**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра систем автоматизированного проектирования**

отчет

**по лабораторной работе №1**

**по дисциплине «Программирование»**

Тема: Изучение и практическое освоение приемов программирования элементарных операций и организации ввода/вывода с использованием библиотек stdio и iostream

|  |  |  |
| --- | --- | --- |
| Студент гр. 1309 |  | Игнатьев А.В |
| Преподаватель |  | Калмычков В.А |

Санкт-Петербург

2021

Оглавление

[**Исходная формулировка задания** 0](#_Toc85577038)

[**Контрольный пример** 0](#_Toc85577039)

[**Математическая постановка задания** 1](#_Toc85577040)

[**Разработка интерфейса** 1](#_Toc85577041)

[**Способ организации ввода-вывода** 1](#_Toc85577042)

[**Особенности выполнения на компьютере** 1](#_Toc85577043)

[**Представление алгоритма решения задачи** 1](#_Toc85577044)

[**Блок-схема** 4](#_Toc85577045)

[**Тесты программы.** 5](#_Toc85577047)

[**Результаты работы программы** 5](#_Toc85577048)

[**Вывод о проделанной работе** 6](#_Toc85577049)

# **Исходная формулировка задания**

Написать программу для вычислений значения полиномов, а также деление полинома на полином. (№30) (№32)

В качестве ответа предоставить деление первого полинома №30 на полином №32.

Представить программу в виде двух версий:

1) С помощью библиотеки <iostream>  
2) С помощью библиотеки <stdio.h>

## **Контрольный пример**

Уравнение №30

1) x=0.5

2) =0.0009765625

3)=-0.0023646484

4) =0.00390625

5)-205.2048046875

6)

7)

5)

6)

Уравнение №32

1)=

2)

3) =0.125

4) =0.125

5) + 0.125 - 0.125 = -0.015625000000

Умножение уравнения №30 на уравнение №32:

### **Математическая постановка задания**

Дано: вводимое число x вещественного типа.

Цель: найти значение деления полинома на полином:

Способ решения: поэтапное разложение полиномов, их дальнейшее деление. Выносим за скобку неизвестную переменную х с самой маленькой степенью, пока в итоге не получим выражения следующего вида:

(((x - 1) \* x + 2) \* x - 1) \* x \* x \* x.

((-2.4214 \* x \* x - 52532.43) \* x \* x + 624.3) \* x \* x \* x \* x \* x \* x.

Вычисление индивидуальных значений, дальнейшее деление полинома на полином.

#### **Разработка интерфейса**

|  |  |
| --- | --- |
| cout | Функция вывода значения на экран |
| Cin | Функция ввода значения переменной |
| printf | Функция вывода значения на экран |

##### **Способ организации ввода-вывода**

Для ввода и вывода ответов мы используем разные конструкции. В первых двух случаях мы выводим ответ, используя:

Ввиду того, что в третьем случае мы используем функцию printf, соответственно, при выводе ответа также будет уместно использовать ее:

Для ввода значений мы будем использовать нехитрую конструкцию. Отдельный элемент уравнения / все уравнение целиком мы будем обозначать какой-либо переменной, дабы сократить количество набираемых символов и ускорить процесс выполнения поставленной задачи. Например: в первом способе решения мы обозначаем целое уравнение за perv и vtor соответственно; во втором случае мы по возрастанию элементы уравнения обозначаем как pn\_46, где n – порядковый номер.

|  |  |  |
| --- | --- | --- |
|  | iostream | Stdio.h |
| Ввод | cin |  |
| Вывод | cout | Printf() |

###### **Особенности выполнения на компьютере**

Для вывода на экран результата после деления полинома на полином в каждом из способов решения отведено разное количество знаков после запятой. Например, при выводе ответа в первом случае количество знаков после запятой равно d1. Во втором случае знаков после запятой выводится d2. В третьем же случае мы выводим 6 знаков после запятой.

