Q1.Write a function to find the greatest number from the given array of any size. (TSRS)

```c
#include <stdio.h>

int greatest(int brr[], int);

int main(int argc, char *argv[])
{
    int size;

    printf("Enter array size = ");
    scanf("%d", &size);

    int arr[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value of arr[%d] = ",i);
        scanf("%d",&arr[i]);
    }

    printf("Greatest number = %d", greatest(arr, size));

    return 0;
}

int greatest(int brr[], int size)
{
    int point = brr[0];

    for (int i = 0; i < (size - 1); i++)
        if (point < brr[i + 1])
            point = brr[i + 1];

    return point;
}
```

Q2. Write a function to find the smallest number from the given array of any size. (TSRS)

```
#include <stdio.h>

int smallest(int[], int);

int main(int argc, char *argv[])
{
    int size;

    printf("Enter size = ");
    scanf("%d", &size);

    int arr[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value of arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    printf("smallest = %d", smallest(arr, size));

    return 0;
}

int smallest(int brr[], int size)
{
    int point = brr[0];

    for (int i = 0; i < (size - 1); i++)
        if (point > brr[i + 1])
            point = brr[i + 1];

    return point;
}
```

Q3. Write a function to sort an array of any size. (TSRS)

```c
#include <stdio.h>

int *sort(int brr[], int size);

int main(int argc, char *argv[])
{
    int size;

    printf("Enter size = ");
    scanf("%d", &size);

    int arr[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value of arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    printf("\n\nAfter sort\n\n");

    int *brr = sort(arr, size);

    for (int i = 0; i < size; i++)
    {
        printf("%d ", brr[i]);
    }

    return 0;
}

int *sort(int brr[], int size)
{
    for (int i = 0; i < (size - 1); i++)
    {
        for (int j = i + 1; j < size; j++)
        {
            if (brr[i] > brr[j])
            {
                int tmp = brr[j];
                brr[j] = brr[i];
                brr[i] = tmp;
            }
        }
    }
    return brr;
}
```

Q4. 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 rotation(int arr[], int size, int n, int d);

int main(int argc, char *argv[])
{
    int size, n, d;

    printf("Enter array size = ");
    scanf("%d", &size);

    int arr[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value of arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    printf("\n\nEnter n position = ");
    scanf("%d", &n);

    printf("\n\nEnter direction (1 for left & 2 for right) = ");
    scanf("%d", &d);

    rotation(arr, size, n, d);

    return 0;
}

void rotation(int arr[], int size, int n, int d)
{
    if (n >= size)
    {
        printf("Enter n positions smaller than array size");
    }
    else if (d == 1)
    {
        for (int i = 0; i < n; i++)
        {
            int tmp = arr[0];

            for (int j = 0; j < (size - 1); j++)
            {
                arr[j] = arr[j + 1];
            }

            arr[size - 1] = tmp;
        }

        printf("\n\nAfter left rotation\n\n");
        for (int i = 0; i < size; i++)
        {
            printf("%d ", arr[i]);
        }
    }
    else if (d == 2)
    {
        for (int i = 0; i < n; i++)
        {
            int tmp = arr[size - 1]; // 1 1 2 3 4 5

            for (int j = (size - 1); j > 0; j--)
            {
                arr[j] = arr[j - 1];
            }

            arr[0] = tmp;
        }

        printf("\n\nAfter right rotation\n\n");
        for (int i = 0; i < size; i++)
        {
            printf("%d ", arr[i]);
        }
    }
}
```

Q5. Write a function to find the first occurrence of adjacent duplicate values in the array. Function has to return the value of the element.

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

int fun(void);

int main(int argc, char *argv[])
{

    printf("\n\nduplicate values = %d", fun());

    return 0;
}

int fun()
{
    int size;

    printf("Enter size of array = ");
    scanf("%d", &size);

    int arr[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    for (int i = 0; i < (size - 1); i++)
    {
        if (arr[i] == arr[i + 1])
            return arr[i];
    }

    exit(0);
}
```

Q6. Write a function in C to read n number of values in an array and display it in reverse order.

```c
#include <stdio.h>

void reverse(void);

int main(int argc, char *argv[])
{
    reverse();

    return 0;
}

void reverse()
{
    int size;

    printf("Enter size of array = ");
    scanf("%d", &size);

    int arr[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    printf("\n\nArray element in reverse order\n\n");

    for (int i = (size-1); i >=0 ; i--)
    {
        printf("%d ", arr[i]);
    }

}
```

