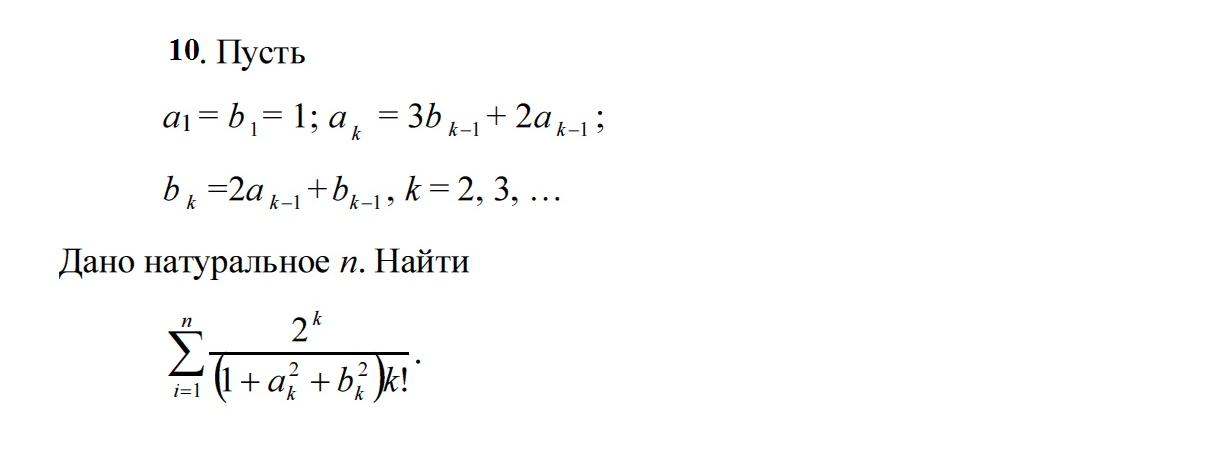
**С чего начать выполнение задания по теме «Рекурсивные функции»?**

