МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

## «Московский технический университет связи и информатики»

Кафедра «Информатика»

Лабораторная работа №7 - Проект 2

**«Разработка проекта, решающего задачу**

**вычисления сложного выражения с условием и выбора**

**наименьшего (наибольшего) из нескольких значений»**

по теме

**«Логические данные и выражения.**

**Операторы разветвления VС++. Программная реализация**

**базовых разветвляющиеся структур и**

**типовых алгоритмов»**

**по дисциплине**

**«Информатика»**

Выполнил: студент гр. БИК2107 Осетрова К.И.

Вариант №19

Проверил: доц. Кафедры

«Информатика» Гуриков С.Р.

Москва, 2021

Оглавление

[Общее задание 3](#_Toc30558)

[Формализация и уточнение задачи 3](#_Toc1654)

[Разработка двух функциональных алгоритмов и алгоритма главной функции main. 3](#_Toc21519)

[Разработка программного кода проекта 6](#_Toc11347)

[Результаты выполнения проекта 8](#_Toc2893)

[Доказательство правильности результата 8](#_Toc30271)

# Общее задание

Создать приложение (решение), состоящее из трех проектов в соответствии с общим заданием для вычисления условной функции r=f(x, z):

# Формализация и уточнение задачи

Алгоритм решения данной задачи представляет собой комбинацию вычисления сложного выражения с условием и выбора наименьшего (наибольшего) из нескольких значений, используя все виды разветвлений. Можно решить эту задачу двумя способами:

• создать функциональный алгоритм и соответствующую программную функцию, используя вложенные разветвления с базовыми алгоритмами нахождения наибольшего и наименьшего значений без использования библиотечных функций max и min;

• создать функциональный алгоритм и соответствующую программную функцию, используя вложенные разветвления и разработанные свои алгоритмы и программные функции нахождения наибольшего и наименьшего из двух значений.

Будем считать, что исходные данные и результат вычислений имеют тип double. Для контроля правильности результата создадим переменную целого типа, этой переменной будем присваивать номер ветви разветвления, по которой выполнились вычисления.

# Разработка двух функциональных алгоритмов и алгоритма главной функции main.

**Разработаны два функциональных алгоритма Razv и Razm:**

• функциональный алгоритм **Razv** – вычисления условного выражения решения задачи, используя вложенные разветвления с базовыми алгоритмами нахождения наибольшего и наименьшего значений без использования библиотечных функций (рисунок 1);

• функциональный алгоритм **Razm** – вычисления условного выражения

решения задачи, используя предопределенные алгоритмы вычисления мак-

симума и минимума двух переменных – **MaxMy** и **MinMy** (рисунок 2);

• схема главной функции **main** представлена на рисунке 3.

