Тоббо Кирилл Вадимович

Группа: 3530904/10005

Лабораторная работа №3. Индивидуальные задания.

**Задание 1.10**

**Общая постановка задачи.**

Напишите функцию, В массиве целых чисел найти число упорядоченных троек, тройка элементов упорядочена, если a[i] > a[i+1] > a[i+2]. 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.

**Детальные требования, тест план.**

1. Создаем массив с константным значением элементов. Создаем функцию (getAnswer) в функцию передается статический массив и размер. Ищем количество упорядоченных троек.
2. Создаем динамический массив, количество элементов которого вводится с клавиатуры. А значение элементов генерируется случайно.
3. Количество элементов второго массива должно быть задано корректно, если количество элементов задано не числом, выводится сообщение “The number of elements of the array must be a number. End of the program.”
4. Для заполнения второго динамического массива используем функцию srand() и rand().Создаем функцию (getAnswer) в функцию передается статический массив и размер. Ищем количество упорядоченных троек.
5. Создаем массив количество элементов, которого и сами элементы будут считываться из файла.
6. Создаем функцию(readAndWorkWithFile), в функцию в качестве аргументов передается имя файла с массивом данных. Ищем количество упорядоченных троек.

**Код LAB3.CPP:**

#include <iostream>

#include <fstream>

#include <string>

const char\* ERROR\_FILE\_NOT\_FOUND = "Error: file not found";

const char\* ERROR\_INVALID\_FILE\_INPUT = "Error: invalid file input.txt structure";

const char\* ERROR\_INVALID\_ARRAY\_INPUT = "Error: array must contain only integers";

const char\* ERROR\_INVALID\_ARRAY\_SIZE\_INPUT = "Error: array size must be integer greater then 0 and placed on new line";

void readAndWorkWithFile(std::string &filename);

int main()

{

std::string filename = R"(C:\Users\orcva\source\repos\LAB1\LAB3\input.txt)";

readAndWorkWithFile(filename);

}

void readAndWorkWithFile(std::string &filename){

try {

int n = 0;

int curPos = 3;

std::cout << "Answers: " << std::endl;

std::ifstream input(filename);

if (!input.is\_open())

{

throw ERROR\_FILE\_NOT\_FOUND;

}

else

{

while(!input.eof())

{

int curN = 0;

input >> curN;

if(curN < 1 || input.fail() || input.peek() != 10 || input.peek() == 32)

{

throw ERROR\_INVALID\_ARRAY\_SIZE\_INPUT;

}

int \*array = new int[curN];

int count = 0;

for (int j = 0; j < curN; ++j)

{

input >> array[j];

if(input.fail())

{

throw ERROR\_INVALID\_ARRAY\_INPUT;

}

}

for (int i = 2; i < curN; ++i) {

if (array[i - 2] < array[i - 1] && array[i - 1] < array[i]) {

++count;

}

}

char check = ' ';

input.get(check);

if(check != 10 && !input.eof())

{

throw ERROR\_INVALID\_FILE\_INPUT;

}

std::cout << count << std::endl;

delete [] array;

}

}

input.close();

}

catch(const char\* error)

{

system("cls");

std::cerr << std::endl << error << std::endl;

exit(-1);

}

}

**Код LAB3-2.cpp**

#include <iostream>

const int size = 6;

void getAnswer(int(&array)[size], int size);

int main()

{

int array[size] = { 1, 2, 3, 5, -1, 0 };

getAnswer(array, size);

return 0;

}

void getAnswer(int(&array)[size], int size)

{

int max = std::numeric\_limits<int>::min();

int count = 0;

for (int i = 2; i < size; ++i) {

if (array[i - 2] < array[i - 1] && array[i - 1] < array[i]) {

++count;

}

}

std::cout << "Answer: " << count << std::endl;

}

**Код LAB3-3.cpp**

#include <iostream>

const char\* ERROR\_INCORRECT\_INPUT = "Error: array size must be integer greater than 0.";

void generateArray();

int main()

{

generateArray();

}

void generateArray() {

int n = 0;

std::cout << "Enter array size:";

std::cin >> n;

if (std::cin.fail() || std::cin.peek() != 10 || n < 1)

{

std::cerr << ERROR\_INCORRECT\_INPUT << std::endl;

exit(-1);

}

srand((unsigned)time(NULL));

int\* array = new int[n];

int max = std::numeric\_limits<int>::min();

int count = 0;

std::cout << "Array:";

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

{

array[i] = rand() % 200 - 100;

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

}

std::cout << std::endl;

for (int i = 2; i < n; ++i) {

if (array[i - 2] < array[i - 1] && array[i - 1] < array[i]) {

++count;

}

}

std::cout << "Answer: " << count << std::endl;

delete[] array;

}