**3530904/00002 - Марфина Валерия Олеговна - Вариант 13**

**Задание.**

Напишите функцию, выполняющую действие в соответствии с вашим вариантом. 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.

**Индивидуальное задание №13.**

В массиве целых чисел определить наибольшую длину монотонного убывающего фрагмента последовательности (то есть такого фрагмента, где каждый элемент меньше предыдущего)

**Примечание.**

* введите число элементов для массива в динамической памяти и выделите память для массива, предусмотрите обработку исключений, если количество элементов задано некорректно и если не память выделена;
* Напишите функцию, выполняющую действие в соответствии с вашим вариантом. Напишите функцию main, в которой примените вашу функцию для двух массивов, размещённых в статической и динамической памяти. Количество элементов динамического массива ввести с клавиатуры, значения элементов динамического массива сгенерировать случайно, статического – задать в тексте функции main при инициализации.

**Входные данные.**

Пользователь вводит количество элементов для динамического массива.

Программе подается файл “mass.cpp”, содержащий данные для считывания и решения поставленной задачи.

**Выходные данные.**

**Спецификации.**

**Тест план.**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Спецификация** | **Входные данные** | **Ожидаемый результат** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Листинг.**

#include <iostream>

#include <ctime>

#include <fstream>

#include <string>

using std::ifstream; //Входной файловый поток

using std::ofstream;// Выходной файловый поток

//ИНДИВИДУАЛЬНОЕ

// 13 Вариант

//В массиве целых чисел определить наибольшую длину

//монотонного убывающего фрагмента последовательности

//(то есть такого фрагмента, где каждый элемент меньше предыдущего)

void outArray(int array[], const int SIZE)

{

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

{

std::cout << array[i] << " ";

}

std::cout << std::endl;

}

void random(int array[], const int SIZE)

{

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

{

array[i] = rand() % 100 - 50;

}

}

int isMonotoneDecreasing(int array[], const int SIZE)

{

int res = 0, max = 0;

for (int i = 0; i < SIZE - 1; i++)

{

if (array[i] >= array[i + 1])

{

res++;

}

else if (res > max)

{

max = res;

res = 0;

}

}

if (res > max)

{

max = res;

}

return max;

}

int main()

{

setlocale(LC\_ALL, "Russian");

int sizeMassfile;

//установка генератора случайных чисел; функция time(0) объявлена в <ctime>

srand(time(0));

try

{

std::cout << "Введите кол-во элементов для динамического массива: " << std::endl;

int size\_array;

std::cin >> size\_array;

if (std::cin.fail())

{

throw std::exception("Некорректный ввод!");

}

else if (size\_array <= 0)

{

throw std::exception("Память не может быть веделена!");

return 1;

}

//динамический

int\* mas\_array = new int[size\_array];

//статический

const int SIZE = 10;

int array[SIZE] = { 9,8,6,5,4,3,-4,2,1,1 };

random(mas\_array, size\_array);

std::cout << "Вывод массива: " << std::endl;

outArray(mas\_array, size\_array);

std::cout << "Hаибольшую длинa монотонного убывающего фрагмента последовательности = ";

std::cout << isMonotoneDecreasing(mas\_array, size\_array) << std::endl;

delete[] mas\_array;

std::cout << "======================================================================================" << std::endl;

std::cout << "Вывод статического массива: " << std::endl;

outArray(array, SIZE);

std::cout << "Hаибольшая длинa монотонного убывающего фрагмента последовательности = ";

std::cout << isMonotoneDecreasing(array, SIZE) << std::endl;

std::cout << "======================================================================================" << std::endl;

//массивы из файла

std::ifstream massfile;

massfile.open("mass.txt");

if (!massfile.eof())// файл не пуст

{

while (!massfile.eof())

{

massfile >> sizeMassfile;

if (sizeMassfile <= 0)

{

throw std::exception("Память не может быть веделена!");

return 1;

}

int\* arrayFromFile = new int[sizeMassfile];

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

{

massfile >> arrayFromFile[i];

}

std::cout << "Вывод массива из файла: " << std::endl;

outArray(arrayFromFile, sizeMassfile);

std::cout << "Hаибольшую длинa монотонного убывающего фрагмента последовательности = ";

std::cout << isMonotoneDecreasing(arrayFromFile, sizeMassfile) << std::endl;

delete[] arrayFromFile;

}

}

else

{

throw std::exception("Файл пуст!");

return 1;

}

massfile.close();

}

catch (const std::exception& e)

{

std::cerr << e.what() << std::endl;

return 1;

}

catch (const std::string& error)

{

std::cerr << std::endl << "Ошибка: файл " << error << std::endl;

return 1;

}

return 0;

}