

Лабораторная работа № 5

Операции с одномерными массивами: нахождение максимумов, минимумов и количества повторений.

Оглавление

Цель	1
Инструкция:	2
Задания для самостоятельного выполнения:.....	4
Домашнее задание:	4
БЛОК А	4
БЛОК В	4
БЛОК С	5
Требования к оформлению программ:.....	6
Контрольные вопросы:	7

Цель

Научиться выполнять элементарные операции с одномерными массивами: искать экстремальные значения и количество повторений.

Инструкция:

Алгоритм нахождения максимального значения среди всех элементов массива:

1. Выдвигаем предположение, что первый элемент массива – искомый максимум. Запоминаем номер максимума.
2. Сравниваем предполагаемый максимум последовательно с каждым элементом массива. Если найдем элемент больше текущего максимума, то этот элемент делаем максимальным. Запоминаем его позицию.
3. Продолжаем проверку, пока не сравним максимум со всеми элементами массива.
4. Выводим полученный результат на экран.

Если в задании необходимо найти максимум среди части элементов (например: среди отрицательных, среди четных, и др.), то:

- Необходимо найти хотя бы один элемент в массиве, который удовлетворяет всем дополнительным условиям. Если такой элемент найдет, тогда предполагаем его максимальным и сравниваем с оставшимися элементами, которые также подходят под критерии поиска. Если таких элементов нет – то выводим соответствующее сообщение на экран.

```
//-----main.cpp-----  
  
//Лабораторная работа №5  
//Поиск максимального значения в массиве  
  
#include <iostream>  
#include "vect.h"  
#include "calc_vect.h"  
#include "max_min_vect.h"  
#define n 5  
  
using namespace std;  
  
int main()  
{  
    double MAX;  
    int num=0;  
    double x=2;  
  
    double *v1 = new double[n];  
  
    InputDescVect(v1,n);  
  
    cout<<"Vector v1:\t";  
    OutputDescVect(v1,n);  
  
    GetMaxVect (v1,n,num);  
    MAX=v1[num];  
  
    cout<<"Maximum value in massive is V["<< num+1 <<"] = "<<MAX<<endl;  
    cout<<"Count of '\"<< x << "\" repeats is "<<GetNumValue (v1, n, x)<<endl;  
  
    delete []v1; v1=NULL;  
}
```

```

//-----max_min_vect.cpp-----

#include <stdio.h>
#include "vect.h"
#include "max_min_vect.h"

//функция для нахождения максимального значения и его позиции в массиве
void GetMaxVect (double* v, int n, int &i_max)
{
    int i,j;
    i_max=0;
    int max=v[i_max];
    for(i=1;i<n;i++)
    {
        if(max<v[i])
        {
            i_max=i;
            max=v[i_max];
        }
    }
}

//функция для нахождения количества повторений числа value в массиве
int GetNumValue (double* v, int n, int value)
{
    int i, res=0;
    for(i=0;i<n;i++)
        if(v[i]==value)
            res++;
    return res;
}

//-----max_min_vect.h-----

#ifndef MAX_MIN_VECT_H_INCLUDED
#define MAX_MIN_VECT_H_INCLUDED

void GetMaxVect (double* v, int n, int &i_max);
int GetNumValue (double* v, int n, int value);

#endif // MAX_MIN_VECT_H_INCLUDED

```

Задания для самостоятельного выполнения:

Написать функции и продемонстрировать их работу:

1. Нахождения максимального значения в массиве и его позиции.
2. Нахождения минимального значения в массиве и его позиции.
3. Функция, которая вычисляет, сколько раз число k встречается в массиве. Число k считывать с экрана.

Домашнее задание:

БЛОК А

Задание на отметку «удовлетворительно»

Выполнить задание и оформить код программы в соответствии с требованиями.

Задание:

Использовать «заготовку» проекта из лабораторного задания №3.

Добавить функции:

1. Для нахождения максимального отрицательного числа и его позиции в массиве. Если отрицательных чисел в массиве нет – выводить сообщение об ошибке.
2. Для нахождения минимального положительного числа и его позиции в массиве. Если положительных чисел в массиве нет – выводить сообщение об ошибке.

Продемонстрировать работу всех функций.

