## Лабораторная работа №4

## Статические одномерные массивы

***Цель лабораторной работы:*** *изучение структурной организации массивов и способов доступа к их элементам; совершенствование навыков процедурного программирования на языке C/C++ при решении задач обработки статических одномерных массивов.*

***Задание на программирование:*** *используя технологию процедурного программирования, разработать программу обработки одномерных статических массивов в соответствии с индивидуальным заданием.*

***Порядок выполнения работы:***

1. Получить у преподавателя индивидуальное задание и выполнить постановку задачи: сформулировать условие, определить входные и выходные данные, их ограничения.

2. Разработать математическую модель: описать с помощью формул и рисунков структуру массивов и процесс их преобразования.

3. Провести структуризацию задачи. С этой целью выделить в ней подзадачи, которые будут реализованы в виде отдельных функций (ввод исходных данных, вывод содержимого массива, обработка массива в соответствии с заданием и т.д.). При этом запрещается совмещать в одной функции решение нескольких подзадач.

4. Обязательное требование – введение собственных типов данных, логически точно разделяющих и группирующих информацию, используемую при решении конкретной задачи (тип значения элемента массива, тип массива и т.п.).

5. Построить схемы алгоритмов основных функций.

6. Составить программу на языке *C/C++*.

7. Входные данные на этапах тестирования и демонстрации работы преподавателю должны задаваться либо с использованием специально подобранных арифметических формул, либо вводиться с клавиатуры по запросу. **Датчики псевдослучайных чисел использовать запрещается**.

Если это явно не оговорено в конкретном варианте задания, значения элементов в каждом исходном массиве **должны быть разными**.

8. Выходные данные должны выводиться на экран с пояснениями. Операторы вывода любых результатов работы должны находиться либо в функции *main*(), либо в специальной функции вывода (например, преобразованного массива), вызов которой осуществляется из функции *main*().

9. Проверить работу программы на полном наборе тестов. Входные и выходные массивы должны выводиться в одном и том же формате.

10. Использовать стандартные потоковые объекты ввода/вывода ***cin*** и ***cout***.

11. Оформить отчет о лабораторной работе в составе: постановка задачи, математическая модель, схемы алгоритмов основных функций, текст программы, контрольные примеры (скриншоты).

12. Скриншоты тестов должны легко читаться. Все их неинформативные части должны быть удалены.

13. Номера вариантов задания те же, что и в работах №№1-3.

***Варианты индивидуальных заданий***

**1.**

Дан массив *a*0, *a*1,…, *a*2*m-*1. Определить количество и произведение абсолютных значений элементов массива с нечётными номерами, лежащих между элементом с наименьшим среди неотрицательных элементов значением и первым элементом с отрицательным значением, имеющим номер больше, чем *m*-1.

**2.**

Дан массив *b*0, *b*1,…, *bn-*1. Определить сумму значений элементов массива с чётными номерами, лежащих между первым элементом массива с отрицательным значением и последним элементом с положительным значением.

**3.**

Дан массив *c*0, *c*1,…, *c*2*n*-1. Определить абсолютную величину суммы значений элементов массива, лежащих между элементом с наименьшим среди неотрицательных элементов значением и первым элементом с отрицательным значением, имеющим номер больше, чем *n*-1.

**4.**

Дан массив *d*0, *d*1,…, *d*2*n*-1. Определить сумму значений элементов массива с чётными номерами, лежащих между элементом с наименьшим по абсолютной величине значением и первым элементом массива с положительным значением, имеющим номер больше, чем *n*-1.

**5.**

Дан массив *x*0, *x*1, *x*2,…, *xm-*1. Определить произведение суммы положительных значений элементов массива, лежащих между элементом с наибольшим среди неположительных элементов значением и последним элементом массива с отрицательным значением, на их количество.

**6.**

Дан массив *y*0, *y*1,…, *yn-*1. Определить сумму отрицательных значений элементов массива, предшествующих последнему элементу с наибольшим неположительным значением.

**7.**

Дан массив *z*0, *z*1,…, *zn-*1. Определить произведение суммы абсолютных значений элементов массива с чётными номерами, предшествующих последнему элементу массива с отрицательным значением, на общее количество элементов массива с отрицательными значениями.