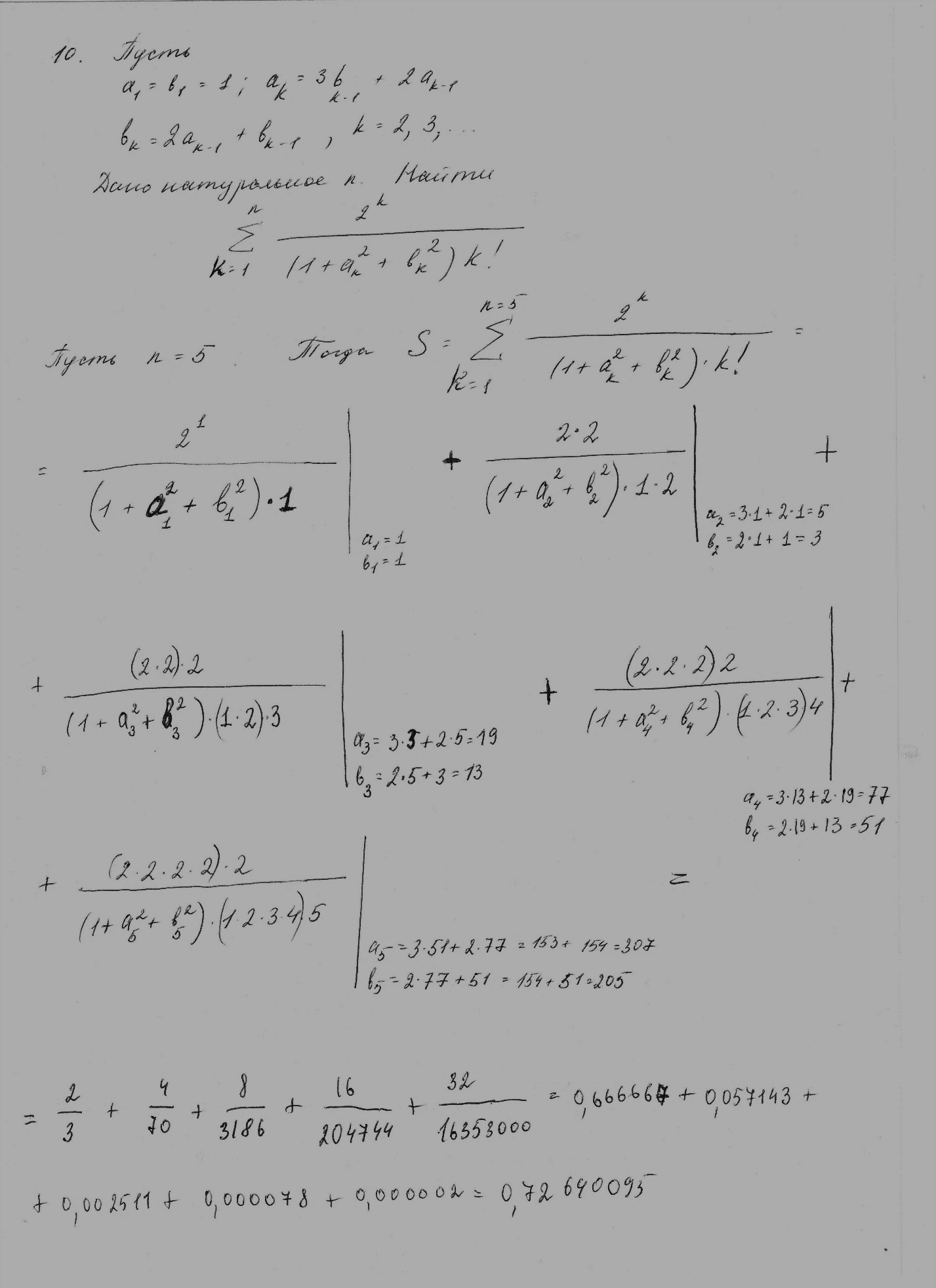
Найти задание для своего варианта.

Например



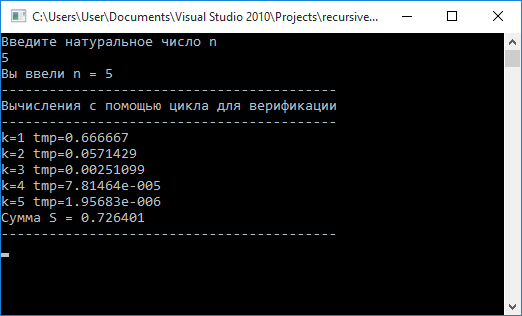
Затем найти значения выражения (-ий) для конктреных данных.

Например, для варианта 10, найти значения суммы при n=5.



Затем пишем программу для определения значений выражения(-ий) с помощью обычных циклов.

Сверяем с вычислениями в листке, они должны совпадать.



Текст программы в Microsoft Visual Studio 2010.

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <locale.h>

using namespace std;

//В функции реализованы вычисления с помощью цикла для верификации

double sum\_1(int n)

{

double ak,bk,ak\_1,bk\_1, sum , fact\_k, tmp, pow\_2\_k;

int k;

if (n>0)

{

ak=1;

ak\_1=1;

bk=1;

bk\_1=1;

sum=0;

fact\_k=1;

pow\_2\_k=2;

k=1;

sum+=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

cout <<"k=1 tmp=" << sum << endl;

for(k=2; k<=n; k++)

{

pow\_2\_k\*=2;

ak\_1=ak;

bk\_1=bk;

tmp=2\*ak\_1;

ak=tmp+3\*bk\_1;

bk=tmp+bk\_1;

fact\_k\*=k;

tmp=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

cout <<"k="<<k<<" tmp=" << tmp << endl;

sum+=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

}

return sum;

}

else return -1;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "Russian");

int n;

cout << "Введите натуральное число n" << endl;

cin >> n;

