**Вычислительная математика**

**Вариант – по последней цифре в номере зачетной книжки**

**1. Решить следующие уравнения методом Ньютона с точностью до 0.01**

В0 

В1 

В2 

В3 

В4 

В5 

В6 1174 

В7 1175 

В8 1176 

В9 1177 

**2. Составить интерполяционный многочлен Лагранжа для функции, заданной таблицей. Найти приближенное значение функции в точке x0**

В0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 0 | 1 | 2 | 3 | 4 |
| y | 1 | 4 | 15 | 40 | 85 |

x0 = 1.5

В1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 2 | 4 | 6 | 8 | 10 |
| y | 0 | 3 | 5 | 4 | 1 |

x0 = 5

В2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 0 | 2 | 3 | 4 |
| y | 3 | 1 | 5 | 7 |

x0 = 2.5

В3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 1 | 3 | 4 | 6 |
| y | -7 | 5 | 8 | 14 |

x0 = 1.5

В4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 2 | 4 | 5 | 10 |
| y | 3 | 7 | 9 | 19 |

x0 = 7.0

В5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 2 | 3 | 4 | 5 | 6 | 5 |
| y | 3 | 7 | 13 | 21 | 31 | 43 | 57 |

x0 = 1.5

В6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 2.0 | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 |
| y | 0.30103 | 0.32222 | 0.34242 | 0.36173 | 0.38021 | 0.39794 |

x0 = 2.03

В7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 4.0 | 5.0 | 6.0 | 7.0 | 8.0 | 9.0 | 10.0 |
| y | 0.60206 | 0.69897 | 0.77815 | 0.84510 | 0.90309 | 0.95424 | 1.0 |

x0 = 6.5

В8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 5.0 | 6.0 | 7.0 | 8.0 |
| y | 25.0 | 36.0 | 49.0 | 64.0 |

x0 = 6.25

В9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 0 | 1 | 2 | 3 | 4 |
| y | 1 | 4 | 15 | 40 | 85 |

x0 = 2.5

**3. Численно взять интегралы, использовать методы прямоугольников, трапеций, Симпсона. Интервал интегрирования необходимо разбить на n=10 частей. Оценить погрешность по методу удвоения шага**

В0 а)  ; б) ;

В1 а)  ; б) ;

B2 а)  ; б) ;

В3 а) ; б) ;

В4 а) ; б) ;

В5 а) ; б) ;

В6 а) ; б) ;

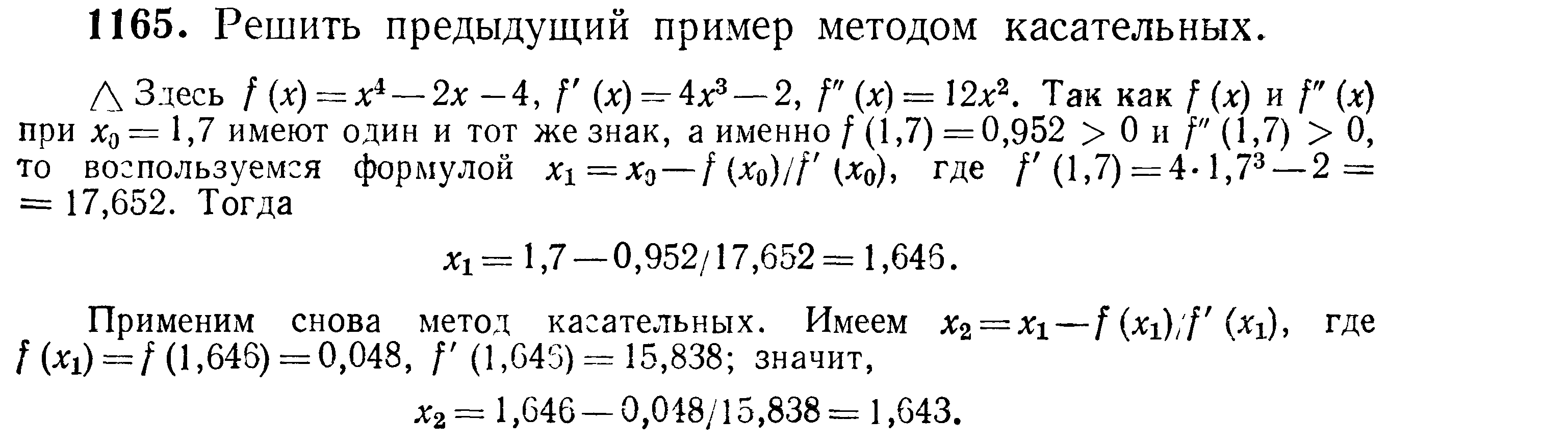
В7 а) ; б) ;

