Submitted By	Chowdhury Fatmi Monzur Neha
Student ID	23201141
Course Code	CSE (108)
Submitted Too	Zaima Sartaj Taheri Lecturer, Department Of Computer Science & Engineering

1.Find the minimum and maximum element in an array.

Solution:

```
#include<stdio.h>
int main( ){
   int n,i,max,min;
   int num [100];

   printf("Enter the number of elements in the array: ");
   scanf("%d",&n);
   printf("Enter %d element: \n",n);
   for(i=0;i<n;i++){
      scanf("%d",&num[i]);
   }

    min=num[0];
    max=num[0];

   for(i=1;i<n;i++){
      if(num[i]<min){
         min=num[i];
      }
if(num[i]>max){
   max=num[i];
  }
}
printf("Minimum value = %d\n",min);
printf("Maximum value = %d\n",max);

return 0;
}
```

2.Array Reverse

Solution:

```
#include<stdio.h>
int main( ){

int arr[10],i;

printf("Enter array element: ");
for(i=0; i<=9; i++){
scanf("%d",&arr[i]);
}

printf("Reverse array elements:\n ");
for(i=9; i>=0; i--){
printf("%d",arr[i]);
}


return 0;
}
```

3. Write a C program to cyclically rotate an array by one.

Solution:

```
#include<stdio.h>
int main( ){
 int i,n;

 printf("Enter the size of the array:");
 scanf("%d",&n);

 int arr[n];

 printf("Enter the element of the array:\n");
 for(i=0;i<n;i++)
 {
 scanf("%d",&arr[i]);
 }
 int last=arr[n-1];
 for(i=n-1;i>0;i--)
 {
 arr[i]=arr[i-1];
 }
arr[0]=last;

printf("Rotated array:\n");
for(i=0;i<n;i++)
{
printf("%d",arr[i]);
}

return 0;
}
```

4.Sort an array

Solution:

```
#include<stdio.h>
int main( ){

int n,i,j,arr[100],temp;

printf("enter the number of elements:\n");
scanf("%d",&n);

printf("Enter the element:\n");
for(i=0;i<n;i++){
scanf("%d",&arr[i]);
}

for(j=i+1;j<n;j++){
if(arr[i]>arr[j])
{
```

```
temp=arr[i];
arr[i]=arr[j];
arr[j]=temp;
  }
}
printf("Elements sorted in assending order are\n");
for(i=0;i<n;i++){
printf("%d",arr[i]);
}
return 0;
}
```

5.Find Duplicates in an array

Solution:

```
#include<stdio.h>

int main() {
   int i, arr[100], j, no;



   printf("Enter the size of array: ");
   scanf("%d", &no);



   printf("Enter any %d elements in the array: ", no);
   for(i = 0; i < no; i++) {
       scanf("%d", &arr[i]);
   }



   printf("Duplicate elements are:\n");
   for(i = 0; i < no; i++) {
       for(j = i + 1; j < no; j++) {
           if(arr[i] == arr[j]) {
               printf("%d\n", arr[i]);
               break;
           }
       }
   }

   return 0;
}
```

6.Find the Occurrance in an integer in the array.

Solution:

```
#include<stdio.h>
int main( ){
int a[]={1,1,2,2,2,3,4,4,5};
int n = sizeof(a)/sizeof(a[0]);
int X = 2;
int Y = 4;
int count = 0;
for(int i=0;i<n;i++){
if(a[i]==X){
count++;
}
else if(a[i]>X){
}
}

for(int i =0;i<n;i++){
if(a[i]==Y){
count++;
}
else if(a[i]>Y){
break;
}
}
return 0;
}
```

8.Move all the negative elements to one side of the array.

Solution:

```
#include <stdio.h>
void rearrangeArray(int a[], int n) {
int left = 0, right = n - 1;
while (left <= right) {
if (a[left] < 0)
left++;

else if (a[right] >= 0)

right--;

else {

int temp = a[left];

a[left] = a[right];

a[right] = temp;

left++;

right--;

  }

}

}

int main() {

int a[] = {-12, 11, -13, -5, 6, -7, 5, -3, -6};

int n = sizeof(a) / sizeof(a[0]);

rearrangeArray(a, n);

printf("Rearranged array: \n");

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

printf("%d ", a[i]);

}

printf("\n");

return 0;

}
```

9.Find the row with a maximum number of 1’s.

Solution:

```c
#include <stdio.h>
int rowWithMax1s(int a[][5], int n, int m) {
int max_row_index = -1;
int j = m - 1;
for (int i = 0; i < n; i++) {
while (j >= 0 && a[i][j] == 1) {
j--;
max_row_index = i;
}
}
return max_row_index;
}
int main() {
int a[][5] = {
{0, 0, 0, 1, 1},
{0, 1, 1, 1, 1},
{0, 0, 0, 0, 1},
{0, 0, 1, 1, 1},
};
int n = sizeof(a) / sizeof(a[0]);
int m = sizeof(a[0]) / sizeof(a[0][0]);
int result = rowWithMax1s(a, n, m);
if (result != -1)
printf("The row with the maximum number of 1s is: %d\n", result);
else
printf("No 1s found in the array.\n");
return 0;

}
```

10.Majority elements

Solution:

```
#include <stdio.h>

int findCandidate(int a[], int n) {

int candidate = 0, count = 1;

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

if (a[i] == a[candidate])

count++;

else

count--;

if (count == 0) {

candidate = i;

count = 1;

}

}

return a[candidate];

}

int isMajority(int a[], int n, int candidate) {

int count = 0;

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

if (a[i] == candidate)

count++;

}
if (count > n / 2)
return candidate;
else
return -1;
}
int majorityElement(int a[], int n) {
int candidate = findCandidate(a, n);
return isMajority(a, n, candidate);
}
int main() {
int a[] = {2, 2, 1, 1, 2, 2, 2};
int n = sizeof(a) / sizeof(a[0]);
int result = majorityElement(a, n);
if (result != -1)
printf("The majority element is: %d\n", result);
else
printf("No majority element exists.\n");
return 0;
}
```

11.Wave array

Solution:

```
#include <stdio.h>
void BubbleSort(int a[], int n) {
for (int i = 0; i < n - 1; i++) {
for (int j = 0; j < n - i - 1; j++) {
if (a[j] > a[j + 1]) {
```

```
int temp = a[j];
a[j] = a[j + 1];
a[j + 1] = temp;
    }
  }
 }
}
void sortInWave(int a[], int n) {
BubbleSort(a, n);
for (int i = 0; i < n - 1; i += 2) {
int temp = a[i];
a[i] = a[i + 1];
a[i + 1] = temp;
 }
}
int main() {
int a[] = {10, 5, 6, 3, 2, 20, 100, 80};
int n = sizeof(a) / sizeof(a[0]);
sortInWave(a, n);
printf("Array in wave form: \n");
for (int i = 0; i < n; i++) {
printf("%d ", a[i]);
}
printf("\n");
return 0;
}
```