**Міністерство Освіти і Науки України**

**Київський Національний Університет імені Тараса Шевченка**

**Факультет Інформаційних Технологій**

Звіт з лабораторної роботи № 6

з дисципліни **Основи програмування**

Виконав студент групи ІР-12

Федоренко Ярослав Юрійович

Викладач: ас. Пороховніченко І.А.

**Київ – 2023**

**Лабораторна робота №6**

**Варіант № 18**

**Тема:** Рекурсія та рекурсивні функції.

**Мета:** Опанувати принципи рекурсивних обчислень. Навчитися вирішувати задачі з використанням рекурсивних функцій. Опанувати конструктивні особливості та правила застосування рекурсивних функцій на прикладі мови програмування високого рівня С++.

**Завдання № 1**





**­Код**

**main.cpp**

#include <iostream>

#include "iteration.h"

#include "recursion.h"

using namespace std;

int main()

{

char q = 'y';

do {

int n = 0;

cout << "Введіть n: ";

cin >> n;

if (n < 1) {

cout << endl << "n має бути більше 1!" << endl << endl;

continue;

}

cout << endl;

cout << "Число Фібоначчі (Ітеративний метод): " << iFibonacci(n) << endl;

cout << "Число Фібоначчі (Рекурсивний метод): " << rFibonacci(n) << endl;

cout << endl;

cout << "Число Фібоначчі F(n + 1) (Ітеративний метод): " << iFibonacci(n + 1) << endl;

cout << "Число Фібоначчі F(n + 1) (Рекурсивний метод): " << iFibonacci(n + 1) << endl;

cout << endl;

int Fr = rSum(n);

int Fi = iSum(n);

cout << endl;

cout << "Значення суми (Ітераційний метод): " << Fi << endl;

cout << "Значення суми (Рекурсивний метод): " << Fr << endl;

cout << endl << endl << "Бажаєте повторити? (y/n): ";

cin >> q;

if (q != 'y')

cout << endl << "До побачення!";

else

cout << endl;

} while (q == 'y');

return 0;

}

**iteration.cpp**

#include "iteration.h"

double iFactorial(int number)

{

double result = 1;

for (int i = 1; i <= number; i++) {

result \*= i;

}

return result;

}

double iCKN(int k, int n)

{

if (k > n && n >= 0) {

return 0;

} else if ((k == n && n >= 0) || (k == 0 && n > 0)) {

return 1;

} else {

double C1 = iFactorial(n - 1) / (iFactorial(k - 1) \* iFactorial((n - 1) - (k - 1)));

double C2 = iFactorial(n - 1) / (iFactorial(k) \* iFactorial((n - 1) - k));

return C1 + C2;

}

}

int iFibonacci(int n)

{

int a = 0, b = 1;

for(int i = 2; i <= n; i++)

{

int c = a + b;

a = b;

b = c;

}

return b;

}

int iSum(int n)

{

double result = 0;

int k = 0;

while (k <= n/2) {

result += iCKN(k, n - k);

k++;

}

cout << "Кількість ітерацій: " << k << endl;

return result;

}

**recursion.cpp**

#include "recursion.h"

double rFactorial(int number)

{

if (number == 0 || number == 1) {

return 1;

} else {

return number \* rFactorial(number - 1);

}

}

double rCKN(int k, int n)

{

if (k > n && n >= 0) {

return 0;

} else if ((k == n && n >= 0) || (k == 0 && n > 0)) {

return 1;

} else {

double C1 = rFactorial(n - 1) / (rFactorial(k - 1) \* rFactorial((n - 1) - (k - 1)));

double C2 = rFactorial(n - 1) / (rFactorial(k) \* rFactorial((n - 1) - k));

return C1 + C2;

}

}

int rFibonacci(int n, int a, int b)

{

if (n == 0) {

return a;

}

if (n == 1) {

return b;

}

return rFibonacci(n - 1, b, a + b);

}

int rSum(int n, int k, double sum)

{

if (k > n/2) {

cout << "Глибина рекурсії: " << k << endl;

return sum;

} else {

return rSum(n, k + 1, sum + rCKN(k, n - k));

}

}

**Тестування**

При введеному значенні n 5 отримуємо наступні числа:



Відповіді збігаються з обчисленнями в калькуляторі. Програма працює коректно.

**Завдання № 2**



**Код**

**main.cpp**

#include <iostream>

#include "coutNumbers.h"

using namespace std;

int main()

{

char q = 'y';

do {

int n = 0;

cout << "Введіть натуральне число n: ";

cin >> n;

if (n < 1) {

cout << endl << "n має бути більше 1!" << endl << endl;

continue;

}

cout << endl << "Числа від 1 до n:";

coutNumbers(n);

cout << endl << endl << "Бажаєте повторити? (y/n): ";

cin >> q;

if (q != 'y')

cout << endl << "До побачення!";

else

cout << endl;

} while (q == 'y');

return 0;

}

**coutNumbers.cpp**

#include "coutNumbers.h"

void coutNumbers(int n, int current)

{

cout << " " << current;

if (current == n) {

return;

}

coutNumbers(n, current + 1);

}

**Тестування**

Вводимо число 5, програма виводить "1 2 3 4 5". Все працює коректно.

**Висновок**

В цій лабораторній роботі я опанував приципи рекурсивних обчислень. Я не тільки познайомився з прикладами використання подібних функцій, а й порівняв ефективність рекурсивного та ітеративного методу обчислень.