Q7. Write a function in C to count a total number of duplicate elements in an array.

```
#include <stdio.h>

void count(void);

int main(int argc, char *argv[])
{
    count();
    return 0;
}

void count()
{
    int size, count = 0;

    printf("Enter size of array = ");
    scanf("%d", &size);

    int arr[size];
    int tmp[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value arr[%d] = ", i);
        scanf("%d", &arr[i]);
        tmp[i] = -1;
    }

    for (int i = 0; i < (size - 1);)
    {
        for (int j = (i + 1); j < size; j++)
        {
            if (arr[i] == arr[j])
            {
                count += 1;
                tmp[j] = j;
            }
        }
        i += 1;

        while (i == tmp[i])
        {
            i += 1;
        }
    }
    printf("\n\nTotal number of duplicate element = %d", count);
}
```

Q8. Write a function in C to print all unique elements in an array.

```c
#include <stdio.h>

void count(void);

int main(int argc, char *argv[])
{
    count();
    return 0;
}

void count()
{
    int size, flag = 0, x = -1;

    printf("Enter size of array = ");
    scanf("%d", &size);

    int arr[size];
    int tmp[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value arr[%d] = ", i);
        scanf("%d", &arr[i]);
        tmp[i] = -1;
    }

    for (int i = 0; i < (size - 1); i++)
    {
        flag = 0;

        for (int j = (i + 1); j < size; j++)
        {
            if (arr[i] == arr[j])
            {
                tmp[i] = arr[i];
                tmp[j] = arr[j];
                break;
            }
        }
    }
    for (int i = 0; i < size; i++)
        {
            if (arr[i] != tmp[i])
            {
                printf("%d ", arr[i]);
            }
        }
}
```

Q9. Write a function in C to merge two arrays of the same size sorted in descending order.

```
#include <stdio.h>

void merge(void);

int main(int argc, char *argv[])
{

    merge();

    return 0;
}

void merge()
{
    int size, i = 0, j = 0, k = 0;

    printf("Enter size of array = ");
    scanf("%d", &size);

    int arr1[size], arr2[size], arr3[size * 2];

    for (int x = 0; x < size; x++)
    {
        printf("Enter value arr1[%d] = ", x);
        scanf("%d", &arr1[x]);
    }

    printf("\n\n");

    for (int x = 0; x < size; x++)
    {
        printf("Enter value arr2[%d] = ", x);
        scanf("%d", &arr2[x]);
    }

    while ((i < size) && (j < size))
    {
        if (arr1[i] > arr2[j])
        {
            arr3[k] = arr1[i];
            i++;
            k++;
        }
        else
        {
            arr3[k] = arr2[j];
            j++;
            k++;
        }
    }

    if (i > 4)
    {
        while (j < size)
        {
            arr3[k] = arr2[j];
            j++;
            k += 1;
        }
    }
    else if (j > 4)
    {
        while (i < size)
        {
            arr3[k] = arr1[i];
            i++;
            k += 1;
        }
    }

    for (int x = 0; x < (size * 2); x++)
    {
        printf("%d ", arr3[x]);
    }
}
```

Q10. Write a function in C to count the frequency of each element of an array.

```c
#include <stdio.h>

void fun();

int main(int argc, char *argv[])
{
    fun();
    return 0;
}

void fun()
{
    int size;

    printf("Enter array size = ");
    scanf("%d", &size);

    int arr[size], tmp[size];

    for (int i = 0; i < size; i++)
    {
        printf("Enter value of arr[%d] = ", i);
        scanf("%d", &arr[i]);
        tmp[i] = -1;
    }

    for (int i = 0; i < (size - 1); i++)
    {
        int count = 1;

        for (int j = (i + 1); j < size; j++)
        {
            if (arr[i] == arr[j] && arr[i] != tmp[i])
            {
                count += 1;
                tmp[j] = arr[j];
            }
        }
        if (arr[i] != tmp[i])
        {
            printf("%d = ", arr[i]);
            printf("%d\n", count);
        }
    }
}
```