Лабораторная работа номер 4 Вариант 5

Сделал: Каравка Денис Иванович Группа: 1БПМИ-УДМО

# Постановка задачи:

Составить тестовые наборы для программ 1 и 2 лабораторных работ.

# Программа лабораторной работы 1 Формулировка задачи лабораторной работы 1:

Даны натуральное число п, действительные числа х1,…,хn. В последователь- ности х1,…,хn все неотрицательные члены, не принадлежащие отрезку [1, 2], заменить на единицу. Кроме того, получить число отрицательных членов и число членов, принадлежащих отрезку [1, 2].

**Описание тестовых наборов:** Тестовый набор номер 1: Входные данные:

|  |  |
| --- | --- |
| **Наименование**  **переменной** | **Значение переменной** |
| N | 3 |
| X1 | 1 |
| X2 | 2 |
| X3 | 4 |
| Результат: | X3 = 1;  countNegativeNums = 0;  countOneAndTwo = 2; |

Тестовый набор 2:

Входные данные:

|  |  |
| --- | --- |
| **Наименование**  **переменной** | **Значение переменной** |
| N | 0 |
| Результат: | countNegativeNums = 0;  countOneAndTwo = 0; |

# Результаты тестирования

Результаты тестирования отображаются в таблице 1. Таблица 1 – результат тестирования программы

|  |  |  |
| --- | --- | --- |
| **Номер теста** | **Тестируемая функция** | **Результат (успех/выявлен- ный дефект)** |
| Т1 | определить число отрицательных чле- нов, числа при действительных числах 1,2,4 | Успех |
| Т2 | определить число членов, принадлежа- щих отрезку [1,2] при отсутствии дей- ствительных чисел | Дефект. Нельзя посчитать кол-во чисел, которые при- надлежат отрезку, тк они от-  сутствуют |

# Текст программы

1. #include <iostream>
2. using namespace std;

3 /\*

Даны натуральное число п, действительные числах1,…,хn. В последовательности х1,…,хnвсе неотрицательные члены, не принадлежащие отрезку [1, 2], заменить на единицу. Кроме того, получить число отрицательных членов и число членов, принадлежащих отрезку [1, 2].

8 \*/

9 int main()

|  |  |  |
| --- | --- | --- |
| 10 | { |  |
| 11 |  | int n = 0; |
| 12 |  | int countNegativeNums = 0; |
| 13 |  | int countOneAndTwo = 0; |
| 14 |  | cout << "Введите число n: "; |
| 15 |  | cin >> n; |
| 16 |  | for(int i = 1; i <= n; i++){ |
| 17 |  | float Xi = 0; |
| 18 |  | cout << "Введите " << i << " число: "; |
| 19 |  | cin >> Xi; |
| 20 |  | if(Xi < 0){ |
| 21 |  | countNegativeNums ++; |
| 22 |  | } |
| 23 |  | if(Xi >= 1 && Xi <= 2){ |
| 24 |  | countOneAndTwo ++; |
| 25 |  | } |
| 26 |  | else if(Xi > 0){ |
| 27 |  | Xi = 1; |
| 28 |  | cout << "Число изменилось и стало 1\n"; |
| 29 |  | } |
| 30 |  | } |
| 31 |  | cout << "Число отрицательных чисел: " << countNegativeNums << "\n" << "Число |
|  |  | чисел, принадлежащих [1,2]: " << countOneAndTwo; |
| 33 | } |  |

# Построение таблицы трассировки для представленной программы и сле- дующего набора данных

Входные данные:

|  |  |
| --- | --- |
| **Наименование**  **переменной** | **Значение переменной** |
| N | 3 |
| X1 | 1 |
| X2 | 2 |
| X3 | 4 |
| Результат: | X3 = 1;  countNegativeNums = 0;  countOneAndTwo = 2; |

Таблица 2 – трассировка программы с тестовым примером

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Строк а** | **Ход выполне- ния** | **I** | **N** | **X1** | **X2** | **X3** | **countNegativeNums** | **countOneAndTwo** |
| 9 | int main() | ? | ? | ? | ? | ? | ? | ? |
| 10 |  |  | 0 |  |  |  |  |  |
| 11 |  |  |  |  |  |  | 0 |  |
| 12 |  |  |  |  |  |  |  | 0 |
| 14 | Введите число N |  | 3 |  |  |  |  |  |
| 16 | (I <= N) = true | 1 |  |  |  |  |  |  |
| 17 |  |  |  | 0 |  |  |  |  |
| 18 | Введите X1 число |  |  | 1 |  |  |  |  |
| 20 | (X1 < 0) = false |  |  |  |  |  |  |  |
| 23 | (X1 >= 1 && X1  <= 2) = true |  |  |  |  |  |  |  |
| 24 |  |  |  |  |  |  |  | 1 |
| 16 | (I <= N) = true | 2 |  |  |  |  |  |  |
| 17 |  |  |  |  | 0 |  |  |  |
| 18 | Введите X2 число |  |  |  | 2 |  |  |  |
| 20 | (X2 < 0) = false |  |  |  |  |  |  |  |
| 23 | (X2 >= 1 && X2  <= 2) = true |  |  |  |  |  |  |  |
| 24 |  |  |  |  |  |  |  | 2 |
| 16 | (I <= N) = true | 3 |  |  |  |  |  |  |
| 17 |  |  |  |  |  | 0 |  |  |
| 18 | Введите X3 число |  |  |  |  | 4 |  |  |
| 20 | (X3 < 0) = false |  |  |  |  |  |  |  |
| 23 | (X3 >= 1 &&  X3 <= 2) = false |  |  |  |  |  |  |  |
| 26 | (X3 > 0) = true |  |  |  |  | 1 |  |  |
| 28 | Вывод 1 |  |  |  |  |  |  |  |
| 29 |  |  |  |  |  |  |  |  |
| 16 | (I <= N) = false | 4 |  |  |  |  |  |  |
| 31 | Вывод 0 и 2 |  |  |  |  |  | **0** | **2** |
| 33 | result 0; |  |  |  |  |  |  |  |

**Программа лабораторной работы 2 Формулировка задачи лабораторной работы 2:** Var x:array [1..9999] of real; s:real;

Вычислить (индекс 1-ого слагаемого каждой суммы-квадрат): s=(x1+x2+x3) (x4+x5+…+x8) (x9+ … +x15)…(x9801+…+x9999).

# Описание тестовых наборов:

Тестовый набор номер 1:

Входные данные:

|  |  |
| --- | --- |
| **Наименование**  **переменной** | **Значение переменной** |
| size | 16 |
| X1 | 1 |
| X2 | 1 |
| … | 1 |
| X16 | 1 |
| Результат: | result = 105; |

Тестовый набор 2:

Входные данные:

|  |  |
| --- | --- |
| **Наименование**  **переменной** | **Значение переменной** |
| size | -10 |
| Результат: | 0 |

# Результаты тестирования

Результаты тестирования отображаются в таблице 1. Таблица 1 – результат тестирования программы

|  |  |  |
| --- | --- | --- |
| **Номер теста** | **Тестируемая функция** | **Результат (успех/выявлен- ный дефект)** |
| Т1 | вычислить сумму-квадрат для массива, состоящего из 16 элементов, каждый эле- мент которого равен единице | Успех |
| Т2 | вычислить сумму-квадрат для массива, состоящего из -10 элементов, каждый элемент которого равен единице. | Дефект (размер массива дол- жен быть положителен и больше нуля) |