**Постановка задачи**

Разработать детальные требования, тест план и программу для следующей задачи:

В массиве целых чисел определить наибольшую длину монотонно убывающего фрагмента  
последовательности (то есть такого фрагмента, где каждый элемент меньше предыдущего). Напишите  
функцию main, в которой примените вашу функцию для нескольких массивов:  
• Для массива, количество и значения элементов которого заданы при инициализации  
(константы).  
• Для массива, размещённого в динамической памяти, количество элементов которого должно  
быть введено, а значения элементов сгенерированы случайно.  
• Для нескольких массивов, размещённых в динамической памяти, значения которых считаны  
из файла. Файл должен содержать несколько строк, первая строка из которых содержит  
количество элементов массива, следующие - сами эти элементы.

**Задача.**

1. В массиве целых чисел определить наибольшую длину монотонно убывающего фрагмента  
   последовательности (то есть такого фрагмента, где каждый элемент меньше предыдущего). Напишите  
   функцию main, в которой примените вашу функцию для нескольких массивов:  
   • Для массива, количество и значения элементов которого заданы при инициализации  
   (константы).  
   • Для массива, размещённого в динамической памяти, количество элементов которого должно  
   быть введено, а значения элементов сгенерированы случайно.  
   • Для нескольких массивов, размещённых в динамической памяти, значения которых считаны  
   из файла. Файл должен содержать несколько строк, первая строка из которых содержит  
   количество элементов массива, следующие - сами эти элементы.
2. **Пример:**

Дан массив целых чисел: -1 0 23 34 1 54 -1 -23 50. Минимальное число: -23. Тогда положительные элементы, стоящие перед минимальным элементом: 23 34 1 54, следовательно их количество: 4.

1. **Детальные требования**
   1. Вводимые данные должны быть заданы корректно.
      1. Вводимые данные – числа.
         1. Если хотя бы одно из вводимых данных не число, то вывести сообщение «Either a non-integer positive number was entered, or a non-number was entered, end of the program!», завершив программу
         2. Если хотя бы одно из данных из файла не число, или не целое число, или неположительное, то вывести сообщение «An error occurred while reading from the file. Either the file was not found, or the sequence number not a positive integer!», завершив программу
      2. Вводимые данные – целые числа.
         1. Если хотя бы одно из вводимых данных не целое число или неположительное, то вывести сообщение «Either a non-integer positive number was entered, or a non-number was entered, end of the program!», завершив программу
   2. Программа должна выдавать требуемое значение, если введенные данные корректны и расчет величин возможен.
      1. Введенные данные корректны и расчет величин возможен.
         1. Если введенные данные корректны и расчет величин возможен, то вывести сообщение «Longest monotonically decreasing fragment length the sequence from the initialized array is [countMax]», где countMax – наибольшая длина монотонно убывающего фрагмента.

**Тест-план**

|  |  |  |  |
| --- | --- | --- | --- |
| Требование | Детальные требования | Данные | Ожидаемый результат |
| 3.1. Вводимые данные должны быть заданы корректно. | | | |
| 3.1.1 Вводимые данные – числа | 3.1.1.1 Если хотя бы одно из вводимых данных не число, то вывести сообщение «Either a non-integer positive number was entered, or a non-number was entered, end of the program!», завершив программу | x | Сообщение: « Either a non-integer positive number was entered, or a non-number was entered, end of the program!» |
| * + - 1. Если хотя бы одно из данных из файла не число, или не целое число, или неположительное, то вывести сообщение «An error occurred while reading from the file. Either the file was not found, or the sequence number not a positive integer!», завершив программу | Сообщение: «An error occurred while reading from the file. Either the file was not found, or the sequence number not a positive integer!», |
| 3.1.2 Вводимые данные – целые числа | 3.1.2.1 Если хотя бы одно из вводимых данных не целое число или неположительное, то вывести сообщение «Either a non-integer positive number was entered, or a non-number was entered, end of the program!», завершив программу | 2,5 | Сообщение: « Either a non-integer positive number was entered, or a non-number was entered, end of the program!», » |
| 3.2. Программа должна выдавать требуемое значение, если введенные данные корректны и расчет величин возможен | | | |
| 3.2.1 Введенные данные корректны и расчет величин возможен | 3.2.1.1 Если введенные данные корректны и расчет величин возможен, то вывести сообщение «Longest monotonically decreasing fragment length the sequence from the initialized array is [countMax]», где countMax – наибольшая длина монотонно убывающего фрагмента. | 5 | Сообщение:  «Longest monotonically decreasing fragment length the sequence from the initialized array is [countMax]», где countMax – наибольшая длина монотонно убывающего фрагмента. |

