

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 ІП-11 Савенко Олексій Андрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів Мартінова О. П.
(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 27

Індивідуальне завдання

27. Перетворення значення у двійковій системі числення в шістнадцятирічне значення.

Постановка задачі

Результатом виконання завдання є переведення значення числа з двійкової системи у шістнадцяткову систему числення. Нам потрібно виконати послідовне переведення з однієї системи в іншу, як трансферна система числення буде використана десяткова. Отримання значень у відповідних функціях буде відбуватися за допомогою механізму рекурсії – самовиклику функції. Перша функція **GetCount()** буде підраховувати кількість знаків у числі цілого, після виконання своєї роботи поверне своє значення у головну функцію **main()**, де його отримає змінна **Countofdig**, після цього за рахунок передання цього значення як фактичного параметру до функції **GetDEC()** буде отримано число у десятковій системі числення, його значення отримає змінна **DecNum**, відбудеться передання цієї змінної як фактичного параметру до функції **GetHEX()**, яка надасть нам значення числа в шістнадцятковій системі числення. Після чого її значення буде отримано змінною **HexNum** та буде виведений відповідний результат значення цієї змінної.

Математична модель

main()

Змінна	Тип	Ім'я	Призначення
Число у двійковому вигляді	Цілий	Num2	Вхідні дані
Початкове значення для підрахунку к-сті цифр двійкового числа	Цілий	lCountofdig	Проміжні дані
Початкове значення для розрахунку десяткового числа	Цілий	IDecNum	Проміжні дані
Початковий рядок для застосування знаходження шістнадцяткового числа	Рядок	lHexNum	Проміжні дані
Число, що приймає та передає результат розрахунку к-сті цифр двійкового числа	Цілий	Countofdig	Проміжні дані
Отримане число у десятковому	Цілий	DecNum	Проміжні дані

вигляді для розрахунку шістнадцяткового числа			
Знайдене число у шістнадцятковому вигляді	Рядок	HexNum	Вихідні дані

GetCount()

Змінна	Тип	Ім'я	Призначення
Число у двійковому вигляді	Цілий	a	Вхідні дані, формальний параметр функції
Початкове значення для підрахунку к-сті цифр двійкового числа	Цілий	b	Вихідні дані, формальний параметр функції

GetDEC()

Змінна	Тип	Ім'я	Призначення
Число у двійковому вигляді	Цілий	a	Вхідні дані, формальний параметр функції
Кількість цифр у числі	Цілий	b	Вхідні дані, формальний

			параметр функції
Початкове значення для розрахунку десятькового числа	Цілий	c	Вихідні дані, формальний параметр функції

GetHEX()

Змінна	Тип	Ім'я	Призначення
К-сть символів кожного числа	Цілий	a	Вхідні дані, формальний параметр функції
Початковий рядок для знаходження шістнадцятькового числа	Рядок	b	Вихідні дані, формальний параметр функції
Зберігач значення десятькового числа	Цілий	a1	Проміжні дані
Проміжна змінна, використовується для перевірки на подільність число	Цілий	a3	Проміжні дані
Отримує відповідне значення символу з шістнадцятькової системи числення	Символ	a4	Проміжні дані

Функції

Підпрограма	Тип	Синтаксис	Призначення
GetCount()	Цілий	GetCount(a,b)	Визначення к-сті цифр двійкового числа
GetDEC()	Цілий	GetDEC(a,b,c)	Перетворення двійкового числа на десяткове
GetHEX()	Цілий	GetHEX(a,b)	Перетворення десяткового число на шістнадцяткове

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію знаходження кількості елементів цілого двійкового числа.

Крок 3. Деталізуємо дію переведення числа з двійкової в десяткову систему числення.

Крок 4. Деталізуємо дію переведення числа з десяткової в шістнадцяткову системи числення.

Псевдокод алгоритму:

Крок 1.

Початок

Введення Num2, ICountofdig = 0, IDecNum = 0, IHexNum, Countofdig, DecNum,
HexNum

знаходження кількості елементів цілого двійкового числа

переведення числа з двійкової в десяткову системи числення

переведення числа з десяткової в шістнадцяткову системи числення

Виведення HexNum

Кінець

Крок 2.

