**Оглавление**

[Введение 2](#_Toc28597974)

[**1.** **Описание метода решения задачи** 3](#_Toc28597975)

[**2.** **Разработка структур данных** 5](#_Toc28597976)

[**3.** **Разработка алгоритмов** 6](#_Toc28597977)

[**4.** **Тестирование программы** 10](#_Toc28597978)

[**5.** **Разработка документации** 17](#_Toc28597979)

[Заключение 22](#_Toc28597980)

[Библиографический список 23](#_Toc28597981)

[Приложение А. Листинг программы 24](#_Toc28597982)

Введение

Целью моего курсового проекта является разработка программной системы для решения математической задачи методом интерполяции математических функций многочленом Лагранжа.

Задачами курсового проектирования является:

* Изучение метода решения задачи;
* Разработка алгоритма и структур данных;
* Разработка программы, позволяющей решить математическую задачу методом интерполяции математических функций многочленом Лагранжа;
* Выполнение контрольного расчёта в ручном режиме;
* Проведение тестирования работы программы на контрольном примере;
* Выполнение тестирования работы программы на нескольких тестовых примерах;
* Разработка документации к программе;

Пояснительная записка к курсовому проекту состоит из следующих разделов:

* Введение;
* Описание метода решения задачи;
* Разработка структур данных;
* разработка алгоритмов;
* Тестирование программы;
* Разработка документации;
* Заключение;
* Библиографический список;
* Приложение (листинг программных модулей).

1. **Описание метода решения задачи**
   1. **Применение метода решения задачи на практике**

Выборка экспериментальных данных представляет собой массив данных, который характеризует процесс изменения измеряемого сигнала в течение заданного времени (либо относительно другой переменной). Для выполнения теоретического анализа измеряемого сигнала необходимо найти аппроксимирующую функцию, которая свяжет дискретный набор экспериментальных данных с непрерывной функцией – интерполяционным полиномом n-степени. Одним из способов представления данного интерполяционного полинома n-степени может быть использован многочлен в форме Лагранжа.

**Интерполяционный многочлен в форме Лагранжа** – это математическая функция позволяющая записать полином n-степени, который будет соединять все заданные точки из набора значений, полученных опытным путём или методом случайной выборки в различные моменты времени с непостоянным временным шагом измерений.

* 1. **Интерполяционная формула Лагранжа**

В общем виде интерполяционный многочлен в форме Лагранжа записывается в следующем виде:

**Где – это** степень полинома **;**

**– это** значение интерполирующей функции в точке ;

– это базисные полиномы (множитель Лагранжа), которые определяются по формуле:

Так, например, интерполяционный многочлен в форме Лагранжа, проходящий через три заданных точки, будет записываться в следующем виде:

Многочлен в форме Лагранжа в явном виде содержит значения функций в узлах интерполяции, поэтому он удобен, когда значения функций меняются, а узлы интерполяции неизменны. Число арифметических операции, необходимых для построения многочлена Лагранжа, пропорционально и является наименьшим для всех форм записи.

* 1. **Методика вычисления полинома в форме Лагранжа**

Алгоритм вычисления полинома в форме Лагранжа позволяет разделить задачи определения коэффициентов и вычисления значений полинома при различных значениях аргумента:

1. В качестве исходных данных задается выборка из n-точек, которая включает в себя значения функции и значения аргумента функции.
2. Выполняется вычисление полинома n-степени в форме Лагранжа по следующей формуле:
3. **Разработка структур данных**

Для решения задачи необходимо определить типы данных, которые будут использоваться для хранения исходных и промежуточных данных, а также результатов вычислений.

Для реализации математических вычислений необходимы данные вещественного типа. Язык *Free Pascal* предоставляет возможность использования вещественных типов данных:

Вещественные типы данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип** | **Диапазон значений** | **Значимых разрядов** | **Размер, байт** |
| Real | Зависит от платформы | Зависит от платформы | 4 или 8 |
| Single | 1.5E-45 ... 3.4E38 | 7-8 | 4 |
| Double | 5.0E-324 ... 1.7E308 | 15-16 | 8 |
| Extended | 1.9E-4932 ... 1.1E4932 | 19-20 | 10 |
| Comp | -2E64+1 ... 2E63-1 | 19-20 | 8 |
| Currency | -922337203685477.5808 | 922337203685477.5807 | 8 |

Тип данных *Real* зависит от платформы, и использовать его нежелательно. Тип *Currency* предназначен для операций с числами с фиксированной точностью (операции с денежными величинами). Тип *Comp* трактуется как вещественное число без дробной части.

Точность данных типа *Single* недостаточна для решения задач вычислительной математики, поэтому предпочтение следует отдать типам *Double* или *Extended*.

1. **Разработка алгоритмов**

Процесс вычисления полинома в форме Лагранжа является итерационным процессом, но в моём случае представляет собой линейный процесс поэтому за основу в качестве обобщенной схемы возьмём общую схему линейного алгоритма, представленную на рисунке (Рисунок 3.1).



*Рисунок 3.1 – Общая схема линейного алгоритма*

1. Входные данные алгоритма:
   1. ;
   2. ;
2. Блок вычисления содержит в себе формулы для вычисления полинома в форме Лагранжа:
   1. Вычисление :
   2. Вычисление :
   3. Вычисление :
   4. Вычисление :
   5. Вычисление :
   6. Вычисление :

Дополнительно вычисляем:

* 1. Вычисление ;
  2. Вычисление ;
  3. Вычисление ;
  4. Вычисление ;
  5. Вычисление ;

1. Выходные данные алгоритма:
   1. Основные данные:
   2. Дополнительные данные:



*Рисунок 3.2 – Подробная схема линейного алгоритма*

1. **Тестирование программы**

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом

Существует следующее определение тестирования, тестирование – это процесс выполнения программы с целью нахождения ошибок.

