(Num);
    cout << "Num in heximal: ";
    to_HEX(Num_b10);
    cout << '\n';
    system("pause");
    return 0;
}

int to_DEC(long long num_bin)
{
    int b_10 = 0;
    for (int i = 0; i < log10(num_bin) + 1; ++i) {
        b_10 += (long long(num_bin / pow(10, i)) % 10) * pow(2, i);
    }
    return b_10;
}

void to_HEX(int num_hex)
{
    int rem = num_hex % 16;
    int quot = num_hex / 16;
    if( quot != 0) to_HEX(quot);
    get_digit(rem);
}

void get_digit(int dig)
{
    char c;
    if (dig < 10) cout << dig;
    else
    {
        if (dig == 10) c = 'A';
        else if (dig == 11) c = 'B';
        else if (dig == 12) c = 'C';
        else if (dig == 13) c = 'D';
        else if (dig == 14) c = 'E';
        else if (dig == 15) c = 'F';
        cout << c;
    }
}
```

Випробування алгоритму

```

Enter Num in binary: 11100
Num in heximal: 1C
Press any key to continue . . .

```

Блок	Дія
Початок	
1	Ввід Num := 11100
2	Num_b10 = to_DEC(11100)
3	b_10 := 0
4	b_10 := 0 + 0 * 1 i = 0, < 5
5	b_10 := 0 + 0 * 2 i = 1, < 5
6	b_10 := 0 + 1 * 4 i = 2, < 5
7	b_10 := 4 + 1 * 8 i = 3, < 5
8	b_10 := 12 + 1 * 16 i = 4, < 5
9	повернути 28
10	to_HEX(28)
11	rem := 28 mod 16
12	quot := 28 div 16
13	(1 != 0) = true
14	to_HEX(1)
15	rem := 1 mod 16
16	quot := 1 div 16

17	$(0 \neq 0) = \text{false}$
18	<code>get_digit(1)</code>
19	$(1 < 10) = \text{true}$
20	Вивід: 1
21	<code>get_digit(12)</code>
22	$(12 < 10) = \text{false}$
23	$(12 == 10) = \text{false}$
25	$(12 == 11) = \text{false}$
25	$(12 == 12) = \text{true}$
26	<code>c := 'C'</code>
27	Вивід: C
Кінець	

Висновки

На лабораторній роботі я дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм. Спочатку я створив підпрограму для переведу числа з двійкової системи числення в десяткову. Потім створив підпрограму для переведу числа з десяткової в шістнадцяткову за допомогою рекурсивного ділення. Також створена підпрограма, яка переводить цифри в алфавіт шістнадцяткової системи числення. Мною був написаний псевдокод, створено блок-схеми та проведено тестування алгоритму.