Початок

введення Num2, ICountofdig = 0, IDecNum = 0, IHexNum, Countofdig, DecNum,
HexNum

Countofdig= GetCount(Num2, ICountofdig)

переведення числа з двійкової в десяткову системи числення

переведення числа з десяткової в шістнадцяткову системи числення

виведення HexNum

Кінець

GetCount(a, b)

якщо $a > 0$

то

$a = a / 10$

$b++$

повернути GetCount(a,b)

все якщо

інакше

повернути b

Кінець

Крок 3.

Початок

введення Num2, lCountofdig = 0, lDecNum = 0, lHexNum, Countofdig, DecNum,
HexNum

$\text{Countofdig} = \text{GetCount}(\text{Num2}, \text{lCountofdig})$

$\text{DecNum} = \text{GetDEC}(\text{Num2}, \text{Countofdig} - 1, \text{lDecNum})$

переведення числа з десятикової в шістнадцяткову системи числення

виведення HexNum

Кінець

GetCount(a, b)

якщо $a > 0$

то

$a = a / 10$

$b++$

повернути GetCount(a,b)

все якщо

інакше

повернути b

Кінець

GetDEC(a, b, c)

якщо $b \geq 0$

то

якщо $a / \text{pow}(10, b) \geq 1$

то

$a = a - \text{pow}(10, b)$

$c = c + \text{pow}(2, b--)$

все якщо

інакше

$b--$

повернути GetDEC(a, b, c)

все якщо

повернути c

Кінець

Крок 4.

Початок

введення Num2, lCountofdig = 0, lDecNum = 0, lHexNum, Countofdig, DecNum,
HexNum

Countofdig = GetCount(sNum, sCountNum)

DecNum = GetDEC(Num2, Countofdig - 1, lDecNum)

HexNum = GetHEX(DecNum, lHexNum)

виведення HexNum

Кінець

GetCount(a, b)

якщо $a > 0$

то

$a = a / 10$

$b++$

повернути GetCount(a,b)

все якщо

інакше

повернути b

Кінець

GetDEC(a, b, c)

якщо $b \geq 0$

то

якщо $a / \text{pow}(10, b) \geq 1$

то

$a = a - \text{pow}(10, b)$

$c = c + \text{pow}(2, b--)$

все якщо

інакше

$b--$

повернути GetDEC (a, b, c)

все якщо

повернути c

Кінець

GetHEX(a, b)

