Text

Description automatically generated**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И ИССЛЕДОВАНИЙ РЕСПУБЛИКИ МОЛДОВА**

**Технический университет Молдовы**

**Факультет вычислительной техники, информатики и микроэлектроники**

**Кафедра компьютерных наук и системной инженерии**

**ОТЧЕТ**

Лабораторная работа № 1

на курсе «Компьютерное программирование»

**Тема: Использование операторов условия и операторов цикла на языке С++.**

Выполнил: Св. гр. СR-253 Кывыржик Алексей

Проверено: Asis.univ. Фалько Николай

**Кишинев 2025**

**Задание:**

**Вариант 6**

Вычислить и вывести на экран в виде таблицы значения функции F на интервалеот Хнач. до Хкон. с шагом dX.



*ax2 + b2x + c при c < 0 и b ≠ 0*

# *F = при c > 0 и b = 0*

* в остальных случаях*

где а, b, с - действительные числа.

Функция F должна принимать действительное значение, если выражение (Ац И Вц) ИЛИ (Ац И Сц) не равно нулю, и целое значение в противном случае. Через Ац, Вц и Сц обозна­чены целые части значений а, b, с, операции И и ИЛИ - поразрядные. Значения а, b, с, Хнач., Хкон., dX ввести с клавиатуры.

|  |  |  |  |
| --- | --- | --- | --- |
| Условие | Мой код | Код чатжпт | Сравнение |
| **Вычислить и вывести на экран значения аргумента и значения функции, заданной несколькими выражениями, для заданного интервала хн ≤ x ≤ хк и шага dx изменения аргумента. Параметры a, b, c – действительные числа. 1/(a\*x) -b при х + а < 0 и с = 0 (x-a)/sin(x) при х + a > 0 и с ≠ 0 (10\*x)/(c-6) в остальных случаях . Код на Си** | #include <iostream>  using namespace std;  int main() {  double a, b, c, x\_start, x\_end, dx;  cout << "Enter a: "; cin >> a;  cout << "Enter b: "; cin >> b;  cout << "Enter c: "; cin >> c;  cout << "Enter X started: "; cin >> x\_start;  cout << "Enter X ended: "; cin >> x\_end;  cout << "Enter step dx: "; cin >> dx;  int a\_int = a;  int b\_int = b;  int c\_int = c;  int temp1 = a\_int & b\_int;  int temp2 = a\_int & c\_int;  int final\_temp = temp1 | temp2;  bool use\_real\_number = (final\_temp != 0);  cout << "\nTable of results:" << endl;  cout << "X\tF\tType" << endl;  cout << "-------------------" << endl;  for (double x = x\_start; x <= x\_end; x += dx) {  double result;  if (c < 0 && b != 0) {  result = a\*x\*x + b\*b\*x + c;  }  else if (c > 0 && b == 0) {  result = x + a;  }  else {  result = x + c;  }  cout << x << "\t";  if (use\_real\_number) {  cout << result << "\tvalid" << endl;  } else {  cout << (int)result << "\twhole" << endl;  }  }  return 0;  } | #include <iostream>  #include <cmath>  #include <iomanip>  using namespace std;  int main() {  double a, b, c, x\_start, x\_end, dx;  cout << "Введите a: ";  cin >> a;  cout << "Введите b: ";  cin >> b;  cout << "Введите c: ";  cin >> c;  cout << "Введите Xнач: ";  cin >> x\_start;  cout << "Введите Xкон: ";  cin >> x\_end;  cout << "Введите шаг dX: ";  cin >> dx;  int a\_int = static\_cast<int>(a);  int b\_int = static\_cast<int>(b);  int c\_int = static\_cast<int>(c);  bool use\_real = ((a\_int & b\_int) | (a\_int & c\_int)) != 0;  cout << "\nРезультаты:\n";  cout << "-------------------------------------------------\n";  cout << "| X | F | Тип |\n";  cout << "-------------------------------------------------\n";  for (double x = x\_start; x <= x\_end + 1e-9; x += dx) {  double result;  if (c < 0 && b != 0) {  result = a \* x \* x + b \* b \* x + c;  } else if (c > 0 && b == 0) {  result = x + a;  } else {  result = x + c;  }  cout << "| " << setw(9) << fixed << setprecision(3) << x << " | ";  if (use\_real) {  cout << setw(14) << fixed << setprecision(6) << result << " | действительное |";  } else {  cout << setw(14) << fixed << setprecision(0) << static\_cast<int>(result) << " | целое |";  }  cout << endl;  }  cout << "-------------------------------------------------\n";  return 0;  } | 1. Форматирование вывода:  Первая программа использует iomanip для красивого табличного вывода с выравниванием  Вторая программа использует простые табы \t, что может привести к неровному выводу  2. Точность вычислений:  Первая: Учитывает погрешность чисел с плавающей точкой (+ 1e-9 в условии цикла)  Вторая: Может пропустить последнее значение из-за погрешности  3. Безопасность преобразований:  Первая: Использует static\_cast<int> - современный и безопасный подход  Вторая: Использует C-style приведение (int) - менее безопасно  4. Процесс вычислений:  Первая: Вычисляет условие use\_real более читаемо  Вторая: Разбивает вычисление на промежуточные переменные (избыточно)  5. Интерфейс:  Первая: Русскоязычный интерфейс, профессиональное оформление таблицы  Вторая: Англоязычный интерфейс, минималистичное оформление. |

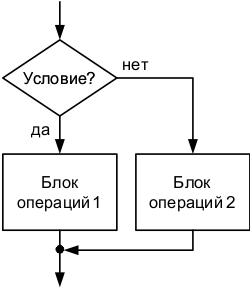


Схема 1. Блок схема для операторов условия на С++

**Вывод:**

В ходе работы была разработана программа на C++, вычисляющая значение функции F(x) по заданным условиям. Освоены практические навыки работы с условными операторами, циклами и преобразованием типов данных. Реализовано определение типа вывода результатов (целый/действительный) с помощью битовых операций. Программа корректно обрабатывает различные входные данные и выводит результаты в удобном табличном формате.