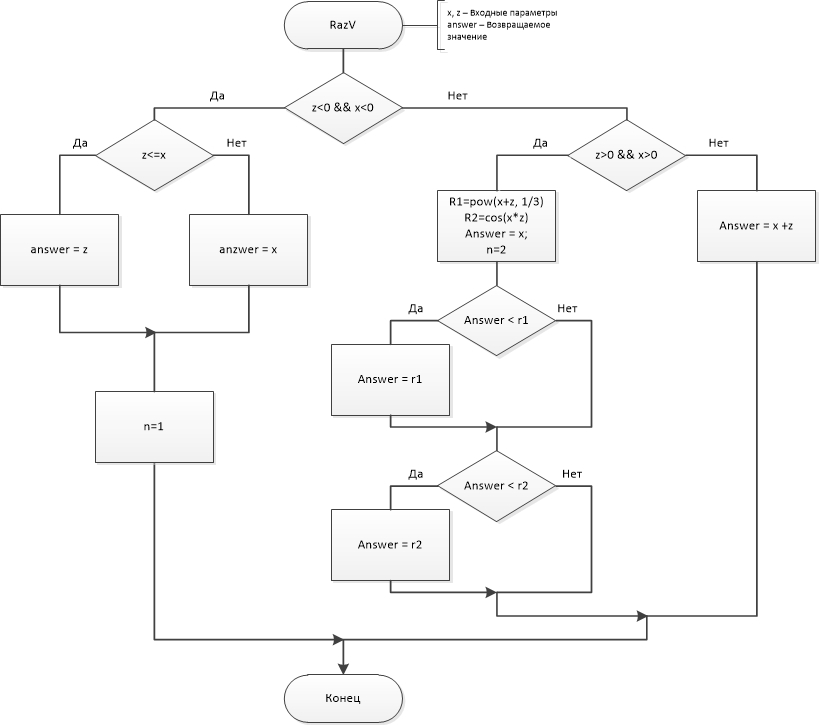


Рисунок 1 - Схема алгоритма **Razv**, использующая базовые алгоритмы нахождения наибольшего и наименьшего значения

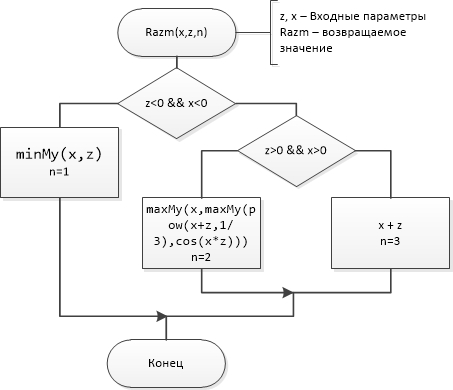


Рисунок 2 – Схема алгоритма Razm, использующая предопределенные алгоритмы maxMy и minMy

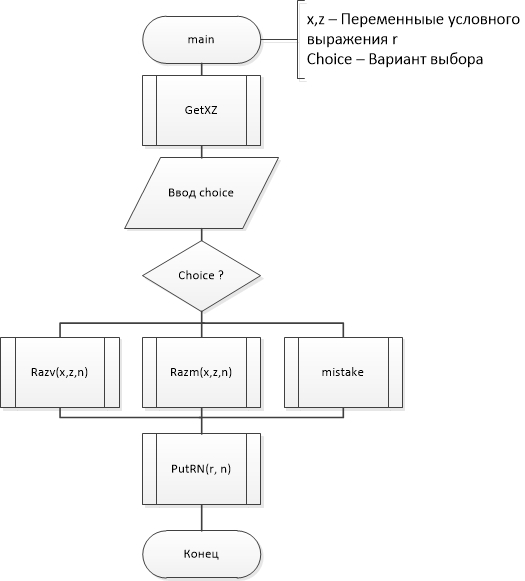


Рисунок 3 – Схема алгоритма главной функции main

# Разработка программного кода проекта

Разработан программный код проекта, состоящий из трех файлов исходного кода:

• файла с функциями **Razv**, **Razm**, **maxMy** и **minMy**, которые решают задачу вычисления условного выражения первым и вторым способами

• файла с функциями ввода исходных данных и вывода результатов;

• файла с главной функцией **main**.

Программный код проекта представлен на рисунке 4.