Для тестирования программы вычисления полинома в форме Лагранжа необходимо:

* Выбрать функцию;
* Взять произвольные точки:

;

* Рассчитать вручную:

;

* При помощи другого программного средства проверить найденное решение с тем, что будет выдано программой;

Проведём тестирование программы для функции:

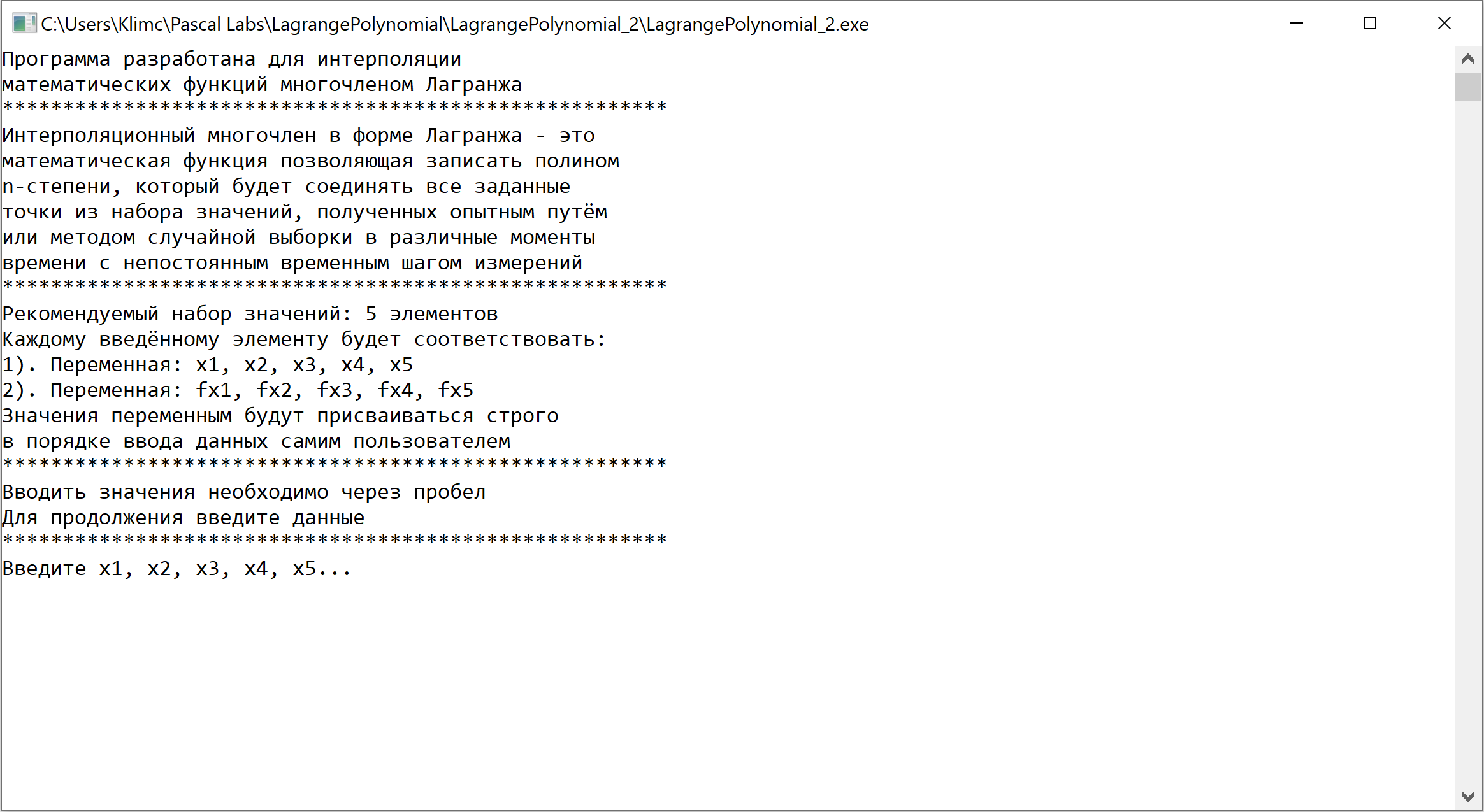
Запишем произвольные точки:

|  |
| --- |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

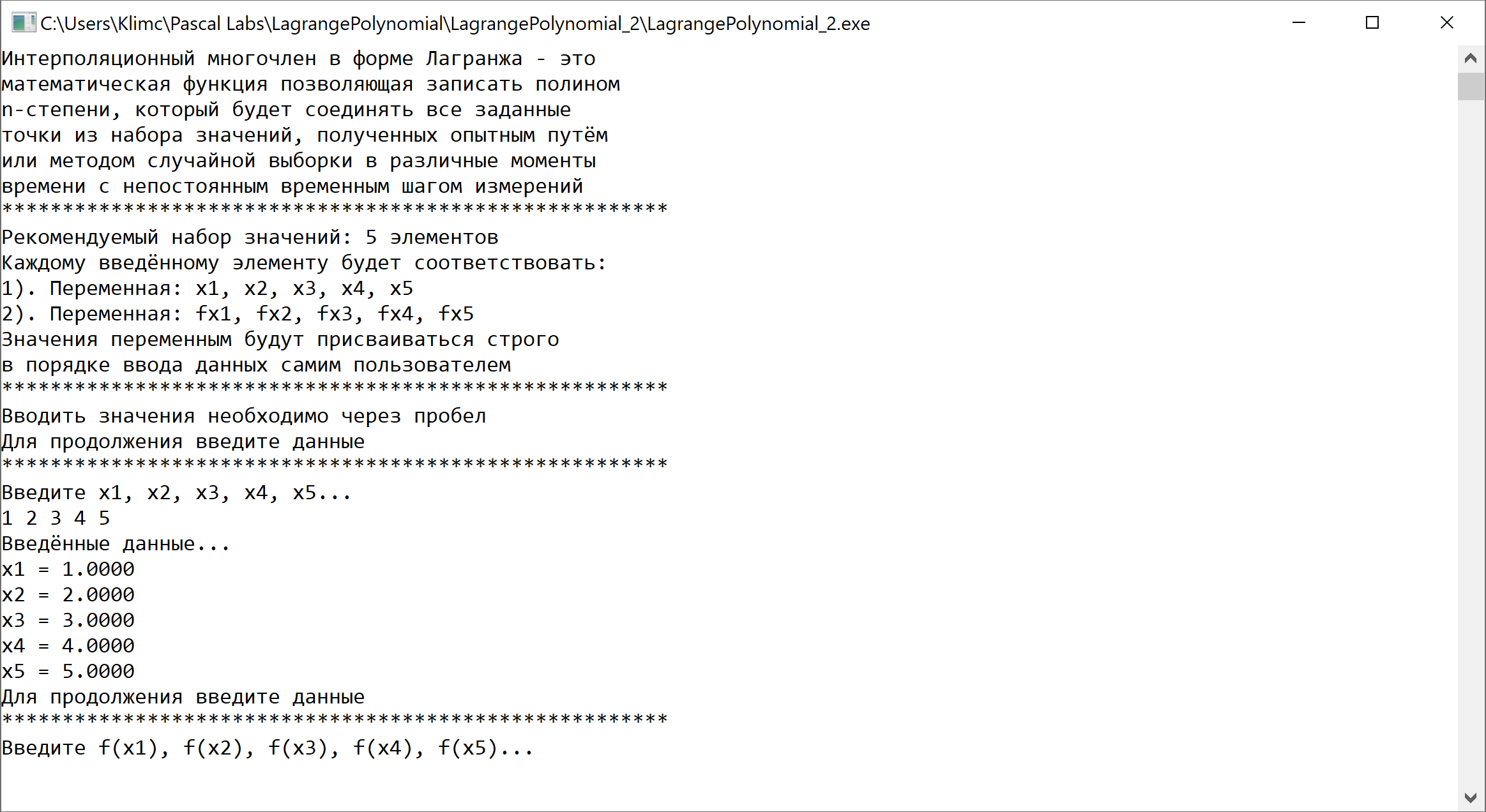
Рассчитаем вручную функцию:

|  |
| --- |
| 1 |
| 4 |
| 9 |
| 16 |
| 25 |

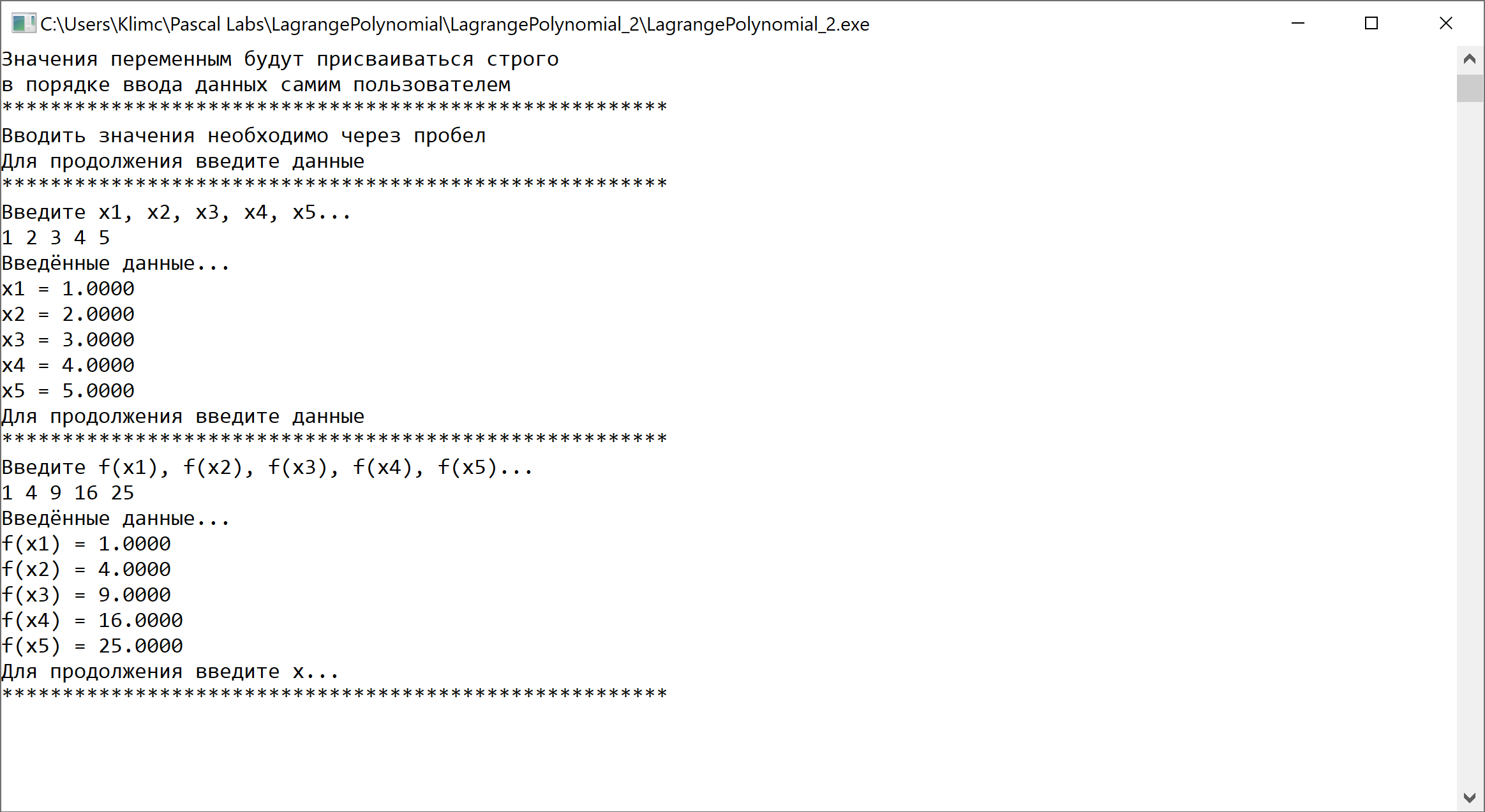
Вводим поочерёдно значения:



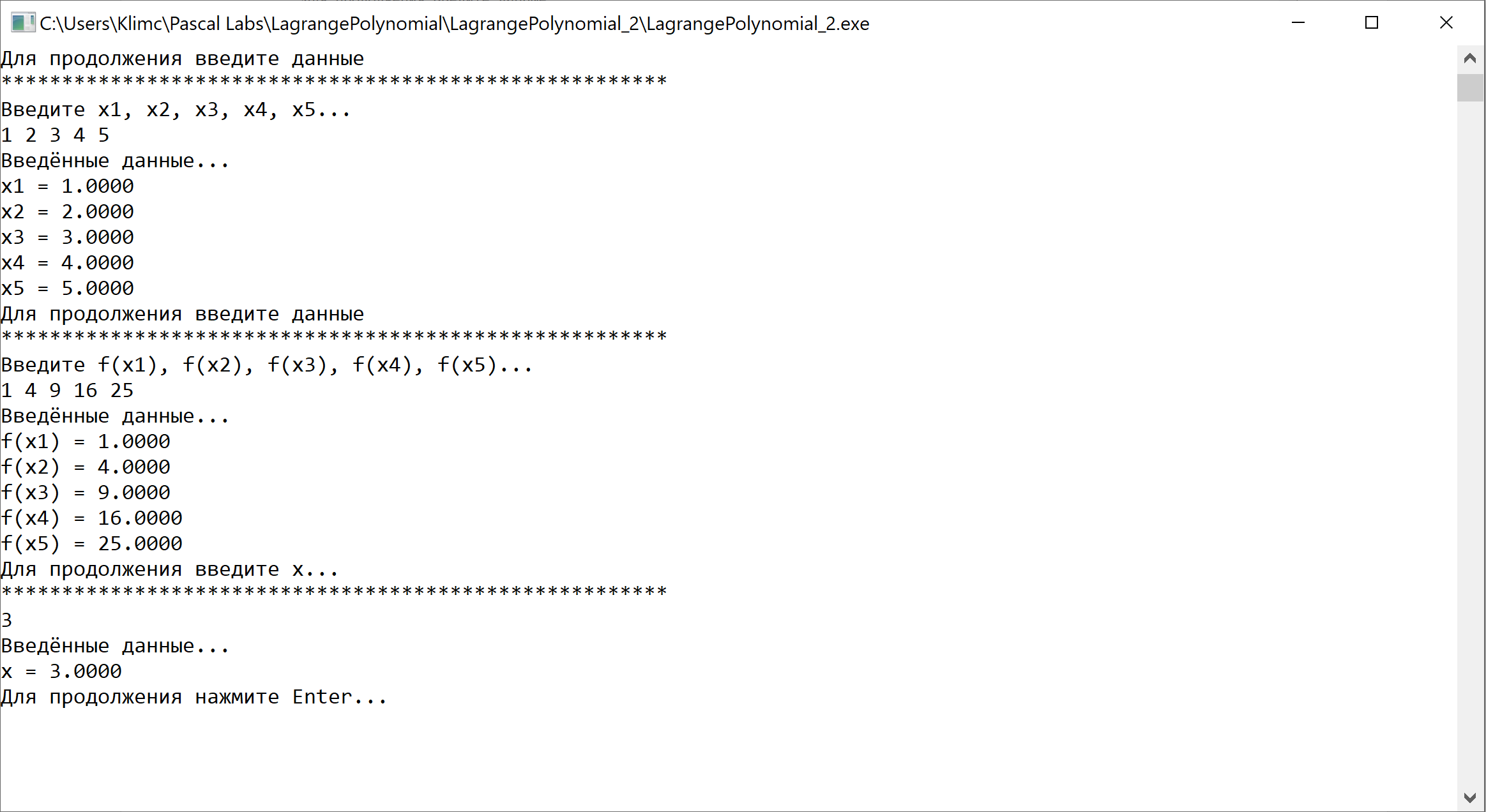
*Рисунок 4.1 – Запуск программы*



*Рисунок 4.2 – Ввод*



*Рисунок 4.3 – Ввод*



*Рисунок 4.4 – Ввод*



*Рисунок 4.5 – Результат выполнения программы*

Построим график опираясь на полученные результаты:

*Рисунок 4.6 – Основной график*

*Рисунок 4.7 – Дополнительный график*

Для наглядности применения полинома в форме Лагранжа я приведу примеры ещё двух функций:

Проведём тестирование программы для функции:

|  |  |  |
| --- | --- | --- |
| 1 | 1 | -0,1019 |
| 4 | 2 | 1,8624 |
| 9 | 3 | 0,5893 |
| 16 | 4 | -0,1129 |
| 25 | 5 | 0,0121 |

*Рисунок 4.8 – Основной график*

Проведём тестирование программы для функции:

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 0 |
| 2 | 0,5 | 0 |
| 4 | 0,25 | 0 |
| 8 | 0,125 | 0,125 |
| 16 | 0,0625 | 0 |

*Рисунок 4.9 – Основной график*

1. **Разработка документации**

В данном разделе приведено руководство программиста по использованию программной системы для решения математической задачи методом интерполяции математических функций многочленом Лагранжа.

* 1. **Назначение и условия применения программы**

Программа LagrangePolynomial предназначена для решения математической задачи методом интерполяции математических функций многочленом Лагранжа. Программа позволяет задавать пять любых значений необходимой функции и находить .

Для выполнения программы необходимо наличие операционной системы Windows. Особые требования к составу периферийных устройств не предъявляются.

Персонал, использующий программу (программист) должен обладать практическими навыками работы с текстовым пользовательским интерфейсом.

* 1. **Характеристика программы**

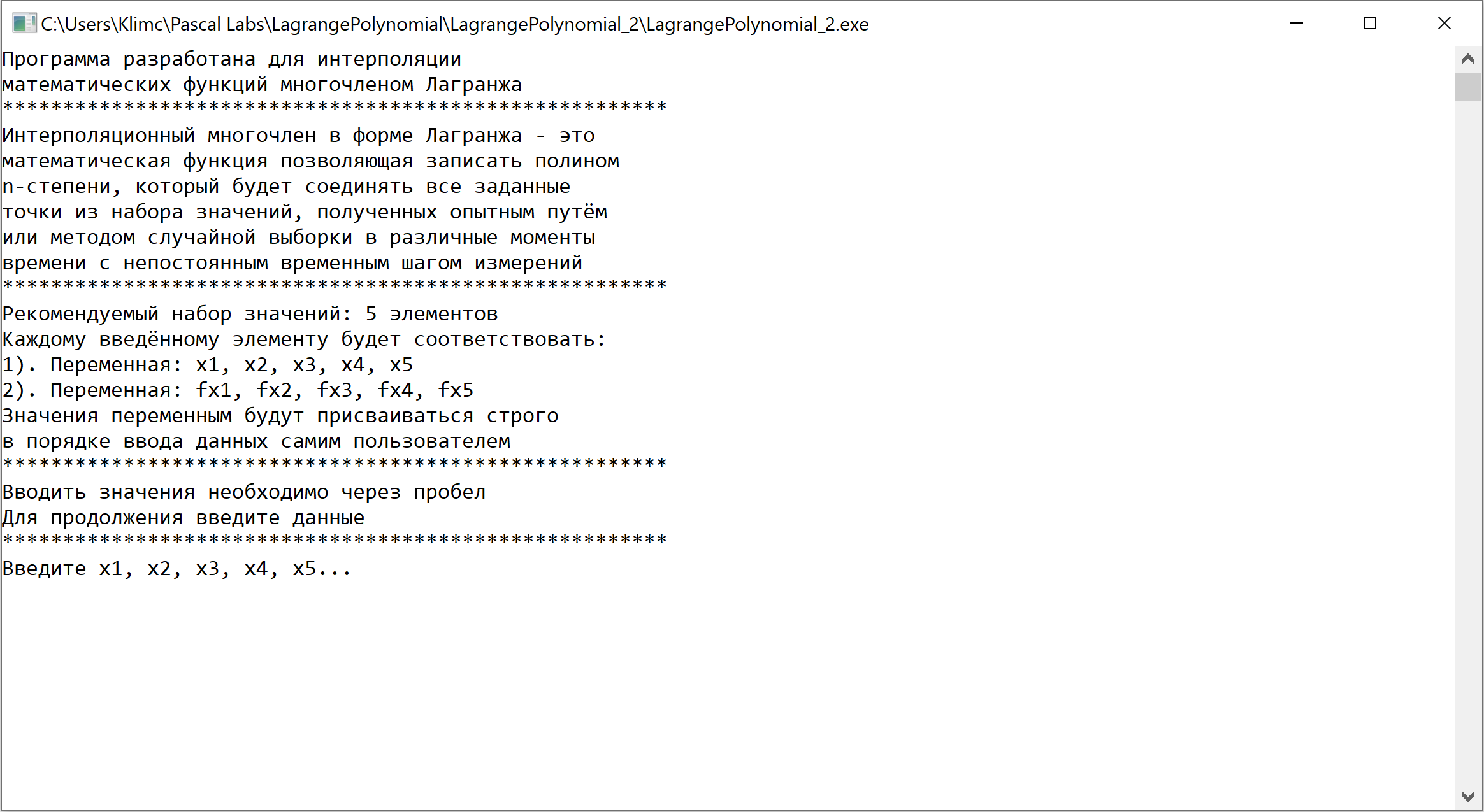
Программа LagrangePolynomial является независимым программным обеспечением и обеспечивает решение задач вычислительной математики – метод интерполяции математических функций многочленом Лагранжа.

* Работа программы осуществляется в консольном режиме.
* Размер исполняемого файла LagrangePolynomial.exe – 98 606 байт.
  1. **Обращение к программе**

