ФИО: Гольчанский Максим Дмитриевич

 Группа: 3530904/10003

Тема: Одномерные массивы( встроенные и в динамической памяти)

Дата отправки задания: 21.11.2021

**Постановка задачи**

Разработать детальные требования и тест план для следующей задачи:

**В 3-ёх видах массивов( в статическом, в динамическом и для нескольких массивов, размещенных в динамической памяти, значения которых считаны из файла), состоящих из целых чисел, определить максимальное количество подряд идущих одинаковых элементов.**

**Задача :**

1. Вводится количество элементов динамического массива, значения элементов которого генерируются случайным образом, дан одномерный массив , количество и значения элементов которого заданы при инициализации считываем файл, в котором находится строка, состоящая из целых чисел, образующих несколько массивов, размещенных в динамической памяти.

2.   Детальные требования

2.1   Число элементов динамического массива должно быть задано корректно.

2.1.1 N – целое и положительное число.

    2.1.1.1 Если N не является целым и положительным числом , выводим сообщение: «Введенное значение некорректно, введите новое»

2.2 Программа должна вывести максимальное количество подряд идущих одинаковых элементов в каждой разновидности массива.

Алгоритм подсчета  максимального количества подряд идущих одинаковых элементов в каждом подзадании одинаков:

2.2.1 Если, проходя по массиву с помощью цикла for, текущий элемент равен следующему, увеличиваем локальный счетчик  подряд идущих одинаковых элементов. Если он больше установленного максимума, то заменяем максимум на этот счетчик.

Если при проходе по массиву соседние элементы перестают быть равными, то сбрасываем локальный счетчик.

2.2.1.1 Выводим сообщение: «"The number of maximum consecutive members of a static array is equal to: "»

2.2.2 Если, проходя по массиву с помощью цикла for, текущий элемент равен следующему, увеличиваем локальный счетчик  подряд идущих одинаковых элементов. Если он больше установленного максимума, то заменяем максимум на этот счетчик.

Если при проходе по массиву соседние элементы перестают быть равными, то сбрасываем локальный счетчик.

2.2.2.1 Выводим сообщение: «"The number of maximum consecutive members of a dynamic array is equal to: "»

2.2.3 Если, проходя по массиву с помощью цикла for, текущий элемент равен следующему, увеличиваем локальный счетчик  подряд идущих одинаковых элементов. Если он больше установленного максимума, то заменяем максимум на этот счетчик.

Если при проходе по массиву соседние элементы перестают быть равными, то сбрасываем локальный счетчик.

2.2.3.1 Выводим сообщения: «">>> From File (arr #" << counter << "): "»

                             «"The maximum number of consecutive numbers: "»

3.   Тест-план

| Требование | Детальные требования | Данные | Ожидаемый результат | |
| --- | --- | --- | --- | --- |
| 2.1 Число элементов динамического массива должно быть задано корректно.   (N – целое число и положительное число) | | | | |
| 2.1.1 N – целое и положительное число. | 2.1.1.1  Если N не является целым и положительным числом , выводим сообщение: «Введенное значение некорректно, введите новое» |  | Сообщение:  «Введенное значение некорректно, введите новое». | |
| 2.2 Программа должна вывести количество чисел, входящих в монотонно-возрастающий фрагмент последовательности | | | | |
| 2.2.1 | 2.2.1.1  Если, проходя по массиву с помощью цикла for, текущий элемент равен следующему, увеличиваем локальный счетчик  подряд идущих одинаковых элементов. Если он больше установленного максимума, то заменяем максимум на этот счетчик.  Если при проходе по массиву соседние элементы перестают быть равными, то сбрасываем локальный счётчик.  Сообщение: «"Максимальное количество последовательных элементов статического массива равно: "» в любой ситуации | 4 ,5 ,3 ,2 ,2 ,2 ,2 ,4 ,8 ,1 ,9 ,3 ,2 ,3 ,0 ,0 | | Сообщение: «"Максимальное количество последовательных элементов статического массива равно: 4"» |
| 2.2.2 | 2.2.2.1  Если, проходя по массиву с помощью цикла for, текущий элемент равен следующему, увеличиваем локальный счетчик  подряд идущих одинаковых элементов. Если он больше установленного максимума, то заменяем максимум на этот счетчик.  Если при проходе по массиву соседние элементы перестают быть равными, то сбрасываем локальный счетчик  .Выводим сообщение: «"Максимальное количество последовательных элементов динамического  массива равно: "» в любой ситуации | 4 5 2 3 3 3 5 9 7 2 1 6 5 5 | | Сообщение: «"Максимальное количество последовательных элементов динамического массива равно: 3"» |
| 2.2.3 | 2.2.3.1 Если, проходя по массиву с помощью цикла for, текущий элемент равен следующему, увеличиваем локальный счетчик  подряд идущих одинаковых элементов. Если он больше установленного максимума, то заменяем максимум на этот счетчик.  Если при проходе по массиву соседние элементы перестают быть равными, то сбрасываем локальный счетчик.   Выводим сообщения:  «">>>  Array From File: "»   «"Максимальное количество последовательных чисел: "»  в любой ситуации | 7 4 5 7 13 3 3 3 4 2 8 8 9 9 2 2 2 2 15 3 2 4 1 1 7 8 9 3 2 10 2 10 4 1 1 4 6 | | Выводим сообщения:  >>> Array From File:  Максимальное количество последовательных чисел: 3  >>>  Array From File:  Максимальное количество последовательных чисел: 2  >>> Array From File:  Максимальное количество последовательных чисел: 3  >>>  Array From File:  Максимальное количество последовательных чисел: 0  >>>  Array From File:  Максимальное количество последовательных чисел: 0  >>>  Array From File:  Максимальное количество последовательных чисел: 0  >>>  Array From File:  Максимальное количество последовательных чисел: 5 |

