Дисциплина «Алгоритмы решения прикладных задач»
Рабочая тетрадь 4.
Рекурсия

Теоретический материал

Рекурсия — это такой способ организации вспомогательного алгоритма (подпрограммы), при котором эта подпрограмма (процедура или функция) в ходе выполнения ее операторов обращается сама к себе. То есть в теле функции она вызывает саму себя.

Практически любую рекурсивную функцию можно переписать нерекурсивным образом с применением циклов и условных операторов.

Например, вычисление факториала числа *N*, т. е. вычисление произведения всех чисел от 1 до *N* (в математике обозначается *N*!) нерекурсивным способом можно на C++ записать в виде следующего цикла;

```
int N=10; //будем вычислять факториал числа 10
int fact = 1; //переменная для «накопления» значения факториала
for(int i=1; i<=N; i++)
{
     fact*=i;
}
//По окончании цикла в переменной fact будет факториал числа N
```

Рекурсивно посчитать факториал можно, написав функцию, которая при выполнении вызывает себя. Такое возможно, поскольку N!=N*(N-1)!

```
int factorial(int i)
{
  if (i==0) return 1;
  else return i*factorial(i-1);
}
```

Пример 1 демонстрирует вычисление факториала рекурсивным и нерекурсивным образом.

Пример 1

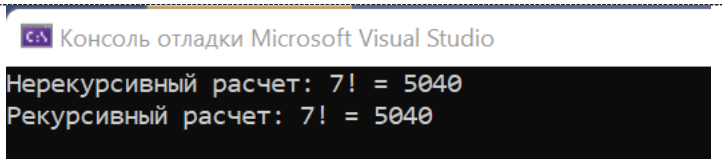
Задача:

Написать на языке C++ программу, в которой факториал числа N считается с помощью цикла и с помощью рекурсивной процедуры.

Решение:

```cpp
#include <iostream>

using namespace std;

//Рекурсивная функция для вычисления факториала
int factorial(int i)
{
    if (i == 0) return 1;
    else return i * factorial(i - 1);
}


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

    int N = 7; //будем вычислять факториал числа 7

    //Вычисление с помощью цикла
    int fact = 1; //переменная для «накопления» значения факториала
    for (int i = 1; i <= N; i++)
    {
        fact *= i;
    }    //По окончании цикла в переменной fact будет факториал числа N

    std::cout <<"Нерекурсивный расчет: " << N << "! = " << fact << endl;

    //Вызов рекурсивной функции
    std::cout << "Рекурсивный расчет: " << N << "! = " << factorial(N) << endl;

}
```

Ответ:



Задание 1

Задача:

Написать программу с рекурсивной функцией нахождения НОД двух чисел по алгоритму Евклида (*алгоритм объяснен в рабочей тетради 2, за основу можно взять нерекурсивный пример 1 из рабочей тетради 2*). Рекурсивно НОД двух чисел определяется следующим образом:

$$НОД(a,b)=\begin{cases} НОД(a-b,b), & если \quad a>b \\ НОД(a,b-a), & если \quad b>a \end{cases}$$

Выход из рекурсии происходит, когда a=b, это и есть НОД

Решение:

Ответ:

Задание 2

Задача:

Написать рекурсивную подпрограмму вычисления чисел Фибоначчи. $X_n=X_{n-1}+X_{n-2}$; $X_0=1$; $X_1=1$ (см. подсказки в слайдах с лекции)

Решение:

Ответ:

Задание 3

Задача:

Переписать программу из рабочей тетради 1 для нахождения корня уравнения методом половинного деления, реализовав метод через рекурсивную процедуру (вариант тот же, что и в рабочей тетради 1, см. подсказки в слайдах с лекции)

Решение:

Ответ:

Задание 4*

Задача:

Написать программу с рекурсивной функцией для нахождения суммы цифр числа.

Решение:

Ответ:

Задание 5*

Задача:

Написать программу с рекурсивной функцией для нахождения значения функции Дейкстры

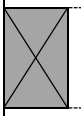
$$\begin{cases} F(1)=1 \\ F(2N)=F(N) \\ F(2N+1)=F(N)+F(N+1) \end{cases}$$

Решение:

Ответ:

Задание 6**

Задача:



Написать программу с рекурсивной функцией для нахождения решения головоломки «Ханойская башня» (без визуализации)

Решение:

Ответ: