

Лабораторная работа № 6

Перестановки в одномерных массивах

Оглавление

Цель	1
Инструкция:	1
Задания для самостоятельного выполнения:	2
Домашнее задание:	3
БЛОК А	3
БЛОК В	3
БЛОК С	3
Требования к оформлению программ:	5
Контрольные вопросы:	5

Цель

Научиться переставлять элементы массива по выбранному правилу, в обратном порядке, выполнять циклический сдвиг.

Инструкция:

Задача: Переставить пятый и максимальный элементы массива

Алгоритм решения:

1. Заполнить массив числами a
2. Найти позицию (индекс) максимального элемента в массиве i_max
3. Выполнить проверку есть ли i -ый элемент в массиве и не совпадает ли он с i_max ? По условиям задачи $i=4$. (*Индексация в массивах начинается с нуля, поэтому пятый элемент имеет индекс 4*)
4. Если элемент с индексом i есть, и он не совпадает с i_max , то переставить элементы по правилу: $c=a[i]$; $a[i]=a[i_max]$; $a[i_max]=c$;
5. Вывести результирующий массив.

Решение: Использовать программу из Лабораторной работы 5, внести изменения в главную программу:

```

//-----main.cpp-----
//Лабораторная работа №6
//Перестановки в массивах

#include <iostream>
#include "vect.h"
#include "calc_vect.h"
#include "max_min_vect.h"
#define n 5

using namespace std;

int main()
{
    double MAX, x;
    int num=0, k=5;

    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;

    if(k<=0 || k>n)
        cout<<"Error of change: incorrect k, k must be from [ 1,"<<n<<"] !"<<endl;
    else
    {
        cout<<"V["<< k <<"] = "<<v1[k-1]<<endl;
        cout<<endl;
        if (k-1!=num)
        {
            x=v1[num];    v1[num]=v1[k-1];    v1[k-1]=x;
        }
    }

    cout<<"Massive after changing:"<<endl;
    OutputDescVect(v1,n);

    delete []v1; v1=NULL;
}

```

Задания для самостоятельного выполнения:

Написать функции и продемонстрировать их работу:

1. Инверсии массива (перестановка наоборот).
2. Циклического сдвига. В программе выполнять циклический сдвиг k -раз. Количество k считывать с экрана.

Домашнее задание:

БЛОК А

Задание на отметку «удовлетворительно»

Выполнить задание и оформить код программы в соответствии с требованиями. Подготовить письменные ответы на контрольные вопросы. В тетрадь выписывать вопрос и ответ на него.

Задание:

Использовать «заготовку» проекта из лабораторного задания №3.

Выполнить задание:

1. Переставить первый и минимальный элементы массива.
2. Переставить максимальный и минимальный элементы массива.

БЛОК В

Задания на отметку «хорошо»

Дан массив из n целых чисел.

1. Переставить k-ый и максимальный элементы массива.
2. Переставить k-ый и минимальный элементы массива.
3. Переставить k-ый и первый равный нулю элемент массива.
4. Переставить k-ый и последний равный нулю элемент массива.
5. Переставить k-ый и первый не равный нулю элемент массива.
6. Переставить k-ый и последний не равный нулю элемент массива.
7. Переставить k-ый и первый нечетный элемент массива.
8. Переставить k-ый и последний нечетный элемент массива.
9. Переставить k-ый и первый четный элемент массива.
10. Переставить k-ый и последний четный элемент массива.
11. Переставить k-ый и первый кратный трем элемент массива.
12. Переставить k-ый и последний кратный трем элемент массива.
13. Переставить k-ый и первый отрицательный элемент массива.
14. Переставить k-ый и последний отрицательный элемент массива.
15. Переставить k-ый и первый положительный элемент массива.
16. Переставить k-ый и последний положительный элемент массива.
17. Переставить максимальный и минимальный элементы массива.
18. В массиве из четного числа элементов переставить первый со вторым, третий с четвертым и т.д.
19. В массиве из четного числа элементов переставить первый с третьим, второй с четвертым и т.д.
20. Переставить два случайных элемента массива.

БЛОК С

Задания на отметку «отлично»

1. Дан массив из n целых чисел. Переставить в нем элементы таким образом, чтобы индексы перечислялись в следующем порядке: $1 \ n \ 2 \ (n-1) \ 3 \ (n-2) \dots$. Дополнительный массив заводить нельзя.
2. Дан массив из n целых чисел. Переставить в нем элементы таким образом, чтобы индексы перечислялись в следующем порядке: $n \ 1 \ (n-1) \ 2 \ (n-2) \ 3 \dots$. Дополнительный массив заводить нельзя.
3. Дан массив из n целых чисел, заполненный 0 и 1. Переставить в нем элементы таким образом, чтобы нули и единицы чередовались. Если нулей и единиц неодинаковое количество, то все «непарные» элементы перечислять в конце массива. Дополнительный массив заводить не разрешается.
4. Дан массив из n целых чисел, заполненный -1 и 1. Переставить в нем элементы таким образом, чтобы положительные и отрицательные элементы чередовались. Если положительных и отрицательных элементов неодинаковое количество, то все «непарные» элементы перечислять в конце массива. Дополнительный массив заводить не разрешается.
5. Дан массив из n целых чисел, заполненный 2 и 1. Переставить в нем элементы таким образом, чтобы четные и нечетные элементы чередовались. Если четных и нечетных элементов неодинаковое количество, то все «непарные» элементы перечислять в конце массива. Дополнительный массив заводить не разрешается.
6. Дан массив из n целых чисел. Все не равные нулю элементы перенести в начало массива, а остальные в конец с сохранением порядка следования. Дополнительный массив заводить не разрешается.
7. Дан массив из n целых чисел. Все положительные элементы перенести в начало массива, а остальные в конец с сохранением порядка следования. Дополнительный массив заводить не разрешается.
8. Дан массив из n целых чисел. Все четные элементы перенести в начало массива, а остальные в конец с сохранением порядка следования. Дополнительный массив заводить не разрешается.
9. Дан массив из n целых чисел. Все равные нулю элементы перенести в начало массива, а остальные в конец с сохранением порядка следования. Дополнительный массив заводить не разрешается.
10. Дан массив из n целых чисел. Все кратные k элементы перенести в начало массива, а остальные в конец с сохранением порядка следования. Дополнительный массив заводить не разрешается.
11. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него каждый нечетный элемент. Дополнительный массив заводить не разрешается.
12. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него каждый третий элемент. Дополнительный массив заводить не разрешается.
13. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него каждый четный элемент. Дополнительный массив заводить не разрешается.
14. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него равные нулю элементы. Дополнительный массив заводить не разрешается.
15. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него все элементы больше 10. Дополнительный массив заводить не разрешается.
16. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него все элементы меньше 10. Дополнительный массив заводить не разрешается.
17. Дан массив из n целых чисел. Все элементы меньше 20 перенести в начало массива, а остальные в конец с сохранением порядка следования. Дополнительный массив заводить не разрешается.
18. Дан массив из n целых чисел. Все элементы больше 20 перенести в начало массива, а остальные в конец с сохранением порядка следования. Дополнительный массив заводить не разрешается.

19. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него максимальный и минимальный элементы. Дополнительный массив заводить не разрешается.
20. Дан целочисленный массив из n элементов. Сжать массив, выбросив из него все повторяющиеся элементы. Дополнительный массив заводить не разрешается.

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. Можно ли переставить два элемента массива в два действия? Почему?
2. Как правильно переставить i -ый и j -ый элементы массива?
3. Напишите алгоритм перестановки элементов массива в обратном порядке
4. Напишите алгоритм циклического сдвига элементов массива