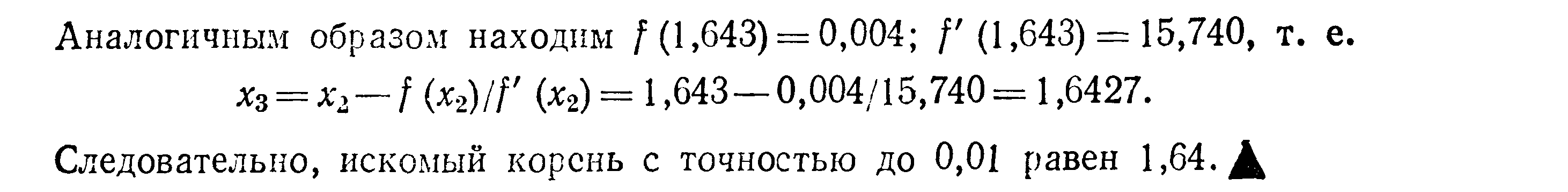
В8 а) ; б) ;

В9 а) ; б) .

**Примеры**

**1) Приближенное решение алгебраических уравнений**





Пример решения на C++

#include <iostream>

#include <cmath>

using namespace std;

int mas\_index[20];

double mas\_root[20];

double f(double x)

{

return pow(x,3)-3\*x-1;

}

double f1(double x)

{

return 3\*pow(x,2)-3;

}

double NewtonIteration(double x\_old)

{

return x\_old - f(x\_old)/f1(x\_old);

}

double NewtonAB(double a, double b, double delta)

{

double x\_old = (a+b)/2;

double x\_new = NewtonIteration(x\_old);

while(abs(x\_new-x\_old)>delta)

{

x\_old = x\_new;

x\_new = NewtonIteration(x\_old);

}

return x\_new;

}

int apply\_Newton(double a, double b, int num\_step, double eps)

{

double x\_current; // текущее значение х

double step = (b-a)/num\_step; // шаг разбиения

int num\_root = 0; // количество корней уравнения

for(int i=0; i<num\_step-1; i++)

{

x\_current = a + i\*step; // вычисляем очередное значение х

// если на промежутке (x\_current, x\_current+step)

// функция меняет знак, то на этом промежутке есть корень

if(f(x\_current)\*f(x\_current+step)<0)

{

mas\_index[num\_root] = i; // помещаем индекс промежутка в массив индексов

num\_root += 1; // наращиваем счетчик индексов

}

}

for(int i=0; i<num\_root; i++)

{

x\_current = a + mas\_index[i]\*step;

mas\_root[i]=NewtonAB(x\_current, x\_current+step, eps);

}

return num\_root;

}

int main(void)

{

int temp;// для задержки клавиатуры

double eps = 0.001;

for(int i=0; i<20; i++) // обнуляем массив индексов корней

mas\_index[i]=0;

double a = -2.0; double b = 3.0;// начало и конец промежутка

int num\_step = 100; // количество разбиений

int num\_root = apply\_Newton(a, b, num\_step, eps);

for(int i=0; i<num\_root; i++)

{

cout<<mas\_root[i]<<endl;

