Assignment - 15
Array and Functions in C Language

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

Program -

```
#include <stdio.h>

int findGreatest(int[],int);
int main()
{
    int n , i;

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

    int a[n];

    printf("Enter %d array values\n",n);
    for(i = 0 ; i < n ; i++)
       scanf("%d",&a[i]);


    printf("Greatest number in the array is %d",findGreatest(a,n));

    return 0;
}

int findGreatest(int b[] , int N)
{
    int i;
    int G = b[0];
    for(i = 1 ; i < N ; i++)
    {
        if(G < b[i])
        {
            G = b[i];
        }
    }
    return G;
}
```

Output -

```
Enter the size of the array: 5
Enter 5 array values
99 12 45 100 78
Greatest number in the array is 100
```

---

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

Program -

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

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

    int a[n];

    printf("Enter %d array values\n",n);
    for(i = 0 ; i < n ; i++)
       scanf("%d",&a[i]);


    printf("Smallest number in the array is %d",findSmallest(a,n));

    return 0;
}

int findSmallest(int b[] , int N)
{
    int i;
    int S = b[0];
    for(i = 1 ; i < N ; i++)
    {
        if(S > b[i])
        {
            S = b[i];
        }
    }
    return S;
}
```

Output -

```
Enter the size of the array: 5
Enter 5 array values
12 3 9 0 -4
Smallest number in the array is -4
```

---

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

Program -

```
#include <stdio.h>

int sort(int[],int);
int main()
{
    int n , i;

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

    int a[n];

    printf("Enter %d array values\n",n);
    for(i = 0 ; i < n ; i++)
       scanf("%d",&a[i]);

    a[n] = sort(a,n);

    printf("Sorted array is\n");
    for(i = 0 ; i < n ; i++)
        printf("%d ",a[i]);


    return 0;
}

int sort(int b[] , int N)
{
    int x , y , s;
    for(x = 0 ; x < N ; x++)
    {
        for(y = x+1 ; y < N ; y++)
        {
            if(b[x] > b[y])
            {
                s = b[y];
                b[y] = b[x];
                b[x] = s;
            }
        }
    }
    return b;
}
```

Output -

```
Enter the size of the array: 5
Enter 5 array values
1 9 6 4 3
Sorted array is
1 3 4 6 9
```

---

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])

Program -

```
#include<stdio.h>

void rotate(int[],int,int);

int main()
{
    int N , i , n;

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

    int a[N];
    printf("Enter %d array values\n",N);
    for(i = 0 ; i < N ; i++)
        scanf("%d",&a[i]);

    printf("Enter the number of positions to rotate: ");
    scanf("%d",&n);

    rotate(a , N , n);
    printf("Resulting array after left rotation %d times is\n",n);
    for(i = 0 ; i < N ; i++)
    printf("%d ",a[i]);

    return 0;
}

void rotate(int b[] , int size , int pos)
{
    int s , i , j , x = size - 1 ;
    while(pos)
    {
        s = b[0] ;
        for(i = 0 ; i < size; i++)
        {
            j =  i + 1;
            b[i] = b[j];
        }
     b[x] = s;
     pos--;
    }
}
```

Output -

```
Enter the size of the array: 5
Enter 5 array values
32 29 40 12 70
Enter the number of positions to rotate: 2
Resulting array after left rotation 2 times is
40 12 70 32 29
```

---

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.

Program -

```
#include<stdio.h>

int findDuplicate(int[],int);

int main()
{
    int size , i;
    printf("Enter the size of the array: ");
    scanf("%d",&size);

    int a[size];
    printf("Enter %d array values\n",size);
    for(i = 0 ; i < size ; i++)
        scanf("%d",&a[i]);

    int Dup = findDuplicate(a,size);
    if(Dup)
        printf("First adjacent duplicate value is %d",Dup);
    else
        printf("No adjacent duplicate values exist");

    return 0;
}

int findDuplicate(int arr[] , int N)
{
    int j , k , duplicate = 0;
    for(j = 0 ; j < N ; j++)
    {
        k = j + 1;
        if((k < N) && (arr[j] == arr[k]))
        {
            duplicate = arr[j];
            break;
        }
    }
    return duplicate;
}
```

Output -

```
Enter the size of the array: 5
Enter 5 array values
1 2 2 3 3
First adjacent duplicate value is 2
```

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

Program -