**8.**

Дан массив *a*0, *a*1,…, *a*2*m-*1. Написать программу преобразования массива в массив с элементами, соответственно равными значениям элементов исходного массива *a*2*m*-1, *a*2*m*-2, *a*2*m*-3,…, *am*, *a*0, *a*1, *a*2,…, *am*-1. Дополнительные массивы не использовать.

**9.**

Дан массив *b*0, *b*1,…, *b*3*n-*1. Построить новый массив с элементами, значения которых соответственно равны *b*0, *b*3,…, *b*3*n-*3, *b*1, *b*4,…, *b*3*n-*2, *b*2, *b*5,..., *b*3*n-*1.

**10.**

Дан массив *c*0, *c*1,…, *c*2*n*-1. Определить сумму абсолютных значений элементов массива, лежащих между элементом с наименьшим среди неотрицательных элементов значением и элементом с номером *n*-1.

**11.**

Дан массив *d*0, *d*1,…, *d*2*n*-1. Определить произведение положительных значений элементов массива, лежащих между элементом с наибольшим среди неположительных элементов значением и элементом с номером *n*-1.

**12.**

Дан массив *x*0, *x*1, *x*2,…, *x*2*m-*1. Определить количество и сумму отрицательных значений элементов массива, лежащих между элементом с наименьшим среди неотрицательных элементов значением и элементом с номером *m*-1.

**13.**

Дан массив *y*0, *y*1,…, *y*2*n-*1. Определить количество и произведение положительных значений элементов массива, лежащих между первым элементом с наименьшим среди неотрицательных элементов значением и элементом с номером *n*-1.

**14.**

Дан массив *z*0, *z*1,…, *z*2*n-*1. Определить сумму значений элементов массива с чётными номерами, лежащих между элементом с наибольшим среди неположительных элементов значением и элементом с номером *n*-1.

**15.**

Дан массив *a*0, *a*1,…, *a*2*m-*1. Определить произведение значений элементов массива с чётными номерами, лежащих между элементом с наименьшим среди неотрицательных элементов значением и первым элементом с отрицательным значением, имеющим номер больше, чем *m*-1.

**16.**

Дан массив *b*0, *b*1,…, *b*2*n-*1. Определить сумму значений элементов массива с нечётными номерами, лежащих между элементом с наибольшим среди неположительных элементов значением и последним элементом с положительным значением, имеющим номер меньше, чем *n*.

**17.**

Дан массив *c*0, *c*1,…, *c*2*n*-1. Определить произведение значений элементов массива с чётными номерами, лежащих между элементом с наибольшим среди неположительных элементов значением и последним элементом с отрицательным значением, имеющим номер больше, чем *n*-1.

**18.**

Дан массив *d*0, *d*1,…, *d*2*n*-1. Определить произведение отрицательных значений элементов массива, лежащих между первым элементом с наименьшим среди неотрицательных элементов значением и элементом с номером *n*-1.

**19.**

Дан массив *x*0, *x*1, *x*2,…, *xm-*1. Определить количество и сумму положительных значений элементов массива с нечётными номерами, лежащих между первым по порядку элементом с наименьшим значением и последним по порядку элементом с наибольшим значением.

**20.**

Дан массив *y*0, *y*1,…, *yn-*1. Найти число элементов этого массива, значения которых принадлежат отрезку [*c*, *d*] и предшествуют первому элементу с наименьшим среди неотрицательных элементов значением.

**21.**

Дан массив *z*0, *z*1,…, *z*2*n-*1. Построить новый массив с элементами, значения которых соответственно равны *z*2*n*-1, *z*0, *z*2*n-*2*, z*1,…, *zn, zn-*1.

**22.**

Дан массив *a*0, *a*1,…, *am-*1. Значения всех элементов массива с четными номерами, предшествующих элементу с наибольшим среди неположительных элементов значением, заменить их квадратами.

**23.**

Дан массив *b*0, *b*1,…, *b*2*n-*1. Построить новый массив с элементами, значения которых соответственно равны *b*0, *bn*, *b*1, *bn*+1,… , *bn-*1, *b*2*n-*1.