4.Реализация:

// CountElements.h

#ifndef COUNTELEMENTS\_H

#define COUNTELEMENTS\_H

namespace CountElements {

int CountOfEqualMaximumElements(int\* new\_array, int P);

}

#endif

// CountElements.cpp

#include "CountElements.h"

int CountElements::CountOfEqualMaximumElements(int\* new\_array, int P) {

int count\_2 = 0, temp\_count\_2 = 0;

int pos\_2 = 0, temp\_pos\_2 = 0;

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

if (\*(new\_array + j) == \*(new\_array + j + 1)) {

++temp\_count\_2;

if (temp\_count\_2 > count\_2) {

count\_2 = temp\_count\_2;

pos\_2 = temp\_pos\_2;

}

}

else {

temp\_count\_2 = 1;

temp\_pos\_2 = j;

}

return count\_2;

}

// main.cpp

#include <iostream>

#include <fstream>

#include <ctime>

#include "CountElements.h"

int main() {

int count = 0, temp\_count = 0;

int pos = 0, temp\_pos = 0;

int N = 0;

int StArr[] = { 4 ,5 ,3 ,2 ,2 ,2 ,2 ,4 ,8 ,1 ,9 ,3 ,2 ,3 ,0 ,0 };

for (int j = 0; j < sizeof(StArr) / sizeof(StArr[0]); ++j)

if (StArr[j] == StArr[j + 1]) {

++temp\_count;

if (temp\_count > count) {

count = temp\_count;

pos = temp\_pos;

}

}

else {

temp\_count = 1;

temp\_pos = j;

}

std::cout << "Максимальное количество последовательных элементов статического массива равно: "

<< count << std::endl;

int n = 0;

int count\_1 = 0;

int temp\_count\_1 = 0;

int pos\_1 = 0, temp\_pos\_1 = 0;

std::cout << "Enter numbers of dynamic array: ";

std::cin >> n;

int\* arr = new int[n];

std::srand(std::time(NULL));

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

\*(arr + i) = std::rand() % 10 + 1;

std::cout << \*(arr + i) << " ";

}

std::cout << std::endl;

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

for (int j = i + 1; j < n; ++j)

if (\*(arr + j) == \*(arr + j + 1)) {

++temp\_count\_1;

if (temp\_count\_1 > count\_1) {

count\_1 = temp\_count\_1;

pos\_1 = temp\_pos\_1;

}

}

else {

temp\_count\_1 = 1;

temp\_pos\_1 = j;

}

}

std::cout << "Максимальное количество последовательных элементов динамического массива равно: "

<< count\_1 << std::endl;

delete[]arr;

int counter = 0;

int Count = 0;

int P = -1;

std::ifstream arrays\_file("C:\\Папка\\ARRAYS\_FILENAME.txt");

if (!arrays\_file) {

return false;

}

while (arrays\_file >> P) {

int\* new\_array = new int[P];

for (int i = 0; i < P; i++) {

arrays\_file >> new\_array[i];

if (\*(new\_array + i) == \*(new\_array + i + 1)) {

counter++;

}

Count++;

}

std::cout << "Array From File" << std::endl;

std::cout << "Максимальное количество последовательных чисел: " << CountElements::CountOfEqualMaximumElements(new\_array, P) << std::endl;

delete[] new\_array;

}

arrays\_file.close();

return 0;

}