**Код:**

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <limits>

#include <stdlib.h>

#include <time.h>

#include <algorithm>

const char\* ERROR\_INVALID\_INPUT = "Either a non-integer positive number was entered, or a non-number was entered, end of the program!\a"; \

const char\* ERROR\_INVALID\_FILE\_NUMBER = "An error occurred while reading from the file. Either the file was not found, or the sequence number not a positive integer!\a";

int MyFunc(int array[], int count)

{

int countThree = 1, countMax = 0;

for (int i = 1; i < count; i++)

{

if (array[i - 1] > array[i])

{

countThree++;

}

else

{

countMax = std::max(countMax, countThree);

countThree = 1;

}

countMax = std::max(countMax, countThree);

}

return countMax;

}

int main()

{

const int size = 5;

int arrayOne[size] = { 1, 6845, 654, 888, 789 };

int countOne = 1, countMaxOne = 0;

countMaxOne = MyFunc(arrayOne, size);

std::cout << "Longest monotonically decreasing fragment length the sequence from the initialized array is " << countMaxOne << std::endl << std::endl;

int arrsize = 0;

try

{

std::cout << "Enter the number of elements in a dynamic array: ";

std::cin >> arrsize;

srand(time(0));

if ((!std::cin) or std::cin.fail() or (std::cin.peek() != 10))

{

throw ERROR\_INVALID\_INPUT;

}

int\* arrayTwo = new int[arrsize];

std::cout << "Values in a dynamic array: " << std::endl;

for (int i = 0; i < arrsize; i++)

{

arrayTwo[i] = rand() % 30001 - 15000;

std::cout << arrayTwo[i] << std::endl;

}

int countTwo = 1, countMaxTwo = 0;

countMaxTwo = MyFunc(arrayTwo, arrsize);

if (countMaxTwo != 1)

{

std::cout << "Longest monotonically decreasing fragment length a sequence from a dynamic array is " << countMaxTwo << std::endl << std::endl;;

}

else

{

std::cout << "There is no monotonically decreasing fragment in the sequence!" << std::endl;

}

delete[] arrayTwo;

}

catch (const char\* error)

{

std::cerr << error;

return -1;

}

int arrsizes = 0, countThree = 0, countMaxThree = 0, k = 0;;

std::ifstream input("C:/Users/PetrK/Documents/Visual Studio Files/ConsoleApplication12/input.txt");

try

{

while (!input.eof())

{

countThree = 1;

arrsizes = 0;

countMaxThree = 1;

input >> arrsizes;

k++;

if (arrsizes == 0)

{

return 0;

}

if ((!input) or input.fail() or (input.peek() != 10))

{

throw ERROR\_INVALID\_FILE\_NUMBER;

}

int\* arrayThree = new int[arrsizes];

for (int i = 0; i < arrsizes; i++)

{

input >> arrayThree[i];

if ((!input) or input.fail() or (input.peek() != 10))

{

delete[] arrayThree;

throw ERROR\_INVALID\_FILE\_NUMBER;

}

}

countMaxThree = MyFunc(arrayThree, arrsizes);

std::cout << k << " array : ";

if (countMaxThree > 1)

{

std::cout << "The longest length of a monotonically decreasing fragment from a file is " << countMaxThree << std::endl;

}

else

{

std::cout << "There is no monotonically decreasing fragment in the sequence from file!" << std::endl;

}

delete[] arrayThree;

}

input.close();

}

catch (const char\* error)

{

std::cerr << error;

return -1;

}

return 0;

}