введення a1, a3, a4

$a1 = a$

$a3 = 0$

якщо $a / 16 \geq 1$

то

$a = a / 16$

$a3 = a1 - (16 * a)$

якщо $a3 < 10$

то

$b = \text{to_string}(a3) + b$

все якщо

інакше якщо $a3 \geq 10$

то

якщо $a3 == 10$

то

$a4 = 'A'$

все якщо

інакше якщо $a3 == 11$

то

$a4 = 'B'$

все якщо

інакше якщо $a3 == 12$

то

$a4 = 'C'$

все якщо

інакше якщо $a3 == 13$

то

$a4 = 'D'$

все якщо

інакше якщо $a3 == 14$

то

$a4 = 'E'$

все якщо

інакше якщо $a3 == 15$

то

$a4 = 'F'$

все якщо

$b = a4 + b$

все якщо

повернути $\text{GetHEX}(a, b)$

все якщо

інакше якщо $a \geq 10$

то

якщо $a3 == 10$

то

$a4 = 'A'$

все якщо

інакше якщо $a3 == 11$

то

$a4 = 'B'$

все якщо

інакше якщо $a3 == 12$

то

$a4 = 'C'$

все якщо

інакше якщо $a3 == 13$

то

$a4 = 'D'$

все якщо

інакше якщо $a3 == 14$

то

$a4 = 'E'$

все якщо

інакше якщо $a3 == 15$

то

$a4 = 'F'$

все якщо

$b = a4 + b$

все якщо

інакше

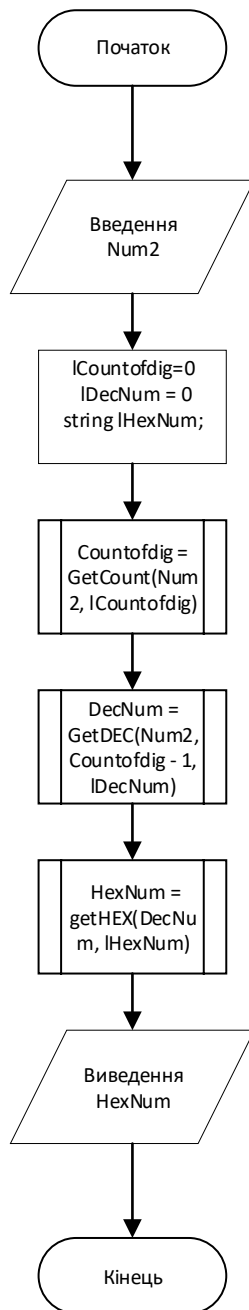
$b = \text{to_string}(a) + b$

повернути b

Кінець

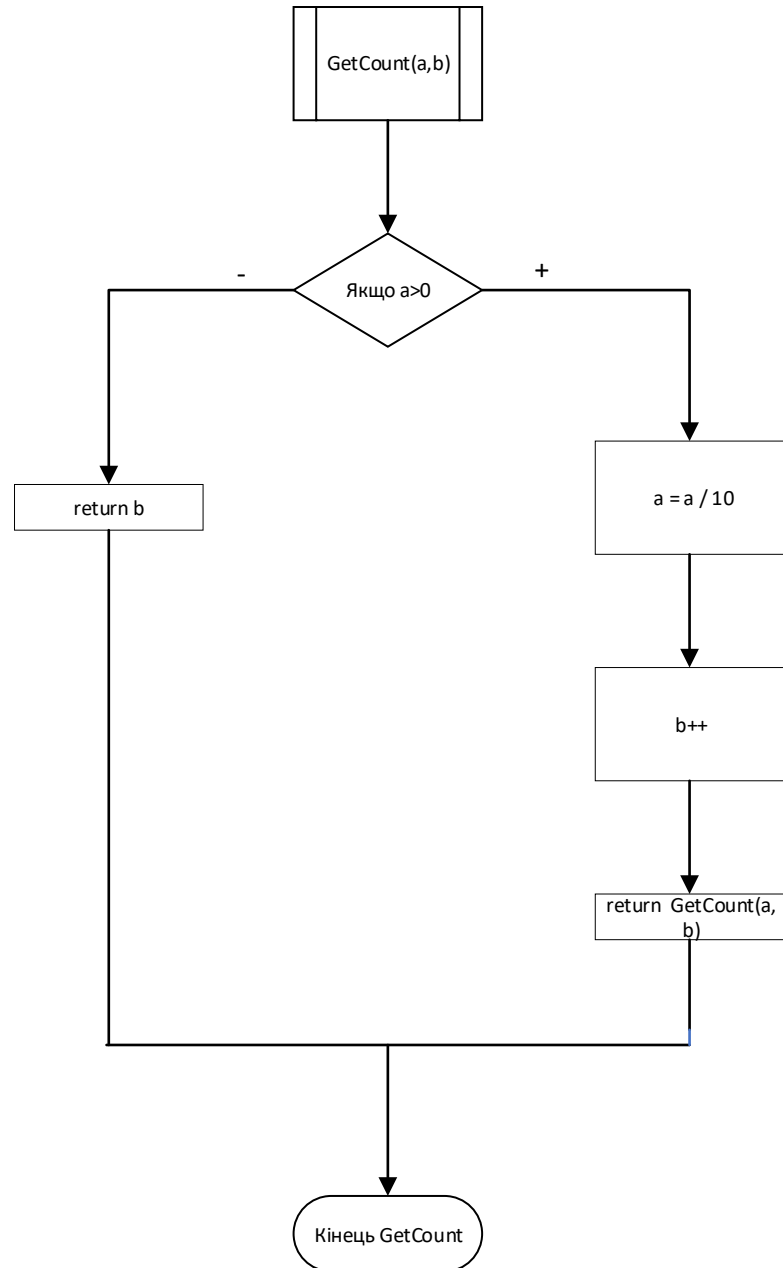
Блок-схема алгоритму:

Блок-схема основної програми:

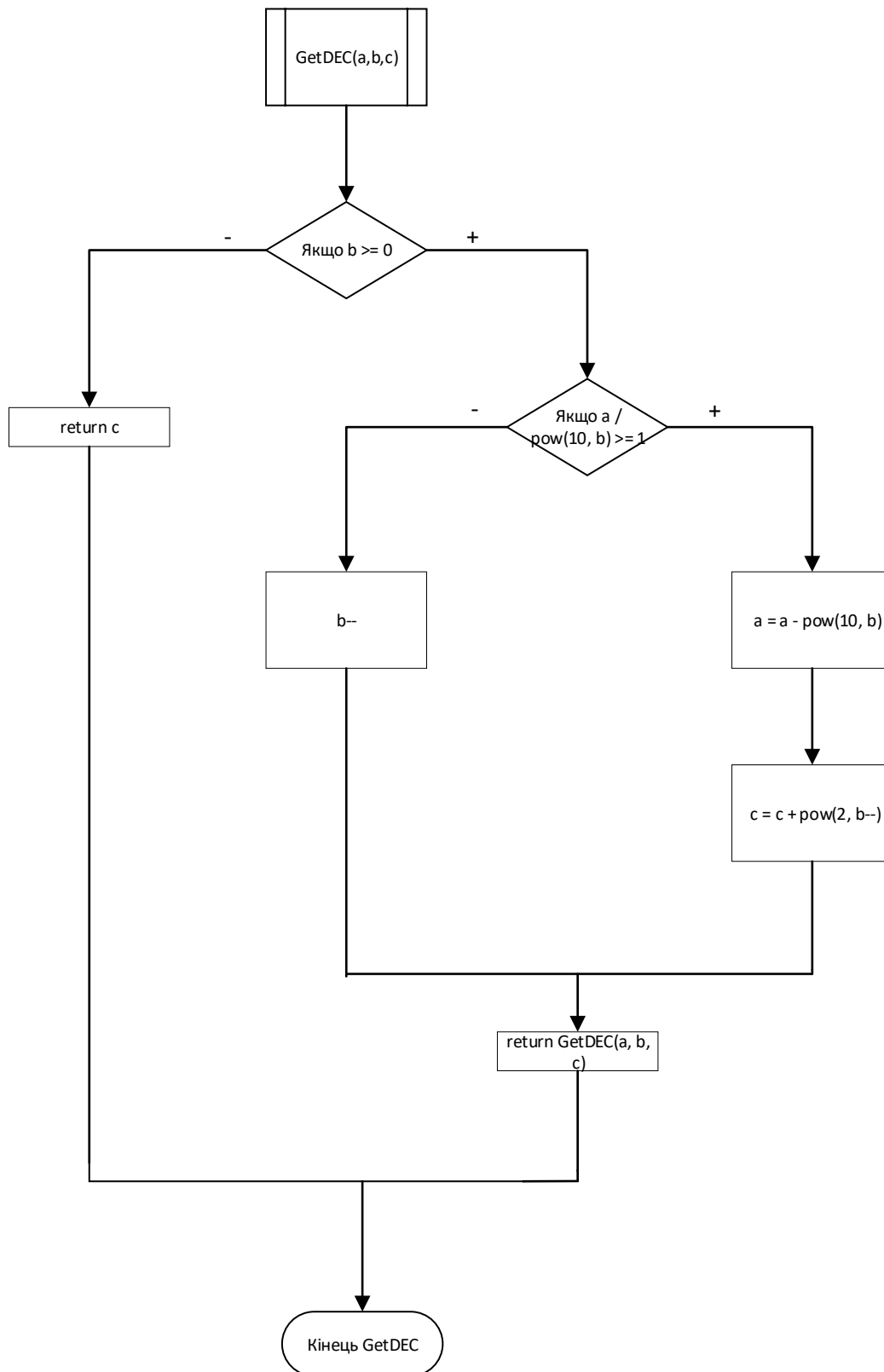


Блок-схеми функцій:

GetCount():



GetDEC():



GetHEX():

Код програми на мові C++:

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

int GetCount(int a, int b); //обрахування к-сті елементів числа
int GetDEC(int a, int b, int c); //знаходження десяткового числа
string GetHEX(int a, string b); //знаходження шістнадцяткового числа

int main() {
    setlocale(LC_ALL, "Russian");

    int Num2, lCountofdig = 0, lDecNum = 0;
    cout << "Введіть, будь ласка, число в бінарній системі исчислення: ";
    cin >> Num2;
    string lHexNum;
    int Countofdig = GetCount(Num2, lCountofdig);
    int DecNum = GetDEC(Num2, Countofdig - 1, lDecNum);
    string HexNum = GetHEX(DecNum, lHexNum);
    cout << endl << "Ваше число в шістнадцятковій системі исчислення: " << HexNum;
}

int GetCount(int a, int b) { //обрахування к-сті елементів числа
    if (a > 0) {
        a = a / 10;
        b++;
        return GetCount(a, b); //рекурсія
    }
    return b; //повернення к-сті елементів числа
}

int GetDEC(int a, int b, int c) { //знаходження десяткового числа
    if (b >= 0) {
        if (a / pow(10, b) >= 1) { //перевірка на подільність розряду
            a = a - pow(10, b);
            c = c + pow(2, b--); //десятькове число, за правилом поділу двійкового числа на
розряди
        }
        else b--;
        return GetDEC(a, b, c); //рекурсія
    }
    return c; //повернення десяткового числа
}

string GetHEX(int a, string b) { //знаходження шістнадцяткового числа
    int a1 = a; //ініціалізація дублікату а для зручності зміни числа
    int a3 = 0;
    char a4='0';
    if (a / 16 >= 1) { // перевірка десяткового числа, впливає, що а<16
        a = a / 16;
        a3 = a1 - (16 * a);
        if (a3 < 10) b = to_string(a3) + b; //приєднання числа в рядок
        else if (a3 >= 10) {
            if (a3 == 10) a4 = 'A';
            else if (a3 == 11) a4 = 'B';
            else if (a3 == 12) a4 = 'C';
            else if (a3 == 13) a4 = 'D';
        }
    }
}
```

```

        else if (a3 == 14) a4 = 'E';
        else if (a3 == 15) a4 = 'F';
        b = a4 + b; //приєднання букви, яка заміняє число в шіст. сист. числ., в рядок
    }
    return GetHEX(a, b); //рекурсія
}
else if (a >= 10) { //перевірка десяткового числа, впливає, що 10<=a<16
    if (a == 10) a4 = 'A';
    else if (a == 11) a4 = 'B';
    else if (a == 12) a4 = 'C';
    else if (a == 13) a4 = 'D';
    else if (a == 14) a4 = 'E';
    else if (a == 15) a4 = 'F';
    b = a4 + b; //приєднання букви, яка заміняє число в шіст. сист. числ., в рядок
}
else b = to_string(a) + b; //приєднання числа в рядок, якщо воно <10
return b; //повернення результуючого числа
}

```

Випробування алгоритму на мові C++:

```

Консоль отладки Microsoft Visual Studio
Введите, пожалуйста, число в бинарной системе исчисления: 11111
Ваше число в шестнадцатичной системе исчисления: 1F
C:\Users\Oleksii Savenko\source\repos\laba6asd\Debug\laba6asd.exe (процесс 13684) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```

Випробування алгоритму:

Блок	Дія
	Початок
1	введення Num2 = 11111, lCountofdig = 0, lDecNum = 0, lHexNum = "",

	Countofdig, DecNum, HexNum
2. Функція GetCount() для визначення к-сті елементів числа	b = 5 Countofdig = 5
3. Функція GetDEC() для перетворення двійкового числа в десяткове	c = 31 DecNum = 31
3. Функція GetHEX() для перетворення десяткового числа в шістнадцяткове	b = 1F HexNum = b = 1F
4	виведення: HexNum = 1F
	Кінець

Висновок:

Я дослідив особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм. Мною було розроблено алгоритм для переведення бінарного числа у шістнадцяткову систему. Були розроблені постановка задачі та матмодель, де було детально описана логіка алгоритма, зокрема використання функцій у ньому та механізму рекурсії для отримання та повернення до головної функції потрібних значень – кількості цифр числа з функції **GetCount()**, числа у десятковій системі числення з **GetDEC()** та числа у шістнадцятковій системі числення з **GetHEX()**. Було написано псевдокод та блок-схеми алгоритму, застосовано алгоритм для написання програмного коду на мові C++, після чого було проведено випробування алгоритму на бінарному числі **Num2 = 11111**, було отримано значення знаків **Countofdig = 5**, перетворено його на число в десятковій системі числення **DecNum = 31** і у кінці отримано значення у

16-ковій системі числення **HexNum = 1F** та виведено результат **1F**. Це значення відповідає дійсності, отже мій алгоритм та програмний код є вірними і їх можна використовувати для вирішення завдань даного типу.