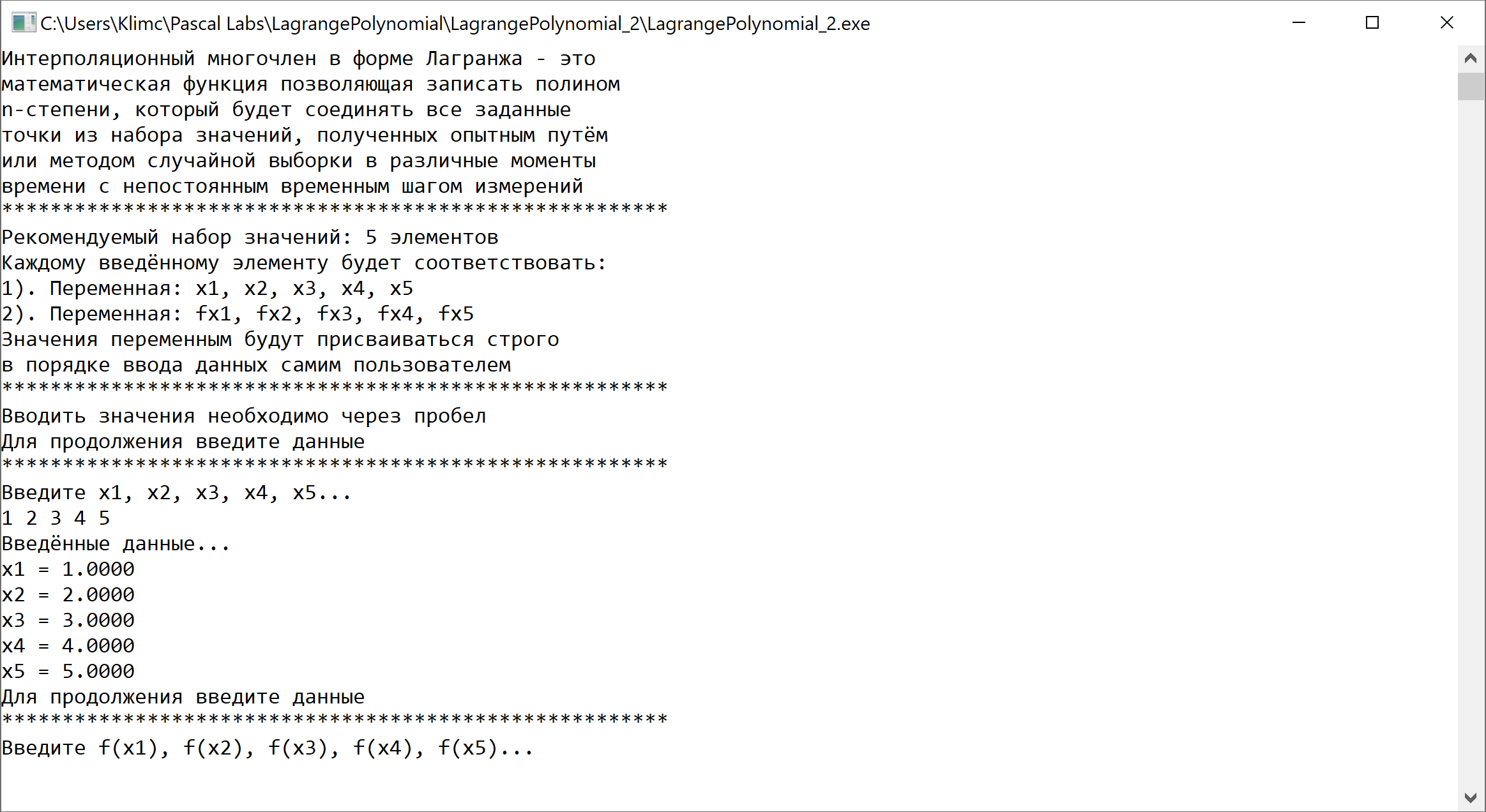
Запуск программы осуществляется стандартным для операционной системы способом – двойным щелчком левой кнопки мыши на исполняемом файле LagrangePolynomial.exe или выполнением команды в LagrangePolynomial в командной строке в каталоге программы.

После запуска в окне консоли (Рисунок 5.1) появляется следующая информация:

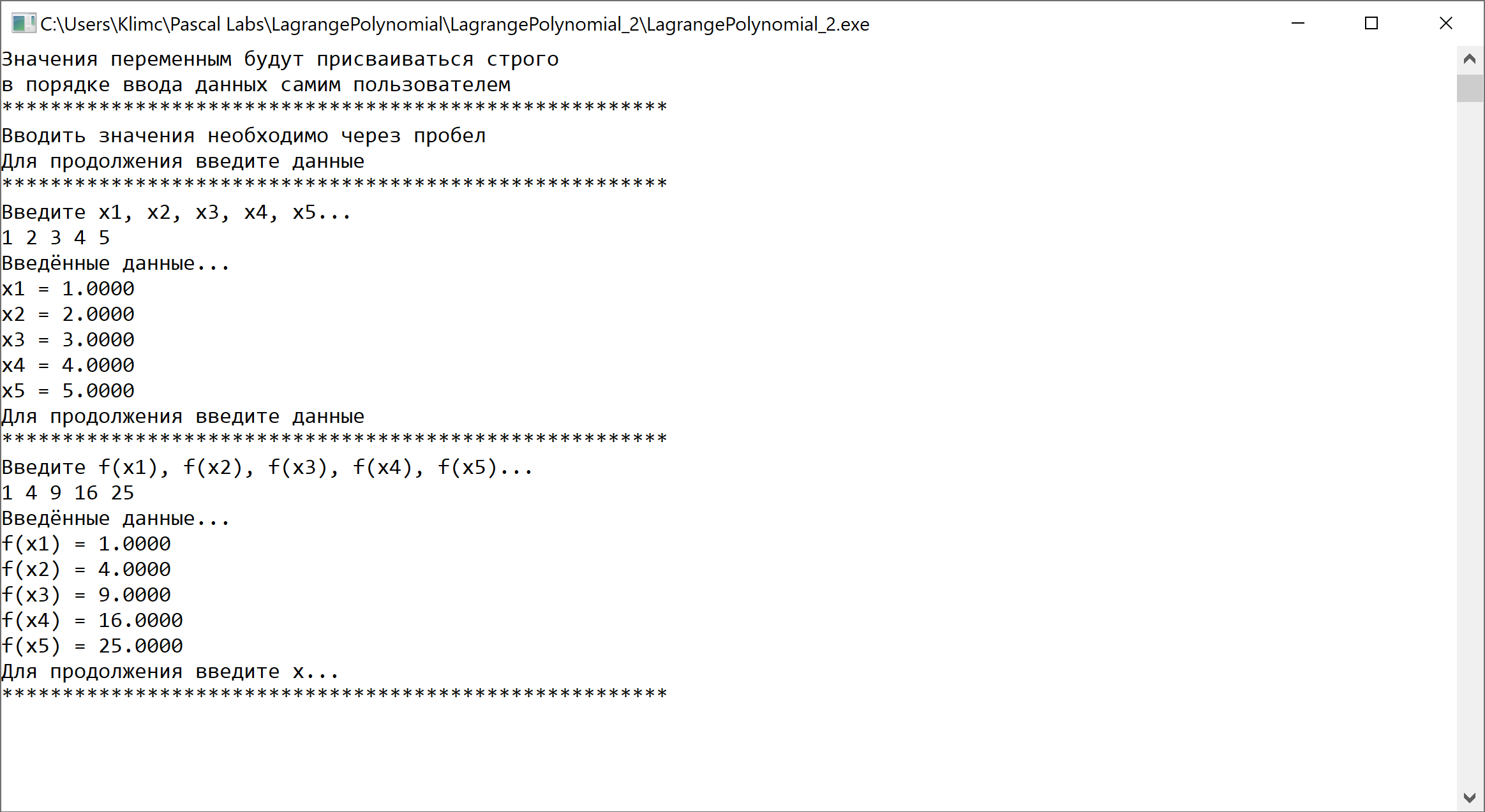
* Назначение программы;
* Описание метода;
* Параметры алгоритма.