**Представление алгоритма решения задачи**

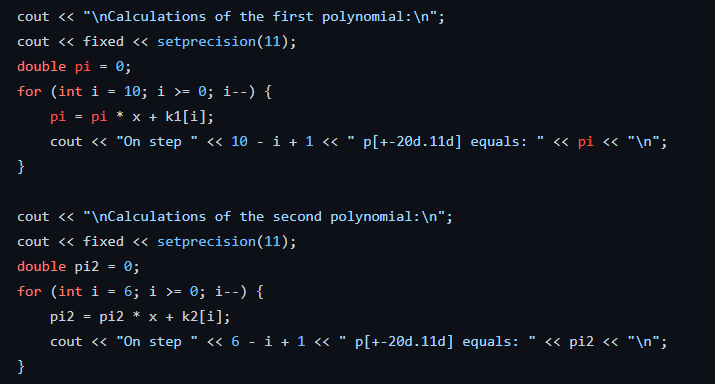
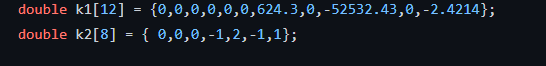
Функционал программы предельно прост и понятен.

В первом аналоге программы мы раскладываем полиномы по формулам Герона. Происходит это путем вынесения за скобки, тем самым совершая произведения степеней, что в конечном итоге дает нам полноценную формулу, но более удобную для вычисления руками, поскольку степени меньше => легче считать.

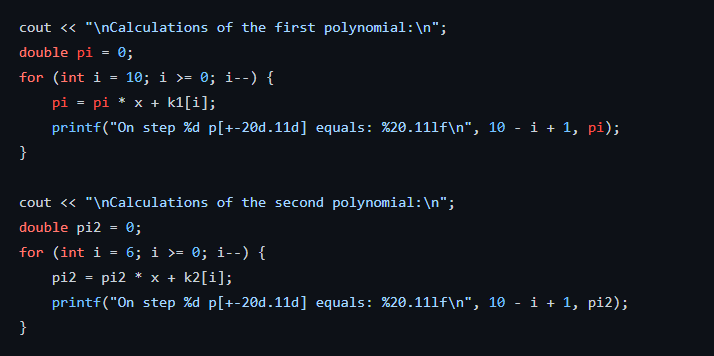
Разложив таким образом обе формулы, мы делим полином №16 на полином №46, получая итоговый результат. От пользователя же требуется ввести лишь значение Х, которое вследствие заменит переменную Х, от которой и будет в дальнейшем зависеть ответ.

C:\Users\admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2.png

Второй аналог программы не сильно отличается от первого. Единственное различие состоит в том, что действия выполняются постепенно. Умножая первую скобочку на какой-либо многочлен, мы в делаем это в цикле, дабы в дальнейшем не переписывать численные значения и упрощать процесс подсчета. Можно с уверенностью сказать, что действия выполняются поочередно, таким образом, легче вычислить ошибку, если таковая в записи исходного неравенства будет существовать.



Различий между второй и третьей формой записи практически нет, разве что конструкцию count мы заменяем на printf, который в дальнейшем служит нам выводом ответа в третьем типе уравнения.



**Блок-схема**



Текст программы:

#include <iostream>

#include <iomanip>

#include <stdio.h>

using namespace std;

// –2.4214 x^10–52532.43x^8+624.3x^6.

// x^6–x^5+2x^4–x^3.

// Autor: Ignatiev Valerii Alekseevich 1309

// data: 22.09.2021

double k1[12] = { 0,0,0,0,0,0,624.3,0,-52532.43,0,-2.4214 };

double k2[8] = { 0,0,0,-1,2,-1,1 };

int main() {

cout << "Autor: Ignatiev Valerii Alekseevich 1309\n"

<< "Version 1.1.1\n"

<< "Start: 22.09.2021\n"

<< "End: 22.09.2021\n\n";

cout << "First polynomial: -2.4214x^10-52532.43x^8+624.3x^6\n"

<< "Second polynomial: x^6-x^5+2x^4-x^3\n\n";

double x;

cout << "Enter x[-5; 5][+-d.9d]: ";

cin >> x;

if (x < -5 or x > 5) {

cout << "Oops wrong x\n";

return 0;

