Содержание

[Задание 1. Программирование линейных алгоритмов 3](#_Toc449694652)

[Задание 2. Программирование ветвящихся алгоритмов. 7](#_Toc449694653)

[Задание 3. Программирование циклических алгоритмов 16](#_Toc449694654)

[Задание 4. Линейные массивы. Сортировка массивов. 20](#_Toc449694655)

[Список используемой литературы 25](#_Toc449694656)

# Задание 1. Программирование линейных алгоритмов

Дано а. Не используя никаких функций и никаких операций, кроме умножения, получить а8 за три операции; а10 и а16 за четыре операции.

1. Работа программы: вводится исходное число а с клавиатуры.

2. Назначение переменных

2.1 Входная информация

а – вещественное число,

2.2 Выходная информация

а - вещественное число,

b – вещественное число

2.3 Промежуточные переменные

b – вещественные числа

3. Формализация задачи

3.1 вводится любое вещественное число а с клавиатуры, и рассчитывается а8 за три операции:

b = а\*а;

a = b\*b;

c = a\*a

3.2 вводится любое вещественное число а с клавиатуры, и рассчитывается а10 за четыре операции:

b = а\*а;

a = b\*b;

а = a\*a;

а = а\*b;

3.3 вводится любое вещественное число а с клавиатуры, и рассчитывается а16 за четыре операции:

b = а\*а;

a = b\*b;

а = a\*a;

а = а\*а;

4. Блок-схема представлена на рис.1

Начало

Ввод а

Вычисление а8 за три операции

Вывод а8

Вычисление а10 за четыре операции

Вывод а10

Вычисление а16 за четыре операции

Вывод а16

Конец

Рисунок 1 – Блок схема программы

7. Программирование интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| Название  компоненты | Содержание | Описание | Реакция на действие  пользователя |
| Buttom1 | Кнопка “рассчитать” | Произвести расчет выражения | Есть |
| Buttom1 | Кнопка”выход” | Выйти из приложения | Есть |
| Edit1 | Поле для ввода А | Ввод А | Отображение вводимого значения |
| Memo | Поле для вывода | Вывод текстовой информации и значения | Возможность прокрутки по вертикали и горизонтали |

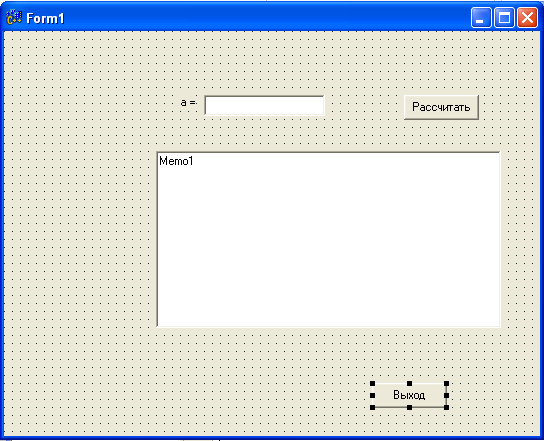


Рисунок 3 - Вид экрана с результатом проектирования интерфейса

Текст программы

//----------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//----------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

double a,b,c,aa;

Memo1->Clear();

aa = StrToFloat(Edit1->Text);

a=aa;

b = a \*a;

a = b \*b;

a = a \*a;

Memo1->Lines->Add("Задание№ 1 Вариант 35: ");

Memo1->Lines->Add("При a = "+FloatToStrF(aa,ffFixed,7,3)

+"; a в степени 8 за три операции = "+FloatToStrF(a,ffFixed,7,3));

b = a \* b;

Memo1->Lines->Add("При a = "+FloatToStrF(aa,ffFixed,7,3)

+"; a в степени 10 за четыре операции = "+FloatToStrF(b,ffFixed,7,3));

a = a \* a;

Memo1->Lines->Add("При a = "+FloatToStrF(aa,ffFixed,7,3)

+"; a в степени 16 за четыре операции = "+FloatToStrF(a,ffFixed,7,3));

}

//---------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------------

Контрольный пример

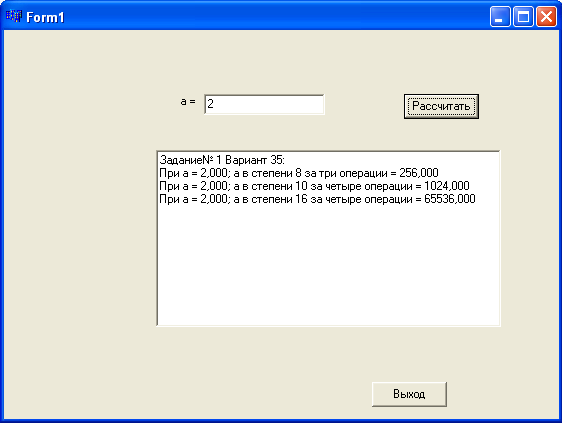


Рисунок 4 - Вид экрана с результатом решения задачи

# Задание 2. Программирование ветвящихся алгоритмов.

Пусть элементами равностороннего треугольника являются: 1) сторона а; 2) площадь S; 3) высота h; 4) радиус вписанной окружности r; 5) радиус описанной окружности R.

Составить программу, которая по заданному номеру и значению соответствующего элемента вычисляла бы значения всех остальных элементов треугольника.

1. Работа программы: выбрать номер решения и ввести соответствующий элемент с клавиатуры.

2. Назначение переменных

2.1 Входная информация

n – целое число, номер задания,

а – вещественное число, сторона треугольника;

S – вещественное число, площадь треугольника;

h – вещественное число, высота треугольника;

r – вещественное число, радиус вписанной окружности;

R – вещественное число, радиус описанной окружности;

2.2 Выходная информация

а – вещественное число, сторона треугольника;

S – вещественное число, площадь треугольника;

h – вещественное число, высота треугольника;

r – вещественное число, радиус вписанной окружности;

R – вещественное число, радиус описанной окружности;

3. Формализация задачи

r:=a/2; R:=a/; s:= \*а2/4; h:=2\*s/a; при выборе пункта 1;

a:= ; h:=2\*s/a; r:=a/2; R:=a/; при выборе пункта 2;

a:=2\*h/; s:=a\*h/2; r:=a/2; R:=a/; при выборе пункта 3;

a:=2\*r; R:=a/; s:= \*а2/4; h:=2\*s/a; при выборе пункта 4;

a:= \*R; s:=(a\*a\*a)/(4\*R); h:=2\*s/a; r:=a/2; при выборе пункта 5;

при вводе любого другого числа – сообщение о неверном вводе

4. Блок-схема представлена на рис.3

7. Программирование интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| Название  компоненты | Содержание | Описание | Реакция на действие  пользователя |
| Buttom1 | Кнопка “рассчитать” | Произвести расчет выражения | Есть |
| Buttom1 | Кнопка”выход” | Выйти из приложения | Есть |
| Edit1 | Поле для ввода n | Ввод n | Отображение вводимого значения |
| Edit2 | Поле для ввода значения параметра | Ввод значения параметра | Отображение вводимого значения |
| Memo | Поле для вывода | Вывод текстовой информации и значения | Возможность прокрутки по вертикали и горизонтали |

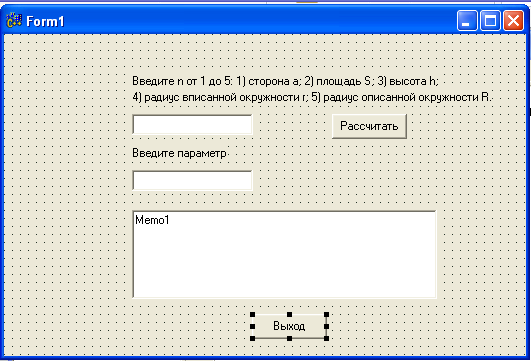


Рисунок 8 - Вид экрана с результатом проектирования интерфейса

Текст программы

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include <math.h>

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

int n;

double a, S, h,r, R;

Memo1->Clear();

n = StrToInt(Edit1->Text);

Memo1->Lines->Add("Задание№ 2 Вариант 35: ");

switch (n)

{

case 1: //если n =1

{a = StrToFloat(Edit2->Text);

S=pow(3,0.5)\*pow(a,2)/4;

h = 2\*S/a;

r = a/2;

R = a/pow(3, 0.5);

Memo1->Lines->Add("При a = "+FloatToStrF(a,ffFixed,7,3)

+"; площадь S = "+FloatToStrF(S,ffFixed,7,3)+";");

Memo1->Lines->Add("высота h = "+FloatToStrF(h,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус вписанной окружности r = "+FloatToStrF(r,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус описанной окружности R = "+FloatToStrF(R,ffFixed,7,3)+";");

}

case 2: //если n =2

{S = StrToFloat(Edit2->Text);

a=pow((4\*S/pow(3,0.5)),0.5);

h = 2\*S/a;

r = a/2;

R = a/pow(3, 0.5);

Memo1->Lines->Add("При S = "+FloatToStrF(S,ffFixed,7,3)

+"; сторона a = "+FloatToStrF(a,ffFixed,7,3)+";");

Memo1->Lines->Add("высота h = "+FloatToStrF(h,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус вписанной окружности r = "+FloatToStrF(r,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус описанной окружности R = "+FloatToStrF(R,ffFixed,7,3)+";");

}

case 3: //если n =3

{h = StrToFloat(Edit2->Text);

a=2\*h/pow(3,0.5);

S = a\*h/a;

r = a/2;

R = a/pow(3, 0.5);

Memo1->Lines->Add("При h = "+FloatToStrF(h,ffFixed,7,3)

+"; сторона a = "+FloatToStrF(a,ffFixed,7,3)+";");

Memo1->Lines->Add("площадь S = "+FloatToStrF(S,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус вписанной окружности r = "+FloatToStrF(r,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус описанной окружности R = "+FloatToStrF(R,ffFixed,7,3)+";");

}

case 4: //если n =4

{r = StrToFloat(Edit2->Text);

a = 2\*r;

a=2\*h/pow(3,0.5);

R = a/pow(3, 0.5);

S = pow(3,0.5)\*(a,2)/4;

h = 2\*S/a;

Memo1->Lines->Add("При r = "+FloatToStrF(r,ffFixed,7,3)

+"; сторона a = "+FloatToStrF(a,ffFixed,7,3)+";");

Memo1->Lines->Add("площадь S = "+FloatToStrF(S,ffFixed,7,3)+";");

Memo1->Lines->Add("высота h = "+FloatToStrF(h,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус описанной окружности R = "+FloatToStrF(R,ffFixed,7,3)+";");

}

case 5: //если n =5

{R = StrToFloat(Edit2->Text);

a=pow(3,0.5)\*R;

S = (a\*a\*a)/(4\*R);

h = 2\*S/a;

r = a/2;

R = a/pow(3, 0.5);

Memo1->Lines->Add("При R = "+FloatToStrF(R,ffFixed,7,3)

+"; сторона a = "+FloatToStrF(a,ffFixed,7,3)+";");

Memo1->Lines->Add("площадь S = "+FloatToStrF(S,ffFixed,7,3)+";");

Memo1->Lines->Add("высота h = "+FloatToStrF(h,ffFixed,7,3)+";");

Memo1->Lines->Add("радиус вписанной окружности r = "+FloatToStrF(r,ffFixed,7,3)+";");

}

default: // если count равно любому другому значению

Memo1->Lines->Add("Такого пункта нет ");

