1. Write a function to find the greatest number from the given array of any size. (TSRS).

```
#include<stdio.h>
int max(int a[]);
int main()
{
  int i,a[10];
      printf("Enter ten numbers\n");
  for(i=0;i<10;i++)
      scanf("%d",&a[i]);
   max(a);
  return 0;
}
 max(int a[])
 {
   int t=0,i,j;
   for(i=0;i<10;i++)
 {
   for(j=0;j<10;j++)
   if(t<a[i])
      t=a[i];
 }
    printf("Maximum number is %d",t);
}
```

2. Write a function to find the smallest number from the given array of any size. (TSRS) .

```
#include<stdio.h>
int min(int a[]);
int main()
{
  int i,a[10];
```

```
      printf("Enter ten numbers\n");
  for(i=0;i<10;i++)
      scanf("%d",&a[i]);
   min(a);
  return 0;
}
 min(int a[])
 {
   int t,i,j;
   for(i=0;i<10;i++)
 {
   for(j=0;j<10;j++)
   if(a[i]<a[j])
     {
      t=a[i];
      a[i]=a[j];
      a[j]=t;
   }
 }
printf("Minimum number is %d",a[0]);
}
```

3. Write a function to sort an array of any size. (TSRN)

```
#include<stdio.h>
int sort(int a[]);
int main()
{
  int i,a[10];
      printf("Enter ten numbers\n");
  for(i=0;i<10;i++)
```

```
        scanf("%d",&a[i]);
        printf("Sorted array is:\n");
    sort(a);
   return 0;
}
sort(int a[])
{
    int t,i,j;
    for(i=0;i<10;i++)
  {
    for(j=0;j<10;j++)
    if(a[i]<a[j])
       {
        t=a[i];
        a[i]=a[j];
        a[j]=t;
    }
  }for(i=0;i<10;i++)
   printf("%d ",a[i]);
}
```

4. Write a function to rotate an array by n position in d direction. The d is an indicative value for left or right. (For example, if array of size 5 is [32, 29, 40, 12, 70]; n is 2 and d is left, then the resulting array after left rotation 2 times is [40, 12, 70, 32, 29])'

```
#include<stdio.h>
void RightRotation(int a[],int size,int d)
 {
    int i,j,x;
    for(i=1;i<=d;i++)
    {
```

```
        x = a[size-1];
        for(j=size-1;j>0;j--)
            a[j] = a[j-1];
        a[0] = x;
    }for(i=0;i<size;i++)
    printf("%d ",a[i]);
}
 int main()
{
     int a[]={1,2,3,4,5,6,7,8,9};
     RightRotation(a,9,2);
     return 0;
}
```

5. Write a function to find the first occurrence of adjacent duplicate values in the array. Function has to return the value of the element.

```
#include <stdio.h>
int findFirstOccurrence(int nums[], int n, int target)
{
int low = 0, high = n - 1;
int result = -1;
while (low <= high)
    {
    int mid = (low + high)/2;
if (target == nums[mid])
        {
            result = mid;
            high = mid - 1;
        }
else if (target < nums[mid]) {
```

```
            high = mid - 1;
        }
else {
            low = mid + 1;
        }
    }
return result;
}
int main()
{
    int nums[] = {2, 5, 5, 5, 6, 6, 8, 9, 9, 9};
    int n = sizeof(nums)/sizeof(nums[0]);
    int target = 8;
    int index = findFirstOccurrence(nums, n, target);
    if (index != -1)
    {
        printf("The first occurrence of element %d is located at index : %d\n",target, index);
    }
    else {
        printf("Element not found in the array\n");
    }
    return 0;
}
```

6. Write a function in C to read n number of values in an array and display it in reverse order.

```
#include<stdio.h>
rev(int a[]);
 int main()
 {
     int a[10],i;
```

```
    printf("Enter ten numbers\n");
   for(i=0;i<10;i++)
   scanf("%d",&a[i]);
    printf("Reverse order is:\n");
    rev(a);
    return 0;
}
rev(int a[])
{
 int i,j,temp;
 for(i=0,j=9;i<=j;i++,j--)
    {
     temp=a[i];
     a[i]=a[j];
     a[j]=temp;
    }
     for(i=0;i<10;i++)
     printf("%d ",a[i]);
}
```

7. Write a function in C to count a total number of duplicate elements in an array.(Means elements that occurs 2 times in an array).

```
#include<stdio.h>
 int duplicate( int a[])
{
    int count=0;
    int i,j;
for(i=0;i<10;i++)
{
    for(j=i+1;j<10;j++)
```

```
  {
      if(a[i] == a[j])
      count++;
  }
}
       printf("Total number of duplicate elements is %d",count);

}
int main()
{
  int a[10];
  int i;
  printf("Enter 10 elements of array\n");
     for(i=0;i<10;i++)
   scanf("%d ",&a[i]);
    duplicate(a);
  return 0;
}
```

8. Write a function in C to print all unique elements in an array.

```
#include<stdio.h>
int main()
{
  int n,i;
  printf("Enter size of an array : ");
  scanf("%d",&n);
  int a[n];
  printf("Enter elements of an array:\n");
 for(i=0;i<n;i++){
  scanf("%d",&a[i]);
```

```
 }unique_elements(a,n);
  return 0;
}
int unique_elements(int a[],int n)
{
    int i,j,count = 1;
    for(i=0;i<n;i++)
    {
       for(j=0;j<n;j++)
       {
          if(a[i]==a[j]&&i!=j)
            break;
          }if(j==n)
          {
             printf("Unique elements in an array is %d : %d\n",count,a[i]);
             ++count;
             }
    }return -1;
}
```

9. Write a function in C to merge two arrays of the same size sorted in descending order.

```
#include <stdio.h>
int mergearray(int a[], int b[], int arr1size, int arr2size)
{

   int arr_resultsize = arr1size + arr2size;
   int c[arr_resultsize], i, j;
for (i = 0; i < arr1size; i++) {
       c[i] = a[i];
   }
```

```
    for (i = 0, j = arr1size;
         j < arr_resultsize && i < arr2size; i++, j++) {
        c[j] = b[i];
    }
    for ( int k = 0; k < arr_resultsize; k++) {
        printf("%d ", c[k]);
    }
}
int main()
{
    int arr1size = 5, arr2size = 5;
    int a[5] = { 1, 2, 3, 4, 5 };
int b[5] = { 6, 7, 8, 9, 10 };
    printf("%d", mergearray(a, b, arr1size, arr2size));
    return 0;
}
```

10. Write a function in C to count the frequency of each element of an array

```
#include<stdio.h>
frequency(int a[],int size)
{
     int hash[10] ={0},i;
    for(i=0;i<20;i++)
        hash[a[i]]++;
    for(i=0;i<10;i++)
    {
        if(hash[i]!=0)
        printf("%d -->%d\n",i,hash[i]);
    }
}
```

```
int main()
{
    int a[]={2,3,4,5,6,7,8,9,1,2,5,3,7,9,2,1,3,4,1};
     frequency(a,20);
   return 0;
}
```