(ARRAYS AS FUNCTION)

(MADE BY ALI AKBER)

(BSCS IST (SS1)

Write a menu driven C++ program with following option.

i. Accept elements of an array.

ii. Display elements of an array.

iii. Sum of all elements of an array.

iv. Sum of even and odd elements of an array.

V. Display maximum & minimum element of an array.

vi. Sort elements in descending order by using Bubble sort

vii. Sort elements in Acceding order by using Selection sort

viii. Search a no form array by using sequential search.

ix. Search a no form array by using Binary search.

X. Maximum 3 numbers from array.

Xi. Minimum 3 numbers from array.

Xii. Display odd and even number.

Xiii. Display total number of odd and even numbers.

Xiv. Swap the maximum and minimun values of array.

Xv. Swap the ist and last values of array.

Xvi. Reverse the array.

Xvii. Double the values of array.

Xviii. Find Duplicate element in the array.

Xix. Exit program.

SOLUTION:

```
#include <iostream>
using namespace std;
                    void getarray(int [],int);
                    void showarray(int[],int);
                    void linearsearch(int [],int);
                    void selectionsort(int[],int);
                    void bubblesort(int[],int);
                    void binarysearch(int [],int);
                    void arraysum(int [],int);
                    void odd_evensum(int [],int);
                    void display_odd_even(int [],int );
                    void total_odd_even(int [],int);
                    void swap(int[],int);
                    void max_min(int[],int);
                    void reverse(int [],int);
                    void arraydouble(int [],int);
                    void swap_maximum(int [],int);
                    void duplicate(int [],int);
                    void maximum(int [],int);
                    void minimum(int [],int);
int main(){
const int size=5;
int arr[size];
int opt;
while (true)
```

```
{
        cout<<"\n\t~~ MENU ~~\n"<<endl;
cout<<"1- Accept an element of array"<<endl;
cout<<"2- Display elements of an array"<<endl;
cout<<"3- Search a number from array using linear search"<<endl;
cout<<"4- Sort elements in ascending order using Selection Sorting"<<endl;
cout<<"5- Sort elements in descending order using Bubble Sorting"<<endl;
cout<<"6- Search a number using Binary Search"<<endl;
cout<<"7- Display sum of array"<<endl;
cout<<"8- Display sum of odd and even number"<<endl;
cout<<"9- Display odd and even numbers separatly"<<endl;
cout<<"10- Display Total number of odd and even numbers"<<endl;
cout<<"11- Swap Ist and last element of the array"<<endl;
cout<<"12- Display maximum and minimum element from the array"<<endl;
cout<<"13- Reverse the elements of the array"<<endl;
cout<<"14- Convert array values into its double"<<endl;
cout<<"15- Swap Maximum and minimum values of array"<<endl;
cout<<"16- Find duplicate elements in the array"<<endl;
cout<<"17- Find three maximum numbers in the array"<<endl;
cout<<"18 Find three minimum numbers in the array"<<endl;
cout<<"19 Exit Program"<<endl;
cout<<"enter option"<<endl;
cin>>opt;
switch(opt)     {
        case 1:
                getarray(arr,size);
```

```
break;

case 2:

showarray(arr,size);

break;

case 3:

    linearsearch (arr,size);

    break;

    case 4:

        selectionsort(arr,size);

        break;

        case 5:

            bubblesort(arr,size);

            break;

            case 6:

                binarysearch(arr,size);

                break;

                case 7:

                    arraysum(arr,size);

                    break;

                    case 8:

                        odd_evensum(arr,size);

                        break;

                        case 9:

                        display_odd_even(arr,size);

                        break;

                        case 10:
```

```
                                    total_odd_even(arr,size);
                                    break;
                                    case 11:
                                        swap(arr,size);
                                        break;
                                        case 12:

    max_min(arr,size);
                                            break;
                                            case 13:

    reverse(arr,size);
                                                break;
                                                case 14:

    arraydouble(arr,size);

    break;

    case 15:

    swap_maximum(arr,size);
break;

    case 16:

        duplicate(arr,size);

        break;
```

```
}
        case 17:
            maximum(arr,size);
            break;
            case 18:
            minimum(arr,size);
            case 19:
                exit(1);
                default:
                    cout<<"invalid option"<<endl;

}
}
return 0;
}
void getarray(int arr[],int s){
    cout<<endl;
    for (int i=0; i<s; i++){
        cout<<"enter value at "<<i<<" index ";