**24.**

Дан массив *c*0, *c*1,…, *c2n*-1. Если значение наибольшего среди неположительных элементов левой половины этого массива больше значения наибольшего среди неположительных элементов правой половины этого массива, то значения всех отрицательных элементов массива заменить их квадратами.

**25.**

Даны упорядоченные по возрастанию значений их элементов массивы *a*0, *a*1,…, *am-*1 и *b*0, *b*1,…, *bn-*1. Построить из них новый массив *c*0, *c*1,…, *cn*+*m*-1, также упорядоченный по возрастанию значений его элементов.

**26.**

Дан массив *d*0, *d*1,…, *dn*-1. Определить произведение значений элементов этого массива с чётными номерами, предшествующих элементу с наибольшим среди неположительных элементов значением.

**27.**

Дан массив *x*0, *x*1, *x*2,…, *xm-*1. Определить произведение абсолютных значений элементов массива с чётными номерами и сумму значений элементов с нечётными номерами, номера которых больше номера элемента с наименьшим среди неотрицательных элементов значением.

**28.**

Дан массив *y*0, *y*1,…, *yn-*1. Определить сумму отрицательных значений элементов массива с чётными номерами, лежащих между первым элементом с наибольшим значением и последним элементом с наименьшим значением.

**29.**

Дан массив *z*0, *z*1,…, *zn-*1. Определить количество и произведение значений элементов массива, лежащих между элементами с наибольшим неположительным значением и с наименьшим по абсолютной величине значением.

**30.**

Дан массив *a*0, *a*1,…, *am-*1. Определить количество и произведение положительных значений элементов массива, лежащих между элементами с наибольшим по абсолютной величине значением и с наименьшим неотрицательным значением.

**31.**

Дан массив *b*0, *b*1,…, *bn-*1. Определить сумму, произведение и количество компонент этого массива, значения которых принадлежат отрезку [*x*, *y*] и предшествуют первому элементу с наибольшим среди неположительных элементов значением.

**32.**

Преобразовать исходный массив *c*0, *c*1,…, *c*2*n*-1 так, чтобы его элементы расположились в следующем порядке: *c*0, *c*2*n*-1, *c*1, *c*2*n*-2,…, *cn*-1, *cn*. Дополнительных массивов не использовать.

**33.**

Дан массив *d*0, *d*1,…, *dn*-1. Определить сумму положительных значений элементов массива с нечётными номерами, лежащих между первым элементом с наименьшим значением и последним элементом с наибольшим значением.

**34.**

Из массива *x*0, *x*1, *x*2,…, *x*3*m-*1. получить массив *y*0, *y*1,…, *ym-*1, очередная компонента которого равна среднему арифметическому тройки очередных компонент массива *x*.

**35.**

Дан массив *z*0, *z*1,…, *zn-*1. Определить произведение отрицательных значений элементов массива с нечётными номерами, лежащих между первым элементом с наименьшим значением и первым элементом с наибольшим значением.

**36.**

Дан целочисленный массив *a*0, *a*1,…, *am-*1, среди элементов которого могут быть элементы с равными значениями. Построить новый массив, включив в него из каждой группы равных между собой элементов только один, выбросив все остальные, и сохранив исходный порядок их следования.

**37.**

Дан массив *b*0, *b*1,…, *bn-*1. Построить новый массив, значения элементов которого равны *b*0, *b*0+*b*1, *b*0+*b*1+*b*2, , *b*0+*b*1+*b*2+…+*bn-*1.

**38.**

Дан массив *c*0, *c*1,…, *c*2*n*-1. Определить произведение значений элементов массива с нечётными номерами, лежащих между элементом с наименьшим среди неотрицательных элементов значением и последним элементом с отрицательным значением, имеющим номер больше, чем *n*-1.

**39.**

Дан массив *d*0, *d*1,…, *dn*-1. Из всех непрерывных участков этого массива, состоящих из элементов с нулевым значением, выбрать наибольший по длине. Вывести индексы его начала и конца.

**40.**

Дан массив *x*0, *x*1, *x*2,…, *x*2*m-*1. Определить произведение отрицательных значений элементов массива, лежащих между элементом с наименьшим среди неотрицательных элементов значением и первым элементом с отрицательным значением, имеющим номер меньше, чем *m*.