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------------------------

Контрольные примеры

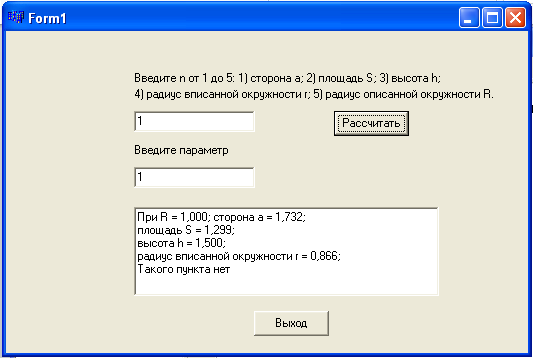


Рисунок 9 - Вид экрана с результатом решения задачи

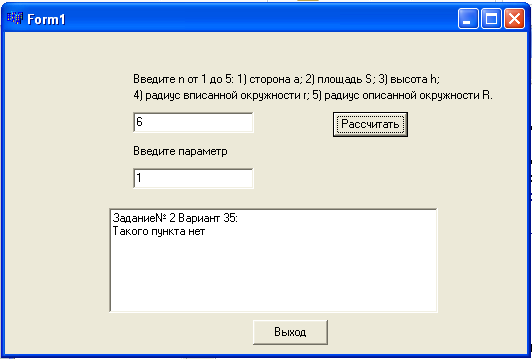


Рисунок 10 - Вид экрана с результатом решения задачи

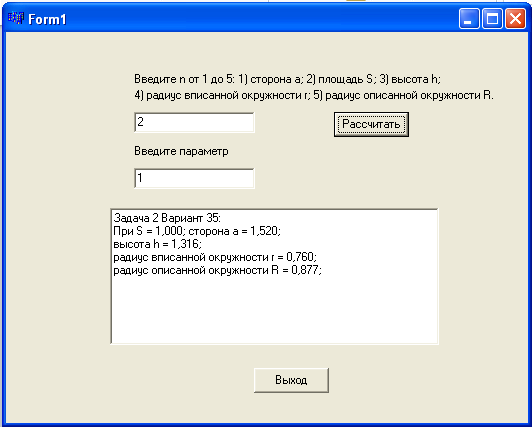


Рисунок 11 - Вид экрана с результатом решения задачи

Рисунок 3 – Блок схема программы

Начало

Конец

n = 1..5

нет

Ввод стороны а

Расчет S, h, r, R

Вывод S, h, r, R на экран

Ввод номера n

Да

Ввод радиуса R

Расчет a, S, h, r

Вывод a, S, h, r на экран

Ввод площади S

Расчет a, h, r, R

Вывод a, h, r, R на экран

Ввод высоты h

Расчет a, S, r, R

Вывод a, S, r, R на экран

Ввод радиуса r

Расчет a, S, h, R

Вывод a, S, h, R на экран

Такого пункта нет

1

2

3

4

5

# Задание 3. Программирование циклических алгоритмов

Найти наименьший номер последовательности, для которого выполняется условие |an-an-1|<ε. Вывести на экран этот номер и все элементы ai, где i=1, 2, 3,…, n, если ε<10-3.

****

1. Работа программы: находиться наименьший номер последовательности, для которого выполняется заданное условие, выводиться этот номер на экран, а также все элементы.

2. Назначение переменных

2.1 Входная информация

Все исходные данные заданы в тексте программы

2.2 Выходная информация

i – целое число, наименьший номер последовательности

an – вещественные числа - все элементы, удовлетворяющие заданному условию

2.3 Промежуточные переменные

е – вещественное число, константа, используемая в условии;

а – вещественное число, равное предыдущему значению искомого элемента,

3. Формализация задачи

Выводим на экран все значения функции ****, пока не достигнем условие |an-an-1|<ε.

4. Блок-схема представлена на рис.7

Начало

Присвоение исходных данных e = 0.001; an = 0.5; i = 1;

Вывод первого элемента последовательности

|an-an-1|<е

да

Нахождение следующего элемента, номера

элемента

Вывод каждого последующего элемента

нет

Вывод номера элемента

Конец

Рисунок 12 – Блок схема программы

7. Программирование интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| Название  компоненты | Содержание | Описание | Реакция на действие  пользователя |
| Buttom1 | Кнопка “рассчитать” | Произвести расчет выражения | Есть |
| Buttom1 | Кнопка”выход” | Выйти из приложения | Есть |
| Edit1 | Поле для ввода n | Ввод n | Отображение вводимого значения |
| Memo | Поле для вывода | Вывод текстовой информации и значения | Возможность прокрутки по вертикали и горизонтали |

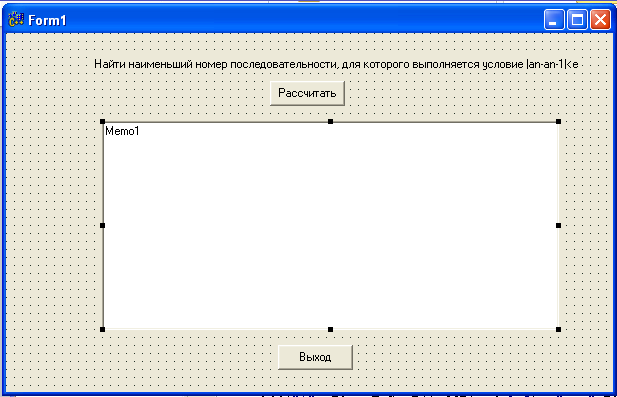


Рисунок 14 - Вид экрана с результатом проектирования интерфейса

Текст программы

//----------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

# include <math.h>

#include "Unit1.h"

//----------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//----------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//----------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

double a, an, e;

int i;

