Учреждение образования

Белорусский государственный университет информатики и радиоэлектроники

Кафедра интеллектуальных информационных технологий

Индивидуальное практическое задание № 2

по дисциплине «Метрология, стандартизация и сертификация  
в информационных технологиях»

Вариант № 8

Выполнил студент: Рублевская Е.А.

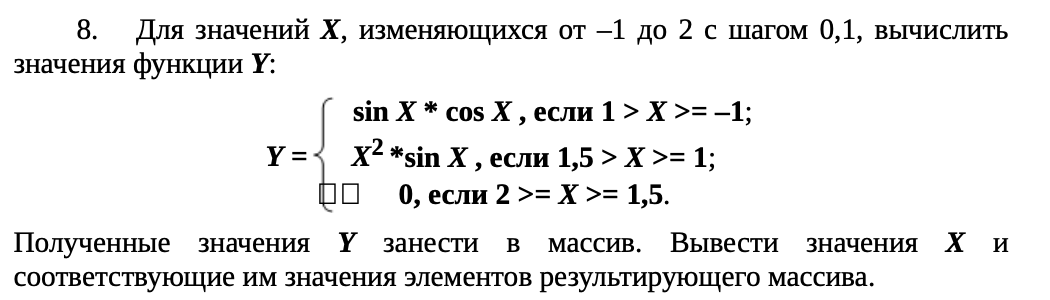
группа 321702

Зачетная книжка № 32170067

Минск 2024

**Цель:** написать текст программы на языке программирования, заданном преподавателем. По тексту программы рассчитать метрики Холстеда и метрики сложности потока данных (спен и метрику Чепина).