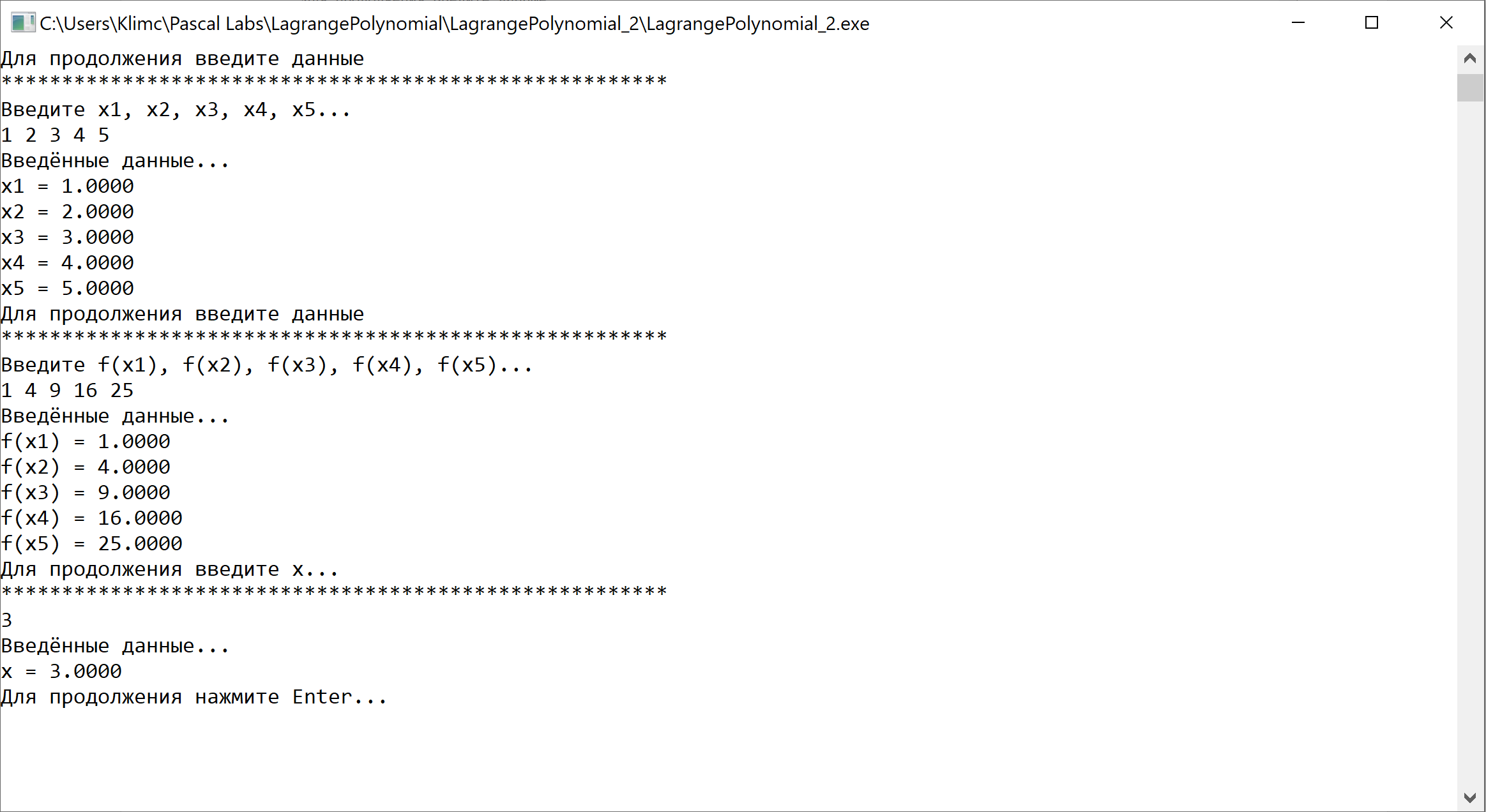
*Рисунок 5.1 – Запуск программы*



*Рисунок 5.2 – Ввод*



*Рисунок 5.3 – Ввод*



*Рисунок 5.4 – Ввод*



*Рисунок 5.5 – Результат выполнения программы*

* 1. **Входные и выходные данные**

Входные данные алгоритма:

Выходные данные алгоритма:

Основные данные:

Дополнительные данные:

Заключение

В ходе данной курсовой работы была решена задача интерполяции математических функций многочленом Лагранжа. Разработан алгоритм для решения математической задачи методом интерполяции математических функций многочленом Лагранжа. По этому алгоритму составлена и отлажена программа для интерполяции значения функции в заданной точке с помощью многочленом Лагранжа.