|  |
| --- |
| // Файл Raz с функциями Razv, Razm, maxMy и minMy  #include <cmath>  // Определение ф-ции с вложенными разветвлениями,без дополнительных функций  double Razv(double x, double z, int& n)  {  double answer; // Локальная переменная ФУНКЦИИ  if (z < 0 && x < 0)  {  if (z <= x)  answer = z;  else  answer = x;  n = 1; // Номер ветки  }  else  {  if (z > 0 && x > 0)  {  double r1 = pow(x + z, 1 / 3), r2 = cos(x \* z);  answer = x;  if (answer < r1)  answer = r1;  if (answer < r2)  answer = r2;  n = 2;  }  else  {  answer = x + z;  n = 3;  }  }  return (answer);  }  double minMy(double x, double z);  double maxMy(double x, double z);  double Razm(double x, double z, int& n)  {  if (z < 0 && x < 0)  {  n = 1;  return (minMy(x, z));  }  else  {  if (z > 0 && x > 0)  {  n = 2;  return (maxMy(x, maxMy(pow(x + z, 1 / 3), cos(x \* z))));  }  else  {  n = 3;  return (x + z);  }  }    }  double minMy(double x, double z)  {  double f;  if (x < z) f = x; else f = z;  return f;  }  double maxMy(double x, double z)  {  double f;  if (x > z) f = x; else f = z;  return f;  }  // Файл GetPut.cpp с функциями ввода и вывода  #include <iostream>  using namespace std;  // Определение функции ввода  void GetXZ(double& x, double& z)  {  setlocale(LC\_ALL, "rus");  cout << " Введите x, z\n ";  cin >> x >> z;  }  // Определение функции вывода  void PutRN(double R, int N)  {  setlocale(LC\_ALL, "rus");  cout << " Ответ R= " << R << endl;  cout << " Номер ветки разветвления " << N << endl;  }  // Файл main1.cpp  #include <iostream>  using namespace std;  void GetXZ(double& x, double& z);  void PutRN(double, int);  double Razv(double x, double z, int& n);  double Razm(double x, double z, int& n);  int main()  {  double x, z, r;  int n; // Номер ветки  GetXZ(x, z); // Вызов функции ввода исходных данных  int choice; // Вариант выбора решения  cout << " Каким способом решать задачу?\n";  cout << " 1 - с вложенными разветвлениями ";  cout << " без дополнительных функций max и min \n ";  cout << "2 - со своими функциями minMy и maxMy \n ";  cout << " Что выбираете 1 или 2 ?\n ";  cin >> choice;  switch (choice)  {  case 1:  r = Razv(x, z, n);  break;  case 2:  r = Razm(x, z, n);  break;  default:  cout << " Вы ввели что-то не то! ";  cout << endl;  system("PAUSE");  return 0;  }  PutRN(r, n); // Вызов функции вывода результатов  system("PAUSE");  return 0;  } |

Рисунок 4 – Программный код проекта

# Результаты выполнения проекта

Результаты выполнения функций Razv и Razm при заданных значениях исходных данных должны быть одинаковы. На рисунке 5 приведены результаты работы проекта для тестовых данных второй ветви.

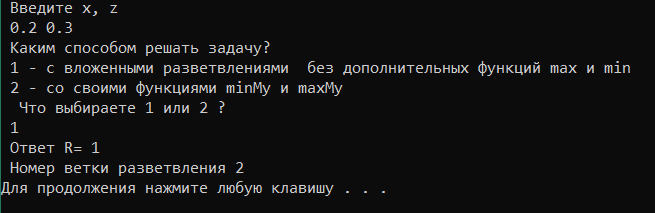


Рисунок 5 – Результаты выполнения проекта для тестовых данных

второй ветви

# Доказательство правильности результата

Для тестовых исходных данных, проверяющих все ветви разветвления,

были получены следующие результаты выполнения проекта:

|  |  |  |  |
| --- | --- | --- | --- |
| Исходные данные | | Результат | № ветви |
| x = -0.2 | z = -0.3 | 0,999999 | N = 2 |
| x = 1 | z = -5 | -4 | N = 3 |
| x = 2 | z = 6 | 2 | N = 1 |

Доказательство правильности результатов по ветвям разветвления:

Для первой строки таблицы x<0 и z<0, тогда расчёты идут по третьей ветви и находят )= max(-0.2,0.39,0.99999) - верно.

Для второй строки x>0, а z<0, следовательно расчёт идёт по третьей ветви алгоритма: x+z=1 + (-5) = -4 -верно.

Для третьей строки x>0 и z>0, тогда расчёт идёт по первой ветви алгоритма и находится min(x,z)=min(2,6)=2 - верно.

При тестовых данных результаты ручного расчёта и вычисления на компьютере совпадают.