**Условие:**



**Исходный код программы:**

**#include <iostream>**

**#include <cmath>**

**#include <vector>**

**#include <iomanip>**

**using namespace std;**

**int main() {**

**// Диапазон значений X**

**double start = -1.0;**

**double end = 2.0;**

**double step = 0.1;**

**// Массивы для хранения значений X и Y**

**vector<double> x\_values;**

**vector<double> y\_values;**

**// Вычисление значений Y для каждого X**

**for (double x = start; x <= end; x += step) {**

**double y = 0.0;**

**if (x >= -1 && x < 1) {**

**y = sin(x) \* cos(x);**

**} else if (x >= 1 && x < 1.5) {**

**y = pow(x, 2) \* sin(x);**

**} else if (x >= 1.5 && x <= 2) {**

**y = 0;**

**}**

**// Сохранение значений в массивы**

**x\_values.push\_back(x);**

**y\_values.push\_back(y);**

**}**

**// Вывод значений X и Y**

**cout << fixed << setprecision(2);**

**cout << "X\t\t\tY\n";**

**for (size\_t i = 0; i < x\_values.size(); ++i) {**

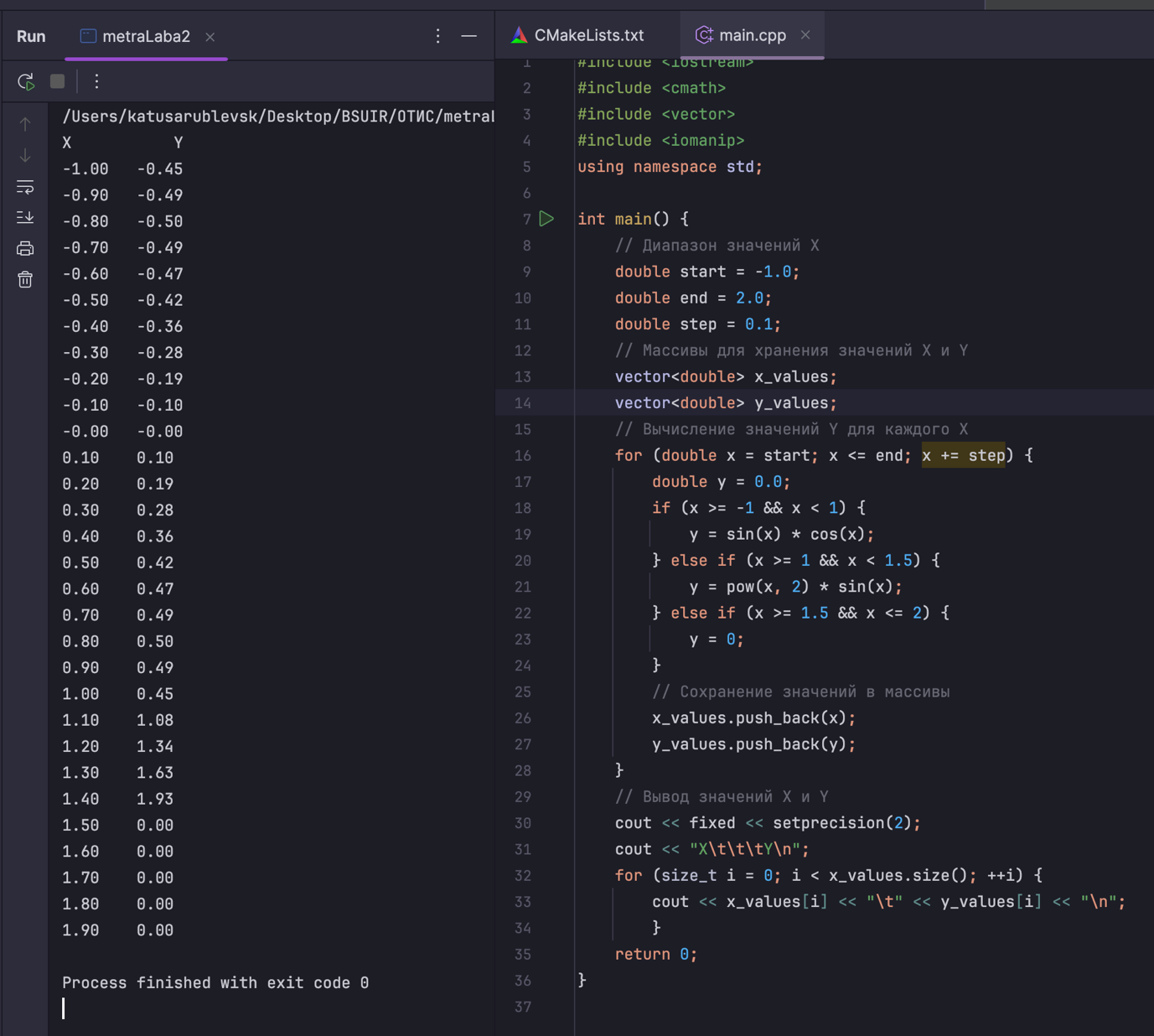
**cout << x\_values[i] << "\t" << y\_values[i] << "\n";**

**}**

**return 0;**

**}**

**Описание программы:**

****

**Назначение программы**

Программа предназначена для вычисления значений функции Y в зависимости от значения переменной X, изменяющейся от -1 до 2 с шагом 0.1, в соответствии с заданными условиями:

1. Если -1 <= X < 1 , то Y = sin(X)\*cos(X).

2. Если 1 <= X < 1.5 , то Y = X^2\*sin(X) .

3. Если 1.5<= X<= 2, то Y = 0.

Результаты вычислений (значения X и соответствующие значения Y) заносятся в массивы и выводятся на экран в виде таблицы.

Программа позволяет проанализировать поведение функции Y на заданном интервале значений переменной X и проверить, как разные условия влияют на результат.

**Описание переменных**

1. *double start* - начальное значение переменной X (-1.0). Ипользуется для задания нижней границы диапазона, по которому изменяется X.

2. *double end* - конечное значение переменной X (2.0). Используется для задания верхней границы диапазона, по которому изменяется X.

3. *double step* - шаг, с которым изменяется переменная X (0.1). Определяет, на сколько будет увеличиваться X на каждом шаге цикла.

4. *vector<double> x\_values* - массив для хранения всех значений X, которые программа генерирует в цикле. Позволяет сохранить последовательность значений переменной X для дальнейшего вывода.

5. *vector<double> y\_values* - массив для хранения всех вычисленных значений функции Y, соответствующих значениям X. Позволяет сохранить результаты вычислений для каждого X.

6. *double x* - текущая переменная цикла, представляющая текущее значение X, которое используется для вычисления Y.

7. *double y* - временная переменная для хранения текущего значения функции Y, вычисленного для конкретного значения X.

8. *size\_t i* - индексная переменная, используемая в цикле при выводе значений из массивов *x\_values* и *y\_values*. Обеспечивает последовательный доступ к элементам массивов.