Библиографический список

1. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
2. ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.
3. ГОСТ 19.104-78\* ЕСПД. Основные надписи.
4. ГОСТ 19.105-78\* ЕСПД. Общие требования к программным документам.
5. ГОСТ 19.106-78\* ЕСПД. Общие требования к программным документам, выполненным печатным способом.
6. ГОСТ 19.504-79\* ЕСПД. Руководство программиста. Требования к содержанию и оформлению.
7. ГОСТ 19.604-78\* ЕСПД. Правила внесения изменения в программные документы, выполненные печатным способом.
8. Самарский А.А., Гулин А.В. Численные методы. - М.: 1989.
9. Калиткин Н.Н. Численные методы. - М.: Наука, 1978.
10. Демидович Б.П., Марон И.А. Основы вычислительной математики. - М.: 1970.
11. Копченова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. - М.: Наука, 1972.
12. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран, Паскаль. – Томск: МП «Раско», 1991.

Приложение А. Листинг программы

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

{ }

{ Проект LagrangePolynomial }

{ Copyright (c) 2019 ФГБОУ ВО РГРТУ им. В.Ф. Уткина }

{ ФВТ/кафедра КТ }

{ }

{ Разработчик: ст. гр. 848 Климчук Н.В }

{ Модифицирован: 20 декабря 2019 года }

{ }

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

program LagrangePolynomial\_2;

{$codepage utf8}

var

x, x1, x2, x3, x4, x5: extended;

fx1, fx2, fx3, fx4, fx5: extended;

a1, b2, c3, d4, e5: extended;

a, b, c, d, e, f: extended;

g, h, i, j, k: extended;

begin

writeln('Программа разработана для интерполяции');

writeln('математических функций многочленом Лагранжа');

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Интерполяционный многочлен в форме Лагранжа – это');

writeln('математическая функция позволяющая записать полином');

writeln('n-степени, который будет соединять все заданные');

writeln('точки из набора значений, полученных опытным путём');

writeln('или методом случайной выборки в различные моменты');

writeln('времени с непостоянным временным шагом измерений');

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Рекомендуемый набор значений: 5 элементов');

writeln('Каждому введённому элементу будет соответствовать:');

writeln('1). Переменная: x1, x2, x3, x4, x5');

writeln('2). Переменная: fx1, fx2, fx3, fx4, fx5');

writeln('Значения переменным будут присваиваться строго');

writeln('в порядке ввода данных самим пользователем');

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Вводить значения необходимо через пробел');

writeln('Для продолжения введите данные');

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Введите x1, x2, x3, x4, x5...');

readln(x1, x2, x3, x4, x5);

writeln('Введённые данные...');

writeln('x1 = ', x1:5:4);

writeln('x2 = ', x2:5:4);

writeln('x3 = ', x3:5:4);

writeln('x4 = ', x4:5:4);

writeln('x5 = ', x5:5:4);

writeln('Для продолжения введите данные');

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Введите f(x1), f(x2), f(x3), f(x4), f(x5)...');

readln(fx1, fx2, fx3, fx4, fx5);

writeln('Введённые данные...');

writeln('f(x1) = ', fx1:5:4);

writeln('f(x2) = ', fx2:5:4);

writeln('f(x3) = ', fx3:5:4);

writeln('f(x4) = ', fx4:5:4);

writeln('f(x5) = ', fx5:5:4);

writeln('Для продолжения введите x...');

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

readln(x);

writeln('Введённые данные...');

writeln('x = ', x:5:4);

writeln('Для продолжения нажмите Enter...');

readln();

// Вычисление f(x1)\*a;

a:=(((x-x2)\*(x-x3)\*(x-x4)\*(x-x5))/((x1-x2)\*(x1-x3)\*(x1-x4)\*(x1-x5)));

a1:=fx1\*a;

// Вычисление f(x2)\*b;

b:=(((x-x1)\*(x-x3)\*(x-x4)\*(x-x5))/((x2-x1)\*(x2-x3)\*(x2-x4)\*(x2-x5)));

b2:=fx2\*b;

// Вычисление f(x3)\*c;

c:=(((x-x1)\*(x-x2)\*(x-x4)\*(x-x5))/((x3-x1)\*(x3-x2)\*(x3-x4)\*(x3-x5)));

c3:=fx3\*c;

// Вычисление f(x4)\*d;

d:=(((x-x1)\*(x-x2)\*(x-x3)\*(x-x5))/((x4-x1)\*(x4-x2)\*(x4-x3)\*(x4-x5)));

d4:=fx4\*d;

// Вычисление f(x5)\*e;

e:=(((x-x1)\*(x-x2)\*(x-x3)\*(x-x4))/((x5-x1)\*(x5-x2)\*(x5-x3)\*(x5-x4)));

e5:=fx5\*e;

// Вычисление L(x);

f:=a1+b2+c3+d4+e5;

// Вычисление x1 - f(x1)\*a;

g:=(x1-a1);

// Вычисление x2 - f(x2)\*b;

h:=(x2-b2);

// Вычисление x3 - f(x3)\*c;

i:=(x3-c3);

// Вычисление x4 - f(x4)\*d;

j:=(x4-d4);

// Вычисление x5 - f(x5)\*e;

k:=(x5-e5);

// Вывод результатов;

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Промежуточные результаты');

writeln('f(x1)\*a = ', a1:5:4);

writeln('f(x2)\*b = ', b2:5:4);

writeln('f(x3)\*c = ', c3:5:4);

writeln('f(x4)\*d = ', d4:5:4);

writeln('f(x5)\*e = ', e5:5:4);

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Итоговые результаты');

writeln('L(x) = ', f:5:4);

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Дополнительные результаты');

writeln('x1 - f(x1)\*a = ', g:5:4);

writeln('x2 - f(x2)\*b = ', h:5:4);

writeln('x3 - f(x3)\*c = ', i:5:4);

writeln('x4 - f(x4)\*d = ', j:5:4);

writeln('x5 - f(x5)\*e = ', k:5:4);

writeln('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

writeln('Для завершения работы нажмите Enter...');

readln();

end.