Assignment- 15

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

```c
#include <stdio.h>
int largest(int a[], int n)
{
    int max = a[0];
    for (int i = 1; i < n; i++)
    {
        if (max < a[i])
        {
            max = a[i];
        }
    }
    return max;
}
int main()
{
    int n = 5;
    int a[n];
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    int large = largest(a, n);
    printf("Largest element in an array : %d", large);
    return 0;
}
```

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

```c
#include <stdio.h>
int smallest(int a[], int n)
{
    int min = a[0];
    for (int i = 1; i < n; i++)
    {
        if (min > a[i])
        {
            min = a[i];
        }
    }
    return min;
}
```

```c
int main()
{
    int n = 5;
    int a[n];
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    int small = smallest(a, n);
    printf("Smallest element in an array : %d", small);
    return 0;
}
```

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

```c
#include <stdio.h>
void sort(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        int min = a[i];
        for (int j = i + 1; j < n - 1; j++)
        {
            if (a[i] > a[j])
            {
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    for (int i = 0; i < n; i++)
    {
        printf("%d ", a[i]);
    }
}
int main()
{
    int n = 5;
    int a[n];
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    sort(a, n);
    return 0;
```

```
}
```

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 rotate(int a[], int n, int d)
{
    while (d != 0)
    {
        int temp = a[0];
        for (int i = 0; i < n - 1; i++)
        {
            a[i] = a[i + 1];
        }
        a[n - 1] = temp;
        d--;
    }
    for (int i = 0; i < n; i++)
    {
        printf("%d ", a[i]);
    }
}
int main()
{
    int n = 5;
    int a[n], d;
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Enter no. of rotation : ");
    scanf("%d", &d);
    rotate(a, n, d);
    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 duplicate(int a[], int n)
{
```

```
    for (int i = 0; i < n; i++)
    {
        if (a[i] == a[i + 1])
            return a[i];
    }
}
int main()
{
    int n = 5;
    int a[n];
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    int value = duplicate(a, n);
    printf("Adjacent duplicate value in an array : %d", value);
    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>
void reverse(int a[], int n)
{
    for (int i = n - 1; i >= 0; i--)
        printf("%d ", a[i]);
}
int main()
{
    int n = 5;
    int a[n];
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("After reverse an array : \n");
    reverse(a, n);
    return 0;
}
```

7. Write a function in C to count a total number of duplicate elements in an array.

```
#include <stdio.h>
```

```c
int duplicate(int a[], int n)
{
    int count = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (a[i] == a[j])
            {
                count++;
                break;
            }
        }
    }
    return count;
}
int main()
{
    int n = 10;
    int a[n];
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    int value = duplicate(a, n);
    printf("No. of duplicate element : %d", value);
    return 0;
}
```

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

```c
#include <stdio.h>
void unique(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        int count = 0;
        for (int j = 0; j < n; j++)
        {
            if (i != j)
            {
                if (a[i] == a[j])
                {
                    count = 1;
                    break;
                }
            }
```

```
        }
        if (count == 0)
        {
            printf("%d ", a[i]);
        }
    }
}
int main()
{
    int n = 7;
    int a[n];
    printf("Enter %d elements : ", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Unique elements in an array : ");
    unique(a, n);
    return 0;
}
```

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

```
#include <stdio.h>
int main()
{
    int a[] = {1, 3, 5, 7, 9};
    int b[] = {2, 4, 6, 8, 10};
    int c[10], k = 0, i = 0, j = 0;
    for (k = 0; k < 10; k++)
    {
        if (i >= 5)
        {
            while (k < 10)
            {
                c[k] = b[j];
                j++;
                k++;
                if (k == 10)
                    break;
            }
        }
        else if (a[i] < b[j])
        {
            c[k] = a[i];
            i++;
        }
```

```
        else
        {
            c[k] = b[j];
            j++;
        }
    }
    for (k = 0; k < 10; k++)
    {
        printf("%d ", c[k]);
    }
    return 0;
}
```

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

```
#include <stdio.h>
int main()
{
    int a[]={2,4,6,2,4,9,2};
    int h[100]={0};
    for(int i=0; a[i] ; i++)
    {
        h[a[i]]= h[a[i]] + 1;
    }
    for(int i=0; i<100; i++)
    {
        if(h[i]>0)
        {
            printf("%d ---- %d ",i,h[i]);
            printf("\n");
        }
    }
    return 0;
}
```