Memo1->Clear();

e = 0.001;

an = 0.5;

i = 1;

Memo1->Lines->Add("Задание№ 3 Вариант 35: ");

Memo1->Lines->Add("i" + FloatToStrF(i,ffFixed,2,0)+ " = "+FloatToStrF(an,ffFixed,7,3));

while (fabs(an-a)>e)

{

a=an;

an = 1./2.\*(sin(a))/(cos(a));

i++;

Memo1->Lines->Add("i"+ FloatToStrF(i,ffFixed,2,0)+ " = "+FloatToStrF(an,ffFixed,7,4));

}

Memo1->Lines->Add("Номер элемента = "+FloatToStrF(i,ffFixed,7,0));

}

//----------------------------------------------------------------

Контрольный пример

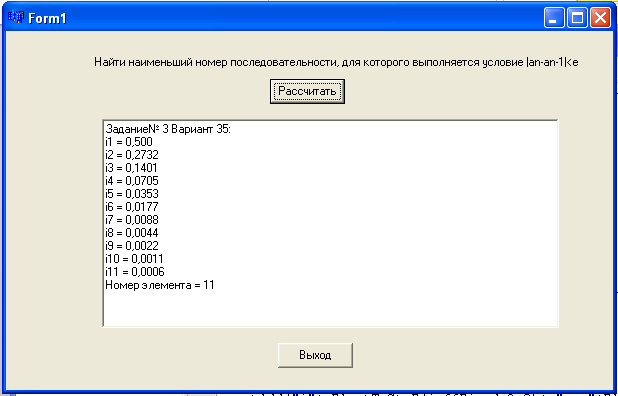


Рисунок 15 - Вид экрана с результатом решения задачи

# Задание 4. Линейные массивы. Сортировка массивов.

Сечение крыши имеет форму полукруга с радиусом R м. Сформировать таблицу, содержащую длины вертикаль опор, устанавливаемых через каждые R/5 м. R = 5-10м с шагом 0.1м.

1. Работа программы: Для каждого значения радиуса в интервале 5 – 10 с шагом 0,1 м. формируется строка, содержащая длины вертикальных опор, установленных через каждые R/5 м.

2. Назначение переменных

2.1. Входная информация

Все исходные данные заданы в тексте программы

2.2 Выходная информация

r – вещественное число, радиус сечения крыши;

h[i] – массив вещественных чисел, длин опор;

3. Формализация задачи

Во внешнем цикле формируется следующие радиус, внутренний цикл – формирует линейный массив длин опор.

4. Блок-схема представлена на рис.9

Начало

Формирование исходных данных r=5; j=0;

r<=10

да

Вывод радиуса

i=0; dx=r/5; x=0;

(x<=(2\*r-dx))

да

x=x+dx; i++;

h[i] = pow((x\*(2\*r-x)), 0.5);

Вывод длин опор

нет

r=r+0.1;

j++;

нет

Начало

Рисунок 16 – Блок схема программы

7. Программирование интерфейса

|  |  |  |  |
| --- | --- | --- | --- |
| Название  компоненты | Содержание | Описание | Реакция на действие  пользователя |
| Buttom1 | Кнопка “рассчитать” | Произвести расчет выражения | Есть |
| Buttom1 | Кнопка”выход” | Выйти из приложения | Есть |
| Memo | Поле для вывода | Вывод текстовой информации и значения | Возможность прокрутки по вертикали и горизонтали |

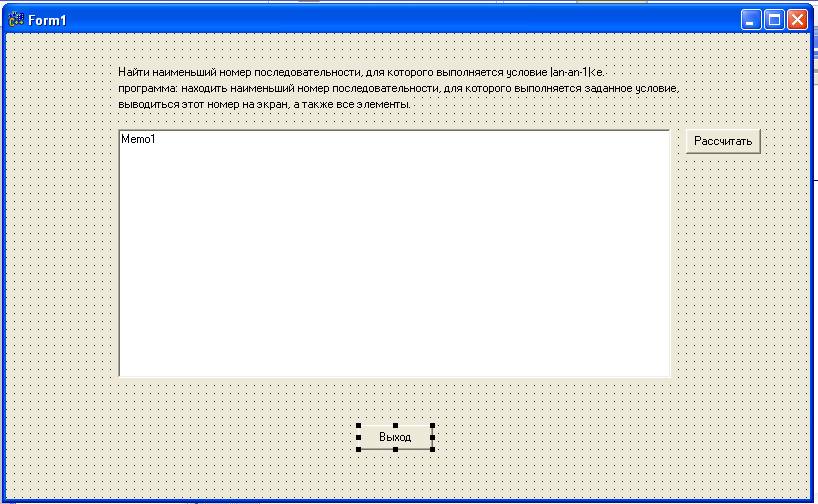


Рисунок 18 - Вид экрана с результатом проектирования интерфейса

Текст программы

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

# include <math.h>

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

double r,x,dx,h[9];

int i,j;

r=5;

j=0;