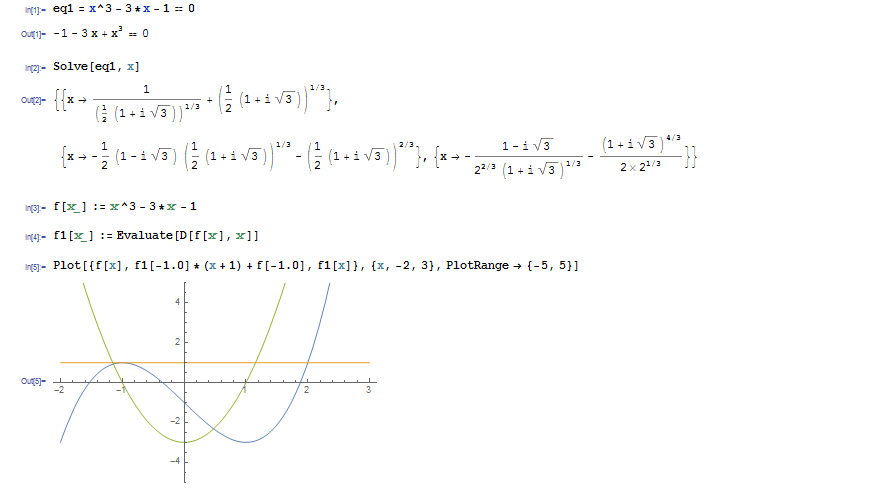
}

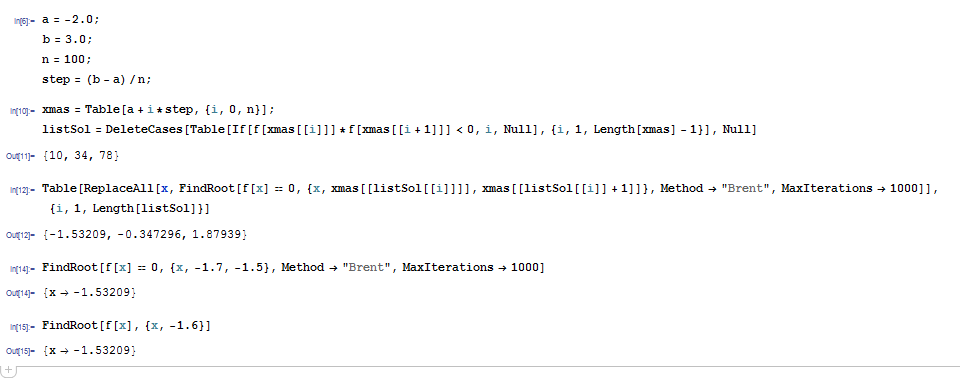
//cout<<NewtonAB(-1.7,-1.4,0.01)<<endl;

cin>>temp;