БЛОК В

Задания на отметку «хорошо»

1. Найти сумму элементов массива, которые находятся между максимальным и минимальным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
2. Найти сумму элементов массива, которые находятся между максимальным и минимальным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
3. Найти сумму элементов массива, которые находятся между первым и минимальным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
4. Найти сумму элементов массива, которые находятся между первым и максимальным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
5. Найти сумму элементов массива, которые находятся между последним и минимальным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
6. Найти сумму элементов массива, которые находятся между последним и максимальным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
7. Найти количество элементов массива, которые находятся между максимальным и минимальным числом. Выводить на экран элементы, удовлетворяющие условию поиска.

6. Найти сумму элементов массива, которые находятся между максимальным положительным и минимальным отрицательным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
7. Найти сумму элементов массива, которые находятся правее максимального положительного и левее минимального отрицательного числа. Выводить на экран элементы, удовлетворяющие условию поиска.
8. Найти сумму элементов массива, которые находятся левее максимального положительного и правее минимального отрицательного числа. Выводить на экран элементы, удовлетворяющие условию поиска.
9. Вывести на экран все элементы массива, которые по модулю больше минимального положительного, и находятся левее максимального отрицательного.
10. Вывести на экран все элементы массива, которые по модулю меньше минимального положительного, и находятся левее максимального отрицательного.
11. Вывести на экран все элементы массива, которые по модулю больше минимального отрицательного, и находятся левее максимального положительного.
12. Вывести на экран все элементы массива, которые по модулю меньше минимального отрицательного, и находятся левее максимального положительного.
13. Вывести на экран все элементы массива, которые по модулю больше минимального положительного, и находятся правее максимального отрицательного.
14. Вывести на экран все элементы массива, которые по модулю меньше минимального положительного, и находятся правее максимального отрицательного.
15. Вывести на экран все элементы массива, которые по модулю больше минимального отрицательного, и находятся правее максимального положительного.
16. Вывести на экран все элементы массива, которые по модулю меньше минимального отрицательного, и находятся правее максимального положительного.
17. Вывести на экран все элементы целочисленного массива, которые по модулю меньше максимального положительного значения, четные, но не кратные k элементы. Число k считывать с экрана.
18. Вывести на экран все элементы целочисленного массива, которые по модулю больше максимального отрицательного значения, четные, но не кратные k элементы. Число k считывать с экрана.
19. Найти количество элементов массива, которые находятся между последним отрицательным и максимальным положительным числом. Выводить на экран элементы, удовлетворяющие условию поиска.
20. Найти количество элементов массива, которые находятся между первым отрицательным и минимальным положительным числом. Выводить на экран элементы, удовлетворяющие условию поиска.

Требования к оформлению программ:

1. **Содержание.** Программа должна делать то, что предусмотрено заданием. Не надо выполнять лишних действий, заданием не предусмотренных.
2. **Спецификация.** В преамбуле программы в комментариях указывать сведения:
 - Кто выполнил.
 - Что делает программа (кратко).
 - Что на входе (имена входных файлов и т.д.).

- Что на выходе (что является результатом работы программы?).

3. Ввод и вывод

- Приглашения к вводу (например, сколько чисел, какого типа и через какой разделитель нужно вводить).
- Контрольный вывод (все введенные данные выводить на экран, и только после этого выполнять необходимые вычисления.)
- «Защита от дурака». Проверять вводимые данные на корректность. Например, если необходимо считать количество чего – то, то эта величина не может быть отрицательной и т.д.

4. Структура кода. Набираемый код должен быть хорошо структурированным.

Использовать:

- Отступы.
- Комментарии – поясняют решение программы.
- Осмысленные названия переменных.
- Пояснения о назначении переменных в комментариях (кроме счетчиков).

5. Декомпозиция кода

- Функциональная. Программу оформлять с помощью функций.

6. Многофайловые проекты

- Все проекты должны состоять минимум из двух модулей: главного и подключаемого.

Главный модуль **main.cpp**. В нем оставить только функцию main, в которой вызывать функции, описанные в других модулях.

Модуль с описанием пользовательских функций **file.cpp** и **file.h**. В этот модуль перенести определение всех функций, необходимых для выполнения задания.

Контрольные вопросы:

1. Напишите алгоритм нахождения минимального числа в массиве.
2. Напишите алгоритм вычисления количества повторений числа k в массиве.
3. Напишите функцию для нахождения максимального числа и его позиции в массиве целых чисел. Результат в функции выводиться не должен. Напишите пример вызова этой функции
4. Напишите функцию для нахождения минимального числа и его позиции в массиве вещественных чисел. Результат в функции выводиться не должен. Напишите пример вызова этой функции.