Memo1->Lines->Add("Задание№ 4 Вариант 35: ");

while (r<10)

{

Memo1->Lines->Add("Радиус опор = "+FloatToStrF(r,ffFixed,7,3));

i=0;

dx=r/5;

x=0;

while (x<=(2\*r-dx))

{

x=x+dx;

i++;

h[i] = pow((x\*(2\*r-x)), 0.5);

Memo1->Text = Memo1->Text + FloatToStr(h[i]) + ' ';

//Memo1->Lines->Add(FloatToStrF(h[i],ffFixed,7,3)+" ");

}

r=r+0.1;

j++;

}

}

//---------------------------------------------------------------------------

Контрольные примеры

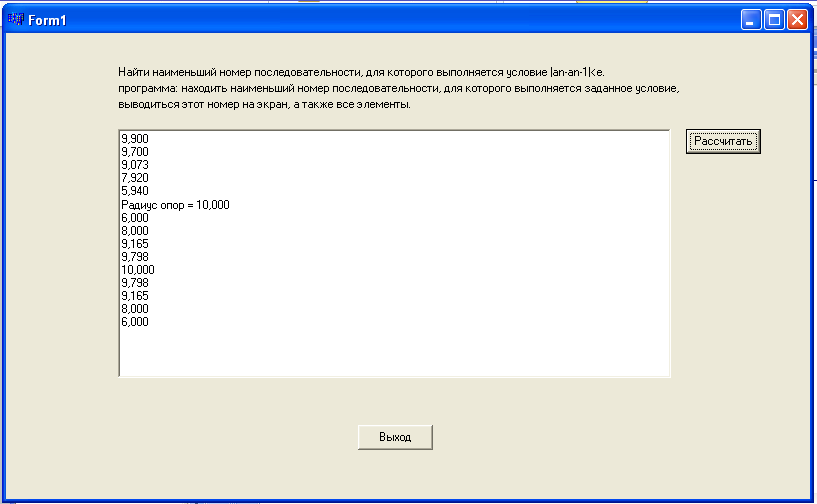


Рисунок 19 - Вид экрана с результатом решения задачи

# Список используемой литературы

1. Березин, Б. И. Начальный курс С и С++ / Б. И. Березин, С. Б. Березин. – М.: Диалог – МРТИ, 1999.

2. Демидович, Е. М. Основы алгоритмизации и программирования. Язык СИ / Е. М. Демидович. – Минск : Бестпринт, 2001.

3. Керниган, Б. Язык программирования СИ / Б. Керниган, Д. Ритчи. – М.: Финансы и статистика, 1992.

4. Касаткин, А. И. Профессиональное программирование на языке СИ: от Турбо-С до *Borland С*++ / А. И. Касаткин, А.Н. Вольвачев: Справочное пособие. Минск : Выш. шк., 1992.

5. Страуструп, Б. Язык программирования C++ / Б. Страуструп: 2-е изд.: В 2 т. – Киев: ДиаСофт, 1993.

6. Больски, М. Н. Язык программирования СИ / М. Н. Больски: Справочник. – М.: Радио и связь. 1988.

7. Архангельский, А. Я. Программирование в С++ *Builder* 6 / А. Я. Архангельский. – М.: ЗАО “Издательство БИНОМ”, 2002.

8. Шилд, Г. Программирование на *Borland С*++ / Г. Шилд. – Минск : ПОПУРРИ, 1999.

9. Тимофеев, В. В. Программирование в среде С++ *Builder* 5 / В. В. Тимофеев. – М.: БИНОМ, 2000.

10. Гради Буч Объектно-ориентированный анализ и проектирование с примерами приложений на С++/Буч Г. 2-е изд. СПб: Невский Диалект,2000.