//эхо-контроль

cout << "Вы ввели n = "<< n << endl;

//Вычисления с помощью цикла для верификации

cout << "------------------------------------------" << endl;

cout << "Вычисления с помощью цикла для верификации" << endl;

cout << "------------------------------------------" << endl;

cout << "Сумма S = "<< sum\_1(n)<< endl;

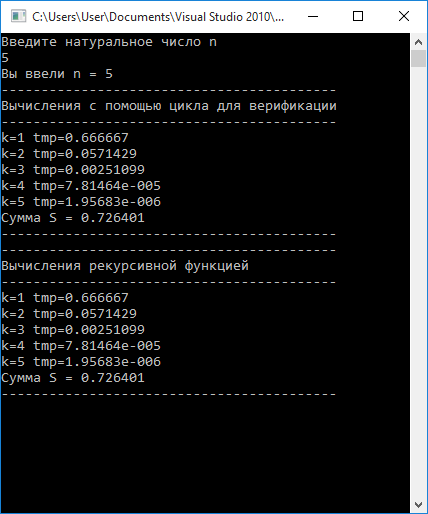
cout << "------------------------------------------" << endl;

\_getch();

return 0;

}

Затем дублируем эту функцию и изменяем ее, чтоб она была рекурсивной. Сверяем значения , полученные в результаты работы обеих функций.



// recursive\_functions\_option\_10.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <locale.h>

using namespace std;

//В функции реализованы вычисления с помощью цикла для верификации

double sum\_1(int n)

{

double ak,bk,ak\_1,bk\_1, sum , fact\_k, tmp, pow\_2\_k;

int k;

if (n>0)

{

ak=1;

ak\_1=1;

bk=1;

bk\_1=1;

sum=0;

fact\_k=1;

pow\_2\_k=2;

k=1;

sum+=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

cout <<"k=1 tmp=" << sum << endl;

for(k=2; k<=n; k++)

{

pow\_2\_k\*=2;

ak\_1=ak;

bk\_1=bk;

tmp=2\*ak\_1;

ak=tmp+3\*bk\_1;

bk=tmp+bk\_1;

fact\_k\*=k;

tmp=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

cout <<"k="<<k<<" tmp=" << tmp << endl;

sum+=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

}

return sum;

}

else return -1;

}

//Рекурсивная функция

double sum\_2(int n)

{

static double ak=1, bk=1, ak\_1=1, bk\_1=1, sum=0, fact\_k=1, pow\_2\_k=2;

static int k=1;

double tmp;

if (n==1)

{

tmp=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

cout <<"k="<<k<<" tmp=" << tmp << endl;

sum+=tmp;

return sum;

}

else if (n>1)

{

tmp=pow\_2\_k/((1+ak\*ak+bk\*bk)\*fact\_k);

cout <<"k="<<k<<" tmp=" << tmp << endl;

sum+=tmp;

k++;

pow\_2\_k\*=2;

ak\_1=ak;

bk\_1=bk;

tmp=2\*ak\_1;

ak=tmp+3\*bk\_1;

bk=tmp+bk\_1;

fact\_k\*=k;

return sum\_2(n-1);

}

else return -1;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "Russian");

int n;

cout << "Введите натуральное число n" << endl;

cin >> n;

//эхо-контроль

cout << "Вы ввели n = "<< n << endl;

//Вычисления с помощью цикла для верификации

cout << "------------------------------------------" << endl;

cout << "Вычисления с помощью цикла для верификации" << endl;

cout << "------------------------------------------" << endl;

cout << "Сумма S = "<< sum\_1(n)<< endl;

cout << "------------------------------------------" << endl;

cout << "------------------------------------------" << endl;

cout << "Вычисления рекурсивной функцией" << endl;

cout << "------------------------------------------" << endl;

cout << "Сумма S = "<< sum\_2(n)<< endl;

cout << "------------------------------------------" << endl;

\_getch();

return 0;

}