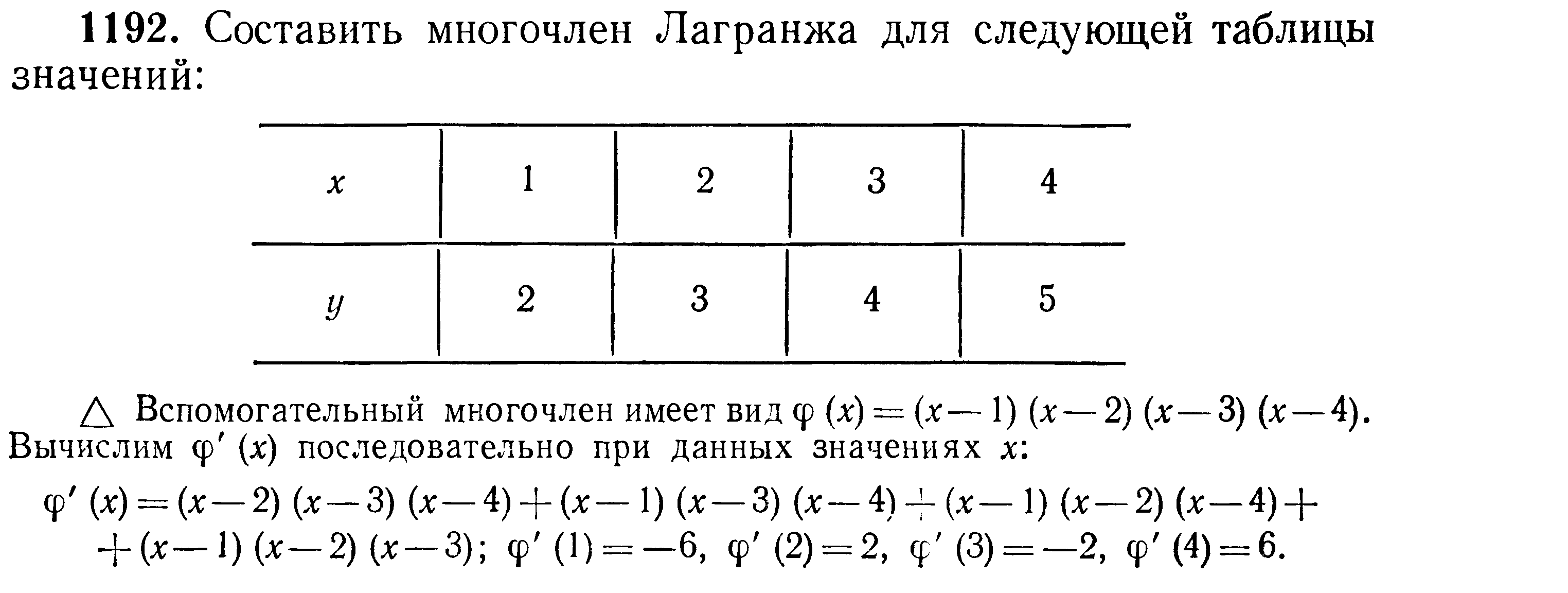
}

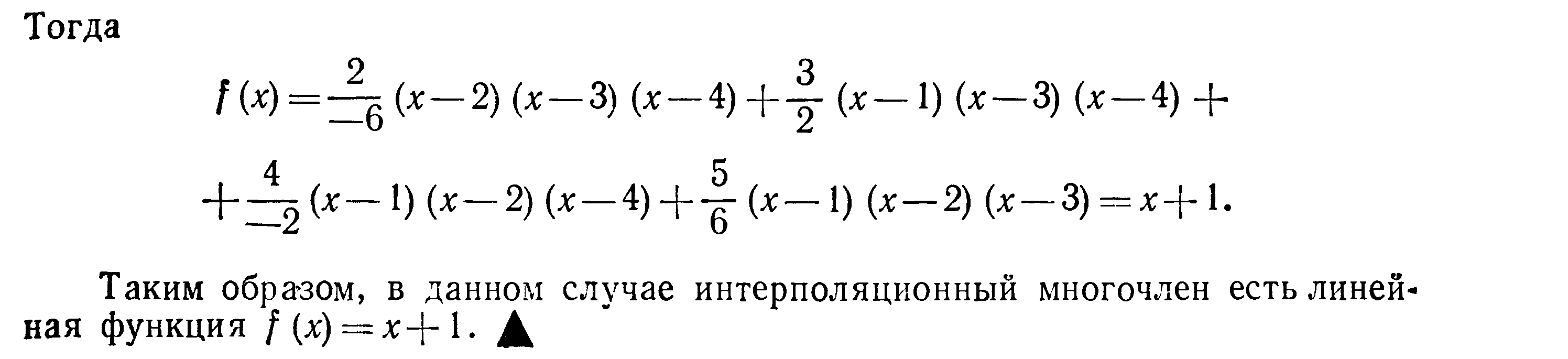
Численное решение уравнений в Wolfram Mathematica