**Этапы работы программы**

***1. Инициализация переменных:***

* Устанавливаются начальное значение X (*start* = -1.0), конечное значение X (*end* = 2.0) и шаг изменения X (*step* = 0.1).
* Создаются два массива (*vector*<*double*>): *x\_values* для хранения значений X и *y\_values* для хранения соответствующих значений Y.

***2. Основной цикл вычислений:***

* Выполняется цикл *for*, в котором переменная X изменяется от *start* до *end* с шагом *step*.
* На каждом шаге цикла:
* Проверяется, в какой диапазон попадает текущее значение X, и на основе этого вычисляется Y по следующим условиям:
* Если -1 <= X < 1 , то Y = sin(X)\*cos(X).
* Если 1 <= X < 1.5 , то Y = X^2\*sin(X) .
* Если 1.5 <= X<= 2, то Y = 0.
* Вычисленные значения X и Y добавляются в массивы *x\_values* и *y\_values*.

***3. Сохранение результатов:***

* По завершении цикла все значения X и соответствующие Y хранятся в массивах *x\_values* и *y\_values*.

***4. Вывод результатов:***

- На экран выводится таблица значений X и Y. Формат вывода:

* Первой строкой выводятся названия столбцов: X и Y.
* Для каждого значения X и соответствующего Y выводится строка с двумя числами, округленными до двух знаков после запятой.

***5. Завершение работы программы:***

- После вывода всех значений программа завершается. Пользователь может проанализировать результаты вычислений.

**Расчёт метрики Холстеда:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| j | Оператор | F1j | i | Операнд | F2i |
| 1 | **sin()** | 2 | 1 | **start** | 1 |
| 2 | **=** | 7 | 2 | **end** | 1 |
| 3 | **{}** | 6 | 3 | **step** | 2 |
| 4 | **&&** | 3 | 4 | **x** | 7 |
| 5 | **<<** | 4 | 5 | **у** | 5 |
| 6 | **else if** | 2 | 6 | **x\_values** | 3 |
| 7 | **for** | 1 | 7 | **y\_values** | 3 |
| 8 | **if** | 3 | 8 | **i** | 1 |
| 9 | **cos()** | 1 | 9 | **-1.0** | 1 |
| 10 | **+=** | 1 | 10 | **2.0** | 1 |
| 11 | **cout** | 2 | 11 | **0.1** | 1 |
| 12 | **pow()** | 1 | 12 | **0.0** | 1 |
| 13 | **push\_back()** | 2 | 13 | **1** | 4 |
| 14 | **return** | 1 | 14 | **1.5** | 1 |
| 15 | **;** | 16 | 15 | **2** | 3 |
| 16 | **setprecision()** | 1 | 16 |  |  |
| 17 | **fixed** | 1 | 17 |  |  |
| η1​=17 |  | N1​=54 | η2​=15 |  | N2​=35 |

Словарь программы: **η**= **15 + 17 = 32**.

Длина программы: ***N*** **= 54 + 35 = 89.**

Объем программы: ***V*** **= 89 log2 32 = 445**

**Расчёт спена программы:**

|  |  |
| --- | --- |
| Идентификатор | Спен |
| start | 2 |
| end | 2 |
| step | 1 |
| x\_values | 5 |
| y\_values | 5 |
| x | 10 |
| y | 7 |
| i | 3 |
| cout | 3 |
| Суммарный спен программы | **38** |

**Расчёт полной метрики Чепина:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Переменные | Полная метрика Чепина | | | | Метрика чепина ввода/вывода | | | |
| Группа переменных | ***P*** | ***M*** | ***C*** | ***T*** | ***P*** | ***M*** | ***C*** | ***T*** |
| Переменные относящиеся к группе | start,end,step | x\_values, y\_values | i, х | у | x\_values, y\_values | -- | i | -- |
| Количество переменных в группе | ***p = 3*** | ***m = 2*** | ***c = 2*** | ***t = 1*** | ***p = 2*** | ***m = 0*** | ***c = 1*** | ***t = 0*** |
| Метрика Чепинна | Q = P + 2⋅M + 3⋅C + 0.5⋅T  = 3 + 4 + 6 + 0.5 = 13.5 | | | | Q = P + 2⋅M + 3⋅C + 0.5⋅T  = 2 + 0 + 3 + 0 = 5 | | | |

**Вывод:** в ходе работы были рассчитаны метрики Холстеда и метрики сложности потока данных, что позволило оценить сложность программы и её структуры.