**41.**

Дан массив *y*0, *y*1,…, *y*2*n-*1. Определить сумму положительных значений элементов массива, лежащих между элементом с наибольшим среди неположительных элементов значением и первым элементом с положительным значением, имеющим номер больше, чем *n*-1.

**42.**

Дан массив *z*0, *z*1,…, *z*2*n-*1. Определить сумму значений элементов массива с нечётными номерами, лежащих между элементом с наименьшим среди неотрицательных элементов значением и последним элементом с отрицательным значением, имеющим номер меньше, чем *n*.

**43.**

Дан массив *a*0, *a*1,…, *a*2*m-*1. Определить произведение значений элементов массива с нечётными номерами, лежащих между элементом с наибольшим среди неположительных элементов значением и последним элементом с отрицательным значением, имеющим номер больше, чем *m*-1.

**44.**

Дан массив *b*0, *b*1,…, *b*2*n-*1. Построить новый массив с элементами, значения которых соответственно равны *bn-*1, *bn-*2, …, *b*0, *b*2*n*-1, *b*2*n*-2, …, *bn*. Исходный массив оставить без изменения.

**45.**

Дан массив *c*0, *c*1,…, *c*4*n*-1. Построить новый массив с элементами, значения которых соответственно равны *c*4*n*-2, *c*4*n*-1, *c*0, *c*1, *c*4*n*-4, *c*4*n*-3, *c*2, *c*3, …, *c*2*n*, *c*2*n*+1, *c*2*n*-2, *c*2*n*-1. Исходный массив оставить без изменения.

**46.**

Дан массив *d*0, *d*1,…, *d*4*n*-1. Преобразовать исходный массив так, чтобы его элементы расположились в следующем порядке: *d*4*n*-2, *d*4*n*-1, *d*0, *d*1, *d*4*n*-4, *d*4*n*-3, *d*2, *d*3, …, *d*2*n*, *d*2*n*+1, *d*2*n*-2, *d*2*n*-1. Дополнительных массивов не использовать.

**47.**

Дан массив *x*0, *x*1, *x*2,…, *x2n-*1. Преобразовать исходный массив так, чтобы его элементы расположились в следующем порядке: *xn-*1, *xn-*2, …, *x*0, *x*2*n*-1, *x*2*n*-2, …, *xn*. Дополнительных массивов не использовать.

**48.**

Дан массив *y*0, *y*1,…, *y*2*n-*1. Построить новый массив с элементами, значения которых соответственно равны *y*0, *y*2*n*-1, *y*1, *y*2*n*-2,…, *yn*-1, *yn*. Исходный массив оставить без изменения.

**49.**

Дан массив *z*0, *z*1,…, *z*2*n-*1. Построить новый массив с элементами, значения которых соответственно равны *zn*, *zn*+1, ,…, *z*2*n-*1, *zn*-1, *zn-*2,…, *z*0. Исходный массив оставить без изменения.

**50.**

Дан массив *a*0, *a*1,…, *am-*1. Определить сумму, произведение и количество компонент этого массива, значения которых принадлежат отрезку [*x*, *y*] и предшествуют первому элементу с наименьшим среди неотрицательных элементов значением.

***Пример 1 решения задачи для варианта задания вида:***

Дан массив *a*0, *a*1,…, *a*2*m-*1. Значения элементов массива, лежащих между первым элементом с наименьшим среди неотрицательных элементов значением и первым элементом с отрицательным значением, имеющим номер больше, чем *m*-1, удвоить. У всех остальных элементов массива инвертировать знак.

***Математическая модель решения***

Решение задачи начинается с ввода исходных данных. Из формулировки задачи понятно, что исходный массив имеет **чётное** число (2*m*) элементов. Следовательно, прежде всего, необходимо ввести значение (*m*) половины размера массива.

После этого необходимо задать значения всех элементов массива. Доступ к элементам массива осуществляется по их индексу (номеру элемента массива). Перебираем индексы элементов массива от 0 до 2*m*-1 (нумерация элементов массива начинается с 0) и задаём значения элементов с текущими номерами.