}

cout << "\nCalculations of the first polynomial:\n";

double pi = 0;

for (int i = 10; i >= 0; i--) {

pi = pi \* x + k1[i];

printf("On step %d p[+-20d.11d] equals: %20.11lf\n", 10 - i + 1, pi);

}

cout << "\nCalculations of the second polynomial:\n";

double pi2 = 0;

for (int i = 6; i >= 0; i--) {

pi2 = pi2 \* x + k2[i];

printf("On step %d p[+-20d.11d] equals: %20.11lf\n", 10 - i + 1, pi2);

}

printf("\nAnswer for first polynomial[+-20d.12d]: %20.11lf\nAnswer for second polynomial[+-20d.12d]: %20.12lf\nTotal answer[+-20d.12d]: %20.12lf\n\n,", pi, pi2, pi \* pi2);

return 0;

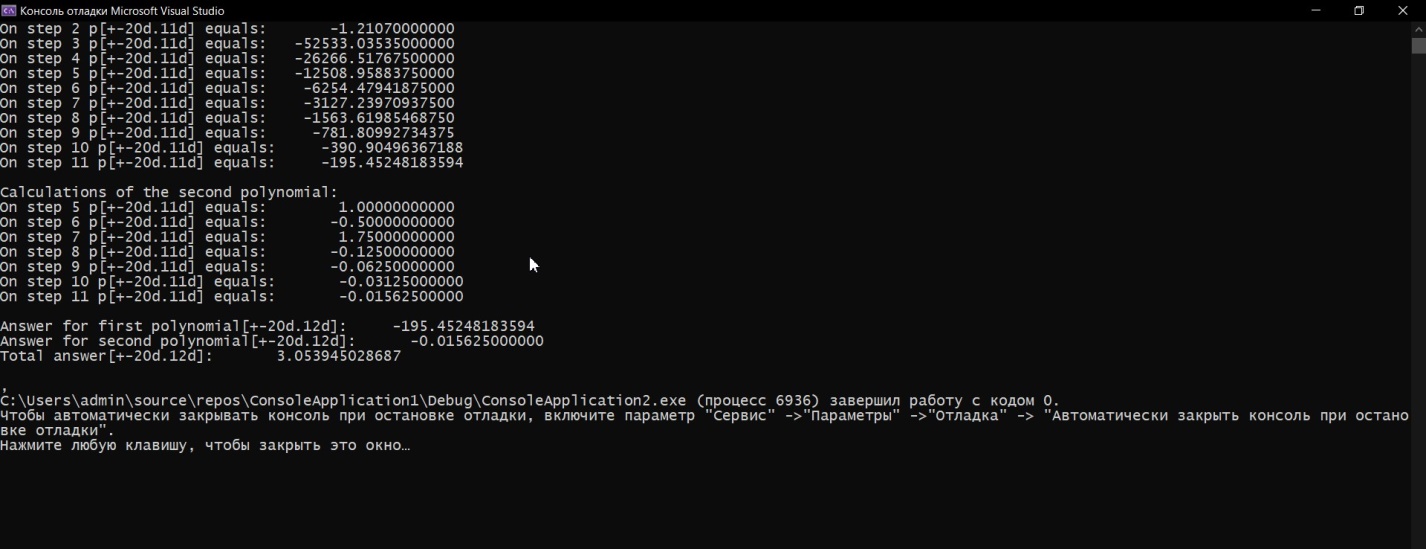
}

**Тесты программы.**

В качестве тестов созданной нами программы, используем такие числовые значения переменной Х: 0.5;

**Результаты работы программы**

Результат работы программы с заданным числом 0.5:



**Вывод о проделанной работе**

Благодаря проделанной работе нам удалось освоить библиотеки <stdio.h> и <iosteam>. Более того, надо удалось познать некоторые основные конструкции языка С++.

Также, нам удалось найти сразу несколько способов решения, казалось бы, однозначного задания.