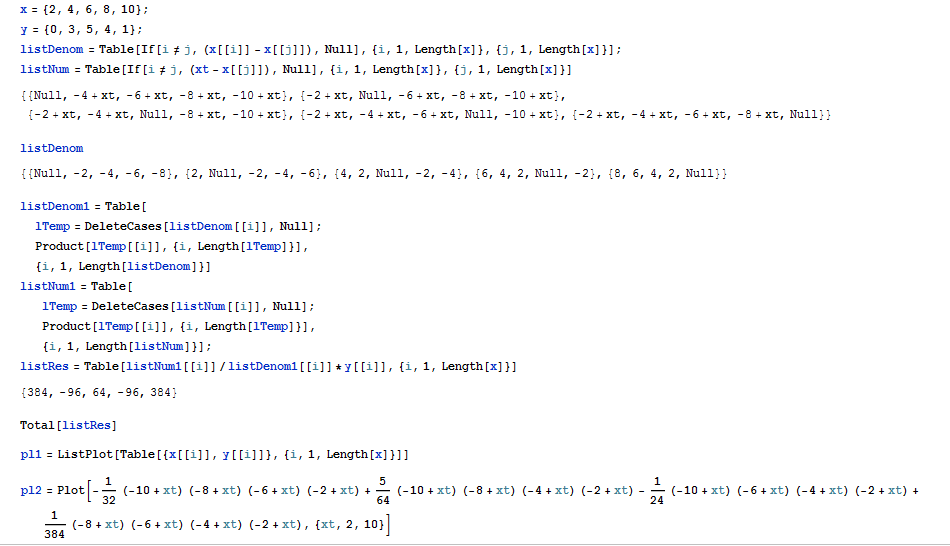


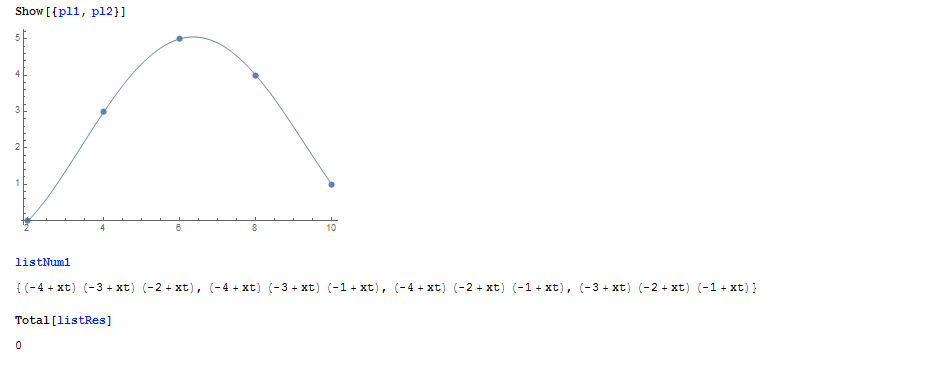


**2) Интерполирование функций. Многочлен Лагранжа.**









**3) Численное интегрирование.**

Необходимо прочитать п. 1 Численное интегрирование из книги Зельдовича и Мышкиса [1], разобраться с методом Симпсона и методом трапеций. Также необходимо прочитать и проделать по книге Данко, Попов, Кожевникова [2] из гл. 9 Методы вычислений, параграф 3 приближенное вычисление определенных интегралов номера 1202 – 1205.

Кроме того, необходимо написать программу (по примеру, приведенному в листинге 1) для вычисления интегралов, указанных в задании, на языке C++. Для справки и первоначального изучения языка можно воспользоваться материалами <http://cppstudio.com/cat/271/>.

Пример 1. Численное интегрирование без использования массивов

#include <iostream>

#include <math.h>

using namespace std;

double funk(double x)

{

return exp(-sqrt(1+pow(x,2)))+0.1\*sin(10\*x)+0.2;

}

double simpson(double a, double b, int n)

{

double h = (b-a)/n;

double x0, x1, x2;

double sum = 0;

for(int i=0; i<=n-2; i+=2)

{

x0=a+i\*h;

x1=x0+h;

x2=x0+2\*h;

sum += funk(x0) + 4\*funk(x1) + funk(x2);

}

sum \*= h/3.0;

return sum;

}

double trapezoidal(double a, double b, int n)

{

double h = (b-a)/n;

double x0, x1;

double sum = 0;

for(int i=0; i<=n-1; i++)

{

x0=a+i\*h;

x1=x0+h;

sum += funk(x0) + funk(x1);

}

sum \*= h/2.0;

return sum;

}

double rectangle(double a, double b, int n)

{

double h = (b-a)/n;

double x0;

double sum = 0;

for(int i=0; i<=n-1; i++)

{

x0=a+i\*h;

sum += funk(x0);

}

sum \*= h;

return sum;

}

int main(void)