        cin>>arr[i];
    }
}
```

```
void showarray(int arr[],int s){
    cout<<endl;
    for (int j=0 ;j<s; j++){
        cout<<"the value at "<<j<<" index is "<<arr[j]<<endl;
    }
}
    void linearsearch(int arr[],int s){
        int key;
  cout<<"enter an element for search"<<endl;
  cin>>key;
  int found=-1;
  for (int k=0; k<s; k++){
  if (arr[k]==key){
  found=k;
  }
  }
  if (found!=-1){
  cout<<key<<" is found at "<<found<<endl;
  }
  else
cout<<"key is not found "<<endl;
}
void selectionsort(int arr[],int s){
    for (int j=0; j<s; j++){
     for ( int k=j+1; k<s; k++) {
        if (arr[k]<arr[j] ){
            int temp =arr[k];
                arr[k]=arr[j];
                arr[j]=temp;
            }
        }
    }
    cout<<" sorted array is "<<endl;
    for ( int l=0; l<s; l++) {
        cout<<arr[l]<<endl;
    }
}
void bubblesort(int arr[],int s){
      int counter=1;
    while (counter<5) {
        for ( int k=0; k<s-counter ; k++) {
            if (arr[k]<arr[k+1] ){
            int temp =arr[k];
                arr[k]=arr[k+1];
                arr[k+1]=temp;
            }
        }
        counter++;
    }
    cout<<" sorted array is "<<endl;
    for ( int l=0; l<s; l++) {
        cout<<arr[l]<<endl;
    }

}

void binarysearch(int arr[],int b){

      int s=0;

      int e=4;

      int mid;

      int found=-1;

      int key;

      cout<<"enter a number for search"<<endl;

      cin>>key;

      while (s<=e && found==-1)

      {


      mid=(s+e)/2;

      if (arr[mid]==key){

      found=mid;

      break;

}

else if (arr[mid]>key){

      e=mid-1;

}

else

      s=mid+1;

}

if (found==-1)

cout<<key<<" not found "<<endl;
```

```
else
cout<<key<<"    found at index "<<found<<endl;
}
void arraysum(int arr[],int s){
int sum=0;
cout<< "the sum of    array are"<<endl;
for (int k=0; k<s; k++)
sum=sum+ arr [k];
cout<<sum<<endl;
}
void odd_evensum(int arr[],int s){
int oddsum=0;
int evensum=0;
for (int k=0; k<s; k++){
if (arr[k]%2==0)
evensum=evensum+arr [k];
else
oddsum=oddsum+arr[k];
}
cout<<"sum of odd numbers are "<<oddsum<<endl;
cout<<"sum of even numbers are "<<evensum<<endl;
}
void display_odd_even(int arr[],int s){
int odd;
int even;
cout<<"even numbers are"<<endl;
```

```
for (int k=0; k<s; k++){
if (arr[k]%2==0)
{
cout<<arr[k]<<endl;
}
}
cout<<"odd numbers are"<<endl;
for (int k=0; k<s; k++){
if (arr[k]%2!=0)
cout<<arr [k]<<endl;
}
}
        void total_odd_even(int arr[],int s){
                int odd=0, even=0;
                for (int i=0 ;i<s; i++){
if (arr[i]%2 == 0)
{
even++;
}
else
{
odd++;
}
}
cout<<"total odd integers are "<<odd<<endl;
cout<<"total even integers are "<<even<<endl;
```

```
    }
    void swap(int arr[],int n){
        int temp;
temp =0;
temp=arr[0];
arr[0]=arr [n-1];
arr [n-1]=temp;
cout<<"new array is"<<endl;
for (int j=0; j<n; j++)
{
cout<<arr [j]<<endl;;
}
    }
    void max_min(int arr[],int s){
        int max,min;
        max=arr [0];
min =arr[0];
for (int j=0 ; j<s ;j++)
{
if (max<arr[j])
max=arr [j];
if (min>arr[j])
min=arr [j];
}
cout<<"the maximum value from integers is "<<max<<endl<<"the minimum value from integers is "<<min<<endl;
```

```
    }
    void reverse(int arr[],int s){
        cout<<"new array is"<<endl;
for (int j=s-1 ; j>=0 ;j--)
{
cout<<arr [j]<<endl;;
}
    }
    void arraydouble(int arr[],int s){
        cout<<"the double value in array are"<<endl;
        int doubvalue;
for (int i=0; i<s; i++)
{
doubvalue=arr[i]*2;
cout<<doubvalue<<endl;
}
}
    void swap_maximum(int arr[],int s){
        int max,min,m,n, temp;
max=min=arr[0];
m=n=0;
for (int j=0; j<s; j++)
{
if (max<arr[j])
{
max=arr [j];
```

```
m=j;
}
if (min>arr[j])
{
min=arr [j];
n=j;
}
}
temp =arr [m];
arr[m]=arr[n];
arr [n]=temp;
cout<<"new array is"<<endl;
for (int k=0; k<s ; k++)
{
cout<<arr [k]<<endl;
}
        }
        void duplicate(int arr[],int s){
                cout<<"duplicate numbers are"<<endl;
for (int i=0; i<s ;i++)
{
for (int j=i+1; j<s;j++)
if (arr[i]==arr[j])
{
cout<<arr[i]<<endl;
break;
```

```
}

}

}

void maximum(int arr[],int s){

            int fmax,smax,tmax;

fmax=smax=tmax=arr[0];

for (int j=1 ; j<s ; j++)

{

if (arr[j]>fmax)

{

tmax= smax;

smax=fmax;

fmax=arr [j];

}

else if (arr[j]>smax)

{

tmax= smax;

smax=arr [j];

}

else if (arr[j]>tmax)

{

tmax=arr [j];

}

}

cout<<" First largest element is : "<<fmax<<endl;

cout<<" Second largest element is : "<<smax<<endl;
```

```
cout<<" Third largest element is : "<<tmax<<endl;

}

void minimum(int arr[],int s){

            int fmin,smin,tmin;

fmin=smin=tmin=arr[0];

for (int j=1 ; j<s ; j++)

{

if (arr[j]<fmin)

{

tmin= smin;

smin=fmin;

fmin=arr [j];

}

else if (arr[j]<smin)

{

tmin= smin;

smin=arr [j];

}

else if (arr[j]<tmin)

tmin=arr [j];

}

cout<<" First smallest element is : "<<fmin<<endl;

cout<<" Second smalest element is : "<<smin<<endl;

cout<<" Third smallest element is : "<<tmin<<endl;

}
```