Пусть исходный массив имеет вид (*m*=5):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** |  |  | ***m*-1** | ***m*** |  |  | **2*m*-2** | **2*m*-1** |
| **1** | **2** | **0** | **-3** | **4** | **5** | **6** | **-7** | **-8** | **9** |

Теперь можно приступать к решению задачи. Вначале определяем индекс первого элемента массива, имеющего минимальное неотрицательное значение.

При этом могут быть два варианта: неотрицательных значений вообще нет или хотя бы один такой элемент есть. Вначале предполагаем, что неотрицательных значений нет, и присваиваем переменной *imin* значение -1 (такого индекса у элементов массива не может быть). После этого ищем первый элемент массива с неотрицательным значением. Если такой элемент в массиве есть, фиксируем индекс этого элемента *imin*. Величину этого элемента массива принимаем за исходное минимальное неотрицательное значение элемента.

Далее, перебирая индексы остальных элементов массива с *imin*+1 до 2*m*-1, для неотрицательных элементов сравниваем их значения со значением элемента с индексом *imin*. Если значение элемента массива с текущим индексом меньше значения элемента массива с индексом *imin*, запоминаем индекс этого элемента, меняя значение *imin*. В результате индекс элемента массива, имеющего минимальное неотрицательное значение, зафиксирован в переменной *imin*.

Если ни один элемент массива с неотрицательным значением не найден, выполнение решения всей задачи прекращается.

Аналогичные действия выполняем далее для определения индекса первого элемента массива, имеющего отрицательное значение для элементов с индексами >*m*-1. Для этого перебираем индексы элементов массива с *m* до 2*m*-1. Если такой элемент в массиве есть, фиксируем индекс этого элемента *iotr*.

Если ни один элемент массива с индексом >*m*-1 и отрицательным значением не найден, выполнение решения всей задачи прекращается.

Если в обоих случаях элементы массива найдены, то результат в нашем случае выглядит следующим образом:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** |  |  | ***m*-1** | ***m*** |  |  | **2*m*-2** | **2*m*-1** |
| **1** | **2** | **0** | **-3** | **4** | **5** | **6** | **-7** | **-8** | **9** |
|  |  | ***imin*** |  |  |  |  | ***iotr*** |  |  |

При этом возможны два случая: *imin*>*iotr* и *imin*<*iotr*. Поскольку по условию задачи необходимо удвоить значения элементов массива, лежащих **между** элементами с индексами *iotr* и *imin*, необходимо сравнить *iotr* и *imin*.

Если *imin*<*iotr*, то индекс начального элемента интервала удвоения значений *i\_beg*=*imin*+1, а индекс конечного элемента этого интервала *i\_end*=*iotr*-1. В противном случае *i\_beg*=*iotr*+1, а *i\_end*=*imin*-1.

Теперь инвертируем значения элементов массива с индексами от 0 до *ibeg*-1 и от *i\_end*+1 до 2*m*-1, а значения элементов с индексами от *i\_beg* до *i\_end* удваиваем.

В результате для рассматриваемого примера преобразованный массив имеет вид:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **-1** | **-2** | **0** | **-6** | **8** | **10** | **12** | **7** | **8** | **-9** |
|  |  |  | ***ibeg*** |  |  | ***iend*** |  |  |  |

Задача решена.

***Схема алгоритма поиска индекса первого элемента массива, имеющего минимальное неотрицательное значение (функция nomer\_min)***

*nomer\_min*(*a*, *n*)

*imin*=-1

*i*=0; *i*<*n*; *i*++

нет

*a*[*i*]>=0

да

*imin*=*i*

*break*

нет

*imin*==*i*

да

*i*=*imin*+1; *i*<*n*; *i*++

*a*[*i*]>=0&&*a*[*i*]<a[*imin*]

нет

да

*imin*=*i*

*return*(*imin*)

***Схема алгоритма поиска индекса первого отрицательного элемента массива с номером >m*-1 *(функция nomer\_otr)***

*nomer\_otr*(*a*, *m*)

*iotr*=-1

*i*=*m*; *i*<2*m*; *i*++

нет

*a*[*i*]<0

да

*iotr*=*i*

*break*

*return*(*iotr*)

