**Лабораторная работа № 6**

**Задание № 3**

1. **Общая постановка задачи**

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.

1. **Детальные требования, тест план**
2. **Данные должны быть введены корректно**

1. Длина динамического массива и массивов из файла – целое положительное число

Если введено не целое число, отрицательное число или ноль, сообщение: "Длина массива – положительное число!".

2. В директории проекта должен существовать файл со входными данными с корректным названием.

Если его нет, сообщение: "Файл не найден"

3. Значения массива из файла – только целые числа

Если введено не целое число, сообщение: "Введено неверное значение!"

4. Границы интервала должны быть заданы только целыми числами

Если введены не целые числа или не числа, сообщение: «Введено неверное значение!»

1. Нахождение максимального элемента, входящего в диапазон
2. Такой элемент существует

Если такой элемент найден, сообщение: «Максимум в заданном интервале: <>»

2. Такой элемент не существует

Если такой элемент не найден, сообщение: «Нет чисел из указанного диапазона!»

1. **Таблица с детальными требованиями и тест план**

|  |  |  |  |
| --- | --- | --- | --- |
| **Требование** | **Детальные требования** | **Данные** | **Ожидаемый результат** |
| 1. **Данные должны быть введены корректно** | | | |
| 1.1  Длина динамического массива и массивов из файла – целое положительное число | Если введено не целое число, отрицательное число или ноль, сообщение: "Длина массива – положительное число!". | А  9.7 | Сообщение: «Длина массива – положительное число!» |
| 1.2 В директории проекта должен существовать файл со входными данными с корректным названием. | Если его нет, сообщение: "Файл не найден" | Inyd.txt | Сообщение: «Файл не найден!» |
| 1.3 Значения массива из файла – только целые числа | Если введено не целое число, сообщение: "Введено неверное значение!" | 9.6  f | Сообщение: "Введено неверное значение!" |
| 1.4 Границы интервала должны быть заданы только целыми числами | Если введены не целые числа или не числа, сообщение: «Введено неверное значение!» | 8.6  g | Cообщение: «Введено неверное значение!» |
| 1. Нахождение максимального элемента, входящего в диапазон | | | |
| 2.1 Такой элемент существует | Если такой элемент найден, сообщение: «Максимум в заданном интервале: <>» | 1 3 4 5 6  Интервал:  0 10 | Сообщение: «Максимум в заданном интервале: 6» |
| 2.2 Такой элемент не существует | Если такой элемент не найден, сообщение: «Нет чисел из указанного диапазона!» | 1 3 4 5 6  Интервал:  -8 0 | Сообщение: «Нет чисел из указанного диапазона!» |

1. **Код программы**

#include <iostream>

#include <fstream>

int maxInInterval(int\* array, int length, int intervalLeft, int intervalRight)

{

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

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

{

if ((array[i] > max) && (array[i] >= intervalLeft) && (array[i] <= intervalRight))

{

max = array[i];

}

}

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

{

throw std::exception();

}

return max;

}

int main()

{

setlocale(LC\_ALL, "russian");

int intervalRight = 0;

int intervalLeft = 0;

std::cout << "Введите левую границу интервала, в котором будет вычисляться максимум: ";

std::cin >> intervalLeft;

if (std::cin.fail())

{

std::cerr << "Введено неверное значение!" << std::endl;

return -1;

}

std::cout << "Введите правую границу интервала, в котором будет вычисляться максимум: ";

std::cin >> intervalRight;

if (std::cin.fail())

{

std::cerr << "Введено неверное значение!" << std::endl;

return -1;

}

const int SIZE = 10; // константный массив

int arr[SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

std::cout << "Константный массив." << "\n";

try

{

std::cout << "Максимум в заданном интервале: " << maxInInterval(arr, SIZE, intervalLeft, intervalRight) << "\n";

}

catch (std::exception& err)

{

std::cerr << "Нет максимума в выбранном интервале!" << "\n";

return -1;

}

std::cout << "Динамический массив." << "\n";

int randomLength = 0; //динамический массив

std::cout << "Введите длину массива:";

std::cin >> randomLength;

if (std::cin.fail())

{

std::cerr << "Введено неверное значение!" << std::endl;

return -1;

}

if (randomLength <= 0)

{

std::cerr << "Длина массива - положительное число!" << std::endl;

return -1;

}

int\* randomArray = new int[randomLength];

std::cout << "Массив: ";

for (int i = 0; i != randomLength; ++i) // генерация элементов от -50 до 49

{

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

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

}

std::cout << std::endl;

try

{

std::cout << "Максимум в заданном интервале: " << maxInInterval(randomArray, randomLength, intervalLeft, intervalRight) << "\n";

}

catch (std::exception& err)

{

std::cerr << "Нет максимума в выбранном интервале!" << "\n";

return -1;

}

delete[] randomArray;

std::ifstream fileIn; // файловый массив

std::ofstream fileOut;

std::string inFileName = "";

std::string outFileName = "";

std::cout << "Введите имя входного файла: ";

std::cin >> inFileName;

std::cout << "Введите имя выходного файла: ";

std::cin >> outFileName;

fileOut.open(outFileName);

fileIn.open(inFileName);

if ((!fileIn) || (!fileOut))

{

std::cerr << "Файл не найден!" << std::endl;

return -1;

}

int fileLength = 0;

fileIn >> fileLength;

if (fileLength <= 0)

{

std::cerr << "Длина массива - положительное число!" << std::endl;

return -1;

}

while (!fileIn.eof())

{

int\* fileArray = new int[fileLength];

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

{

fileIn >> fileArray[i];

if (fileIn.fail())

{

std::cerr << "Введено неверное значение!" << std::endl;

return -1;

}

}

try

{

fileOut << "Максимум в заданном интервале: " << maxInInterval(fileArray, fileLength, intervalLeft, intervalRight) << "\n";

}

catch (std::exception& err)

{

std::cerr << "Нет максимума в выбранном интервале!" << "\n";

return -1;

}

fileIn >> fileLength;

if (fileLength <= 0)

{

std::cerr << "Длина массива - положительное число!" << std::endl;

return -1;

}

delete[] fileArray;

}

fileIn.close();

fileOut.close();

}