{

int a;

cout<<funk(0.6)<<endl;

cout<<simpson(0.0, 10.0, 120)<<endl;

cout<<trapezoidal(0.0, 10.0, 480)<<endl;

cout<<rectangle(0.0, 10.0, 500000)<<endl;

cin>>a;

return 0;

}

Пример 2. Численное интегрирование (с использованием массивов)

#include <iostream>

#include <fstream>

#include <math.h>

#include <malloc.h>

using namespace std;

double func(double x)

{

return log(1+x)/(1+pow(x,2));

}

double trapezoidal\_integration(double a, double b, int n)

{

double sum = (func(a) + func(b))/2.0;

double x;

double step = (b-a)/n;

for(int i = 1; i < n; i++)

{

x = a + i \* step;

sum = sum + func(x);

}

sum = sum \* step;

return sum;

}

double simpson(double a, double b, int n)

{

double sum = 0;

double x0, x1, x2;

double step = (b-a)/n;

for(int i = 0; i <= n-2; i+=2)

{

x0 = a + i \* step;

x1 = x0 + step;

x2 = x0 + 2\*step;

sum = sum + func(x0) + 4\*func(x1) + func(x2);

}

sum = sum \* step/3.0;

return sum;

}

void simpson\_v2(double a, double b, int n, double &res)

{

double sum = 0;

double x0, x1, x2;

double step = (b-a)/n;

for(int i = 0; i <= n-2; i+=2)

{

x0 = a + i \* step;

x1 = x0 + step;

x2 = x0 + 2\*step;

sum = sum + func(x0) + 4\*func(x1) + func(x2);

}

sum = sum \* step/3.0;

res = sum;

}

double simpson\_mas(double a, double b, int n, double\* mas\_y)

{

double sum = 0;

double step = (b-a)/n;

for(int i = 1; i <= n-1; i+=2)

sum = sum + mas\_y[i-1] + 4\*mas\_y[i] + mas\_y[i+1];

sum = sum \* step/3.0;

for(int i = 0; i <= n; i++)

mas\_y[i] = 2.0 \* mas\_y[i];

return sum;

}

int main(void)

{

double a = 0.0;

double b = 1.0;

int n = 20;

double step = (b-a)/n;

// в стиле С++

double \*mas\_int = new double[n+1]; // выделили память под массив

for(int i=0; i<n+1; i++)

mas\_int[i] = func(a+i\*step); // заполнили массив

for(int i=0; i<n+1; i++)

{

cout << a+i\*step << "\t" << mas\_int[i] << endl; // выводим массив

}

cout << endl;

delete[] mas\_int; // освободить память

// в стиле С

double \*mas\_malloc = (double\*)malloc((n + 1) \* sizeof(double));

for(int i=0; i<n+1; i++)

mas\_malloc[i] = func(a+i\*step); // заполнили массив

for(int i=0; i<n+1; i++)

cout << a+i\*step << "\t" << mas\_malloc[i] << endl; // выводим массив

//free(mas\_malloc); // освободить память

cout << endl;

cout << simpson\_mas(a, b, n, mas\_malloc) << endl;

cout << endl;

for(int i=0; i<n+1; i++)

cout << a+i\*step << "\t" << mas\_malloc[i] << endl; // выводим массив

ofstream my\_file; // создали поток вывода в файл

my\_file.open("first.txt"); // открыли файл

for(int i=0; i<n+1; i++)

my\_file << a+i\*step << "\t" << mas\_malloc[i] << endl;

my\_file.close(); // закрыли файл

//cout << trapezoidal\_integration(0.0, 1.0, 400) << endl;

cout << simpson(0.0, 1.0, 20) << endl;

cout << endl;

double res\_v2 = 0;

simpson\_v2(0.0, 1.0, 20, res\_v2);

cout << res\_v2;

cin >> a;

}

1. Зельдович Я.Б., Мышкис А.Д. Элементы прикладной математики. – М.: 1965. – 616 с.

2. Данко П.Е., Попов А.Г., Кожевникова Т.Я. Высшая математика в упражнениях и задачах: Учеб. пособие для студентов втузов. В 2-х частях. Ч. II. – 4-е изд., испр. и доп. – М.: Высш. шк., 1986. – 415 с.