***Схема алгоритма преобразования массива (функция obrabotka\_mas)***

*obrabotka\_mas*(*a*, *n*, *i\_min*, *i\_otr*)

да нет

*i\_min*<*i\_otr*

*i\_beg*=*i\_otr*+1

*i\_beg*=*i\_min*+1

*i\_end*=*i\_min*

*i\_end*=*i\_otr*

*i*=*i\_beg*; *i*<*i\_end*; *i*++

*a*[*i*]*×*=2

*i*=0; *i*<*i\_beg*; *i*++

*a*[*i*]=-*a*[*i*]

*i*=*i\_end*; *i*<*n*; *i*++

*a*[*i*]=-*a*[*i*]

*return*

***Текст программы (пример 1)***

//Дан массив a[2m]. Значения элементов массива, лежащие между первым элементом

//с наименьшим неотрицательным значением и первым отрицательным элементом,

//имеющим индекс >=m, удвоить. У остальных элементов массива инвертировать знак.

#include<iostream>

#include<stdlib.h>

#include<locale.h>

using namespace std ;

const int RAZ = 10 ; //предельный размер массива

typedef int telem ; //определение типа значений элементов массива

typedef telem tmas[RAZ] ; //определение типа массива

int nomer\_min(const tmas, int) ;

int nomer\_otr(const tmas, int) ;

void inputmas(tmas, int) ;

void outputmas(const tmas, int) ;

void obrabotka\_mas(tmas, int, int ,int) ;

//main\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int main()

{

setlocale(LC\_ALL, "Russian");

tmas a; //массив

int m, //половина размера массива

i\_otr, //индекс первого отрицательного элемента >m

i\_min, //индекс наименьшего неотрицательного элемента

i; //ответ в диалоге

//Ввод исходных данных

cout << "\n Введете половину размера массива <= " << RAZ / 2 << " : ";

cin >> m;

//Контроль ввода размера массива

if (2 \* m > RAZ || m <= 0)

{

cout << " Введено недопустимое значение размера массива" << endl;

cout << "\n Повторить-1, Выход-2: ";

cin >> i;

if (i == 1) main();

return 0;

}

inputmas(a, 2 \* m);

//Вывод исходного массива

cout << " Исходный массив:" << endl;

outputmas(a, 2 \* m);

//Поиск индекса элемента с наименьшим неотрицательным значением

i\_min = nomer\_min(a, 2 \* m);

if (i\_min == -1)

{

cout << "\n Неотрицательных значений у элементов массива нет" << endl;

cout << " Массив не изменялся." << endl;

cout << "\n Повторить-1, Выход-2: ";

cin >> i;

if (i == 1) main();

return 0;

}

cout << "\n Индекс min неотр. элемента массива = " << i\_min << endl;

i\_otr = nomer\_otr(a, m);

if (i\_otr == -1)

{

cout << "\n Отрицательных значений у элементов массива";

cout << " с индексами >=" << m << " нет" << endl;

cout << " Массив не изменялся.";

cout << "\n Повторить-1, Выход-2: ";

cin >> i;

if (i == 1) main();

return 0;

}

cout << " Индекс первого отр. элемента >=" << m << " = " << i\_otr << endl;

obrabotka\_mas(a, 2 \* m, i\_min, i\_otr);

cout << " Измененный массив:" << endl;

outputmas(a, 2 \* m);

cout << "\n Повторить-1, Выход-2: ";

cin >> i;

if (i == 1) main();

return 0;}

//main\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//nomer\_min\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int nomer\_min(const tmas a, int n)

{int imin , //индекс минимального неотр. Элемента

I ; //кущий номер элемента массива

imin = -1 ; //факт отсутствия неотрицательного элемента

//Ищем первый неотрицательный элемент

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

if(a[i] >= 0)

{imin = i ;

break ;

}

if(imin == i)

{for(int i = imin + 1 ; i < n ; i++)

if(a[i] >= 0 && a[i] < a[imin]) imin = i ;

}

return imin ;

}

//nomer\_min\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//nomer\_otr\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int nomer\_otr(const tmas a, int m)

