## . Постановка задачи

1. Сформировать массив из n элементов с помощью датчика случайных чисел (n задается пользователем с клавиатуры).
2. Распечатать полученный массив.
3. Выполнить удаление указанных элементов из массива.
4. Вывести полученный результат.
5. Выполнить добавление указанных элементов в массив.
6. Вывести полученный результат.
7. Выполнить перестановку элементов в массиве.
8. Вывести полученный результат.
9. Выполнить поиск указанных в массиве элементов и подсчитать количество сравнений, необходимых для поиска нужного элемента.
10. Вывести полученный результат.
11. Выполнить сортировку массива указанным методом.
12. Вывести полученный результат.
13. Выполнить поиск указанных элементов в отсортированном массиве и подсчитать количество сравнений, необходимых для поиска нужного элемента.
14. Вывести полученный результат.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант | Удаление | Добавление | Перестановка | Поиск | Сортировка |
| 1 | Максимальный элемент | К элементов в начало массива | Перевернуть массив | Первый четный | Простой обмен |

## Методические указания

1. При решении задач использовать псевдодинамические массивы. Псевдодинамические массивы реализуются следующим образом:

1. при определении массива выделяется достаточно большое количество памяти:

const int MAX\_SIZE=100;/\*именованная константа\*/int mas[MAX\_SIZE];

1. пользователь вводит реальное количество элементов массива меньшее N.

int n; cout<<”\nEnter the size of array<”<<MAX\_SIZE<<”:”;cin>>n;

1. дальнейшая работа с массивом ограничивается заданной пользователем размерностью n.

2. Формирование массива осуществляется с помощью датчика случайных чисел. Для этого можно использовать функцию int rand(), которая возвращает псевдослучайное число из диапазона 0..RAND\_MAX=32767, описание функции находится в файле <stdlib.h>. В массиве должны быть записаны и положительные и отрицательные элементы. Например, оператор a[I]=rand()%100-50; формирует псевдослучайное число из диапазона [-50;49].

3. Вывод результатов должен выполняться после выполнения каждого задания. Элементы массива рекомендуется выводить в строчку, разделяя их между собой пробелом.

## Содержание отчета:

1. Постановка задачи (общая и конкретного варианта).
2. Анализ поставленного задания: определить к какому классу задач относится задача и объяснить почему.
3. Текст программы.
4. Результаты тестов.
5. Решение одной из задач с использованием указателей для доступа к элементам массива.

#include <iostream>

#include <stdlib.h>

#include <windows.h>

using namespace std;

void main ()

{

setlocale(LC\_ALL,"Russian");

unsigned int n=0, maxi;

cout<<"Введите количество элементов в массиве < 100 = ";

if ((scanf("%i",&n) != 1))

{

cout<<"необходимо было ввести количество элементов!";

Sleep(4000);

exit(0);

}

if (n>=100)

{

cout<<"необходимо было ввести количество элементов согласно условию!";

Sleep(4000);

exit(0);

}

int m[100],i;

cout<<"\nВыводим начальный массив\n\n";

for(i=0;i<=(n-1);i++) m[i]=rand()%100-50;

for(i=0;i<=(n-1);i++) cout<<m[i]<<"["<<i+1<<"]"<<", ";

cout<<"\b\b.\n\n";

cout<<"\n\nИщем максимальный элемент и удаляем его из начального массива\n\n";

signed int max=m[0];

for(i=0;i<=(n-1);i++)

{

if (max<m[i])

{

max=m[i];

maxi=i;

}

}

cout<<max<<" ["<<++maxi<<"] - максимальный элемент\n\n";

for(i=maxi-1;i<=(n-1);i++) m[i]=m[i+1];

m[n-1]=0;

for(i=0;i<=(n-2);i++) cout<<m[i]<<"["<<i+1<<"]"<<", ";

cout<<"\b\b.\n\n";

unsigned int k=0;

cout<<"\nВведите число элементов, которые необходимо добавить в начало массива\n";

if ((scanf("%i",&k) != 1))

{

cout<<"необходимо было ввести число элементов!";

Sleep(4000);

exit(0);

}

if (k>(101-n))

{

cout<<"число элементов должно было быть не больше числа свободных мест в массиве!";

Sleep(5000);

exit(0);

}

cout<<"\n";

for(i=0;i<=(n-2);i++) m[n+k-2-i]=m[n-2-i];

for(i=0;i<=k-1;i++) m[i]=rand()%100-50;

for(i=0;i<=(n-2+k);i++) cout<<m[i]<<"["<<i+1<<"]"<<", ";

cout<<"\b\b.\n\n\n";

cout<<"Перевернем массив\n";

signed int buff;

for(i=0;i<=((n+k-1)/2-1);i++)

{

buff=m[n+k-2-i];

m[n+k-2-i]=m[i];

m[i]=buff;

}

for(i=0;i<=(n+k-2);i++) cout<<m[i]<<"["<<i+1<<"]"<<", ";

cout<<"\b\b.\n\n\n";

unsigned int ifeven;

for(i=0;i<=(n+k-2);i++)

{

if (m[i]%2==0) {ifeven=i; break;}

}

cout<<"("<<m[ifeven]<<") - первое четное число массива. Количество сравнений: "<<ifeven+1<<"\n\n\nСортировка методом простого обмена\n\n";

for(i=1;i<n+k-2;i++)

for(int j=n+k-2;j>=i;j--)

{

if (m[j]<m[j-1])

{

buff=m[j];

m[j]=m[j-1];

m[j-1]=buff;

}

}

for(i=0;i<=(n+k-2);i++) cout<<m[i]<<"["<<i+1<<"]"<<", ";

cout<<"\b\b.\n\n\n";

for(i=0;i<=(n+k-2);i++)

{

if (\*(m+i)%2==0) {ifeven=i; break;}

}

cout<<"("<<m[ifeven]<<") - первое четное число массива. Количество сравнений: "<<ifeven+1<<"\n\n\n";

system("pause");

}