{int iotr ; //индекс первого отрицательного элемента

iotr = -1 ; //факт отсутствия отрицательного элемента

//Ищем первый отрицательный элемент с индексом >m

for(int i = m ; i < 2 \* m ; i++)

if(a[i] < 0)

{iotr = i ;

break ;

}

return iotr ;

}

//nomer\_otr\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//inputmas\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void inputmas(tmas a, int n)

{cout << " Введите в одной строке элементы массива, состоящего из " << n ;

cout << "\n чисел, и нажмите <Enter>" << endl ;

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

cin >> a[i] ;

}

//inputmas\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//outputmas\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void outputmas(const tmas a, int n)

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

{cout.width(4) ; //ширина поля для вывода значения элемента массива

cout << a[i] << " " ;

}

}

//outputmas\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//obrabotka\_mas\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void obrabotka\_mas(tmas a, int n, int i\_min, int i\_otr)

{int i\_beg, //начало интервала

i\_end, //конец интервала

i ; //текущий номер элемента

if(i\_min < i\_otr)

{i\_beg = i\_min + 1 ;

i\_end = i\_otr ;

}

else{i\_beg = i\_otr + 1 ;

i\_end = i\_min ;

}

for(i = i\_beg ; i < i\_end ; i++) //удваиваем значения внутри интервала

a[i] \*= 2 ;

for(i = 0 ; i < i\_beg ; i++) //инвертируем знак у остальных

a[i] = -a[i] ;

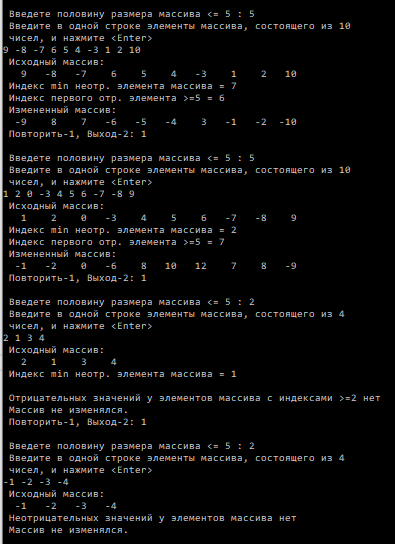
for(i = i\_end ; i < n ; i++) //инвертируем знак у остальных

a[i] = -a[i] ;

}

//obrabotka\_mas\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Скриншоты результатов выполнения программы***



***Пример 2 решения задачи для варианта задания вида:***

Дан массив *a*0, *a*1,…, *am-*1. Найти и вывести номер элемента этого массива, для которого сумма абсолютных величин разностей со значениями соседних элементов максимальна.

Для крайних элементов использовать циклическое замыкание.

***Математическая модель решения задачи примера 2***

Решение задачи начинается с ввода исходных данных. Прежде всего, необходимо ввести значение (*m*) размера массива.

После этого необходимо задать значения всех элементов массива. Доступ к элементам массива осуществляется по их индексу (номеру данного элемента массива). Поэтому перебираем все индексы элементов массива от 0 до *m*-1 (нумерация элементов массива начинается с 0) и задаём значения элементов с текущими номерами.

Пусть исходный массив имеет вид (*m*=8):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 |  |  |  |  | *m*-2 | *m*-1 |
| **1** | **4** | **8** | **2** | **5** | **3** | **7** | **6** |

Теперь можно приступать к решению задачи. Вначале определяем сумму абсолютных величин разностей со значениями соседних элементов для элемента массива с индексом 0. Поскольку этот элемент массива является крайним, по условию задачи используем циклическое замыкание элементов массива и определяем величину *max*=*abs*(*a*[*m*-1]-*a*[0])+*abs*(*a*[1]-*a*[0]). Эту величину принимаем за исходное значение искомой суммы и фиксируем индекс этого элемента *imax*=0.

Далее, перебирая индексы элементов массива с 1 до *m*-1, для остальных элементов вычисляем сумму абсолютных величин разностей их значений со значениями соседних элементов и сравниваем эти значения со значением *max*. Если значение вычисленной суммы превышает текущее значение *max*, меняем значение *max* и запоминаем индекс этого элемента *imax*. При этом для элемента массива с индексом *m*-1 используем циклическое замыкание элементов массива и искомую величину суммы определяем как *abs*(*a*[0]*-a*[*m*-1])*+abs*(*a*[*m*-2]-*a*[*m*-1]).

В результате индекс элемента массива, для которого сумма абсолютных величин разностей со значениями соседних элементов максимальна, зафиксирован в переменной *imax,* а сама величина суммы зафиксирована в переменной *max*.

Для рассматриваемого примера индекс искомого элемента *imax*=2, при этом сумма абсолютных величин разностей со значениями соседних элементов *max*=10:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 |  |  |  | *m*-2 | *m*-1 |
| **1** | **4** | **8** | **2** | **5** | **3** | **7** | **6** |
|  |  | ***imax*** |  |  |  |  |  |

Задача решена.

***Схема алгоритма поиска искомого индекса элемента массива (функция nomer)***

*nomer*(*a*, *n*, &*max*)

*imax*=0

*max*=*abs*(*a*[*n*-1]-*a*[0])+*abs*(*a*[1]-*a*[0])

*i*=1; *i*<*n*-1; *i*++

*pr=abs*(*a*[*i*-1]*-a*[*i*])*+abs*(*a*[*i*+1]-*a*[*i*])

нет

*max*<*pr*

да

*imax*=*i*

*max*=*pr*

*pr=abs*(*a*[0]*-a*[*n*-1])*+abs*(*a*[*n*-2]-*a*[*n*-1])

*max*<*pr*

нет

да

*imax*=*n*-1

*max*=*pr*

*return*(*imax*)

***Текст программы (пример 2)***

//Дан массив. Найти и вывести номер элемента этого массива,

//для которого сумма разностей с соседними элементами максимальна.

//Для крайних элементов использовать циклическое замыкание.

#include<iostream>

#include<math.h>

#include<locale.h>

using namespace std ;

const int RAZ = 10 ; //предельный размер массива

typedef int telem ; //определение типа значений элементов массива

typedef telem tmas[RAZ] ; //определение типа массива

int nomer(const tmas a, int n, telem &max) ;

void inputmas(tmas a, int n) ;

//main\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int main()

{setlocale(LC\_ALL,"Russian") ;

tmas a ; //массив

int n, //размер массива

nom , //номер искомого элемента

i ; //ответ в диалоге

telem max ; //значение максимальной разности

cout << "\n В массиве целых чисел найти номер элемента, для которого" << endl ;

cout << " сумма разностей с соседними элементами максимальна" << endl ;

//Ввод исходных данных

cout << " Введете размер массива <= " << RAZ << " " ;

cin >> n ;

//Контроль ввода размера массива

if(n > RAZ || n <= 0)

{cout << " Введено недопустимое значение размера массива" << endl ;

cout << "\n Повторить-1, Выход-2: ";

cin >> i;

if (i == 1) main();

return 0 ;

}

inputmas(a, n) ;

//Поиск номера элемента

nom = nomer(a, n, max) ;

cout << " Искомый номер элемента массива: " << nom << endl ;

cout << " Значение элемента:" << a[nom] << " , сумма разностей=" << max ;

cout << "\n Повторить-1, Выход-2: ";

cin >> i;

if (i == 1) main();

return 0 ;

}

//main\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//nomer\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Поиск искомого индекса элемента

int nomer(const tmas a, int n, telem &max)

{telem pr ; //текущее значение разности

int imax = 0 ; //за максимум принимаем первый по счету элемент

max = abs(a[n - 1] - a[0]) + abs(a[1] - a[0]) ;

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

if(max < (pr = abs(a[i-1] - a[i]) + abs(a[i+1] - a[i])))

{imax = i ;

max = pr ;

}

if(max < (pr = abs(a[0] - a[n - 1]) + abs(a[n - 2] - a[n-1])))

{imax = n - 1 ;

max = pr ;

}

return imax ;

}

//nomer\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//inputmas\_begin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Ввод исходного массива

void inputmas(tmas a, int n)

{cout << "\n Введите в одной строке элементы массива, состоящего из";

cout << "\n " << n << " целых чисел, и нажмите <Enter>" << endl;

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

cin >> a[i] ;

}

//inputmas\_end\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Скриншоты результатов выполнения программы (пример 2)***