```
#include<stdio.h>

void displayReverse(int[],int);

int main()
{
    int size , i;

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

    int a[size];
    printf("Enter %d array values\n",size);
    for(i = 0 ; i < size ; i++)
        scanf("%d",&a[i]);

    printf("Array in Reverse order is\n");
    displayReverse(a,size);

    return 0;
}

void displayReverse(int arr[],int N)
{
    int k;

    for(k = N-1 ; k >= 0 ; k--)
    {
        printf("%d ",arr[k]);
    }
}
```

Output -

```
Enter the size of the array: 5
Enter 5 array values
1 2 3 4 5
Array in Reverse order is
5 4 3 2 1
```

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

Program -

```
#include<stdio.h>

int main()
{
    int N , i , j , count = 0;
    printf("Enter the size of the array: ");
    scanf("%d",&N);

    int a[N];
    printf("Enter %d elements:-\n",N);
    for(i = 0 ; i < N ; i++)
    {
        printf("Enter element %d: ",i+1);
        scanf("%d",&a[i]);
    }

    for(i = 0 ; i < N ; i++)
    {
         for(j = i+1 ; j < N ; j++)
         {
              if(a[i] == a[j])
              {
                   count++;
                   break;
              }
         }
    }
    printf("Total number of duplicate elements = %d",count);
    return 0;
}
```

Output -

```
Enter the size of the array: 5
Enter 5 elements:-
Enter element 1: 2
Enter element 2: 2
Enter element 3: 2
Enter element 4: 3
Enter element 5: 3
Total number of duplicate elements = 3
```

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

Program -

```
#include <stdio.h>

void printUnique(int[],int);

int main()
{
    int N , i;

    printf("Enter number of elements to be stored in the array: ");
    scanf("%d",&N);

    int a[N];
    printf("Enter %d elements:-\n",N);
    for(i = 0 ; i < N ; i++) {
        printf("Enter element - %d: ",i+1);
        scanf("%d",&a[i]);
    }

    printf("\nAll unique elements of the array are:-\n");
    printUnique(a,N);

    return 0;
}

void printUnique(int arr[] , int n)
{
    int i , j , k , flag;

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

Output -

```
Enter number of elements to be stored in the array: 5
Enter 5 elements:-
Enter element - 1: 1
Enter element - 2: 1
Enter element - 3: 8
Enter element - 4: 3
Enter element - 5: 2

All unique elements of the array are:-
8 3 2
```

---

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

Program -

```
#include<stdio.h>

void merge(int[],int[],int);
int main()
{
    int N , i , j , k;

    printf("Enter the size for both the arrays: ");
    scanf("%d",&N);

    int a[N] , b[N];
    printf("Enter 1st array elements:-\n");
    for(i = 0 ; i < N ; i++)
    {
        printf("Element - %d: ",i+1);
        scanf("%d",&a[i]);
    }

    printf("\nEnter 2nd array elements:-\n");
    for(i = 0 ; i < N ; i++)
    {
        printf("Element - %d: ",i+1);
        scanf("%d",&b[i]);
    }

    merge(a,b,N);
    return 0;
}

void merge(int x[] , int y[] , int size)
{
    int n = size * 2;
    int merge[n] , i , j , k = 0;


    for(i = 0 ; i < n/2 ; i++)
    {
        merge[i] = x[i];
    }
    for(i = n/2 ; i < n ; i++,k++)
    {
        merge[i] = y[k];
    }

    for(i = 0 ; i < n ; i++)
    {
        for(j = i+1 ; j < n ; j++)
        {
            if(merge[i] < merge[j])
            {
                merge[i] = merge[i] + merge[j];
                merge[j] = merge[i] - merge[j];
                merge[i] = merge[i] - merge[j];
            }
        }
    }
    printf("\nMerged array elements in descending order is:-\n");
    for(i = 0 ; i < n ; i++)
        printf("%d ",merge[i]);
}
```

Output -

```
Enter the size for both the arrays: 4
Enter 1st array elements:-
Element - 1: 1
Element - 2: 3
Element - 3: 6
Element - 4: 2

Enter 2nd array elements:-
```

```
Element - 1: 8
Element - 2: 9
Element - 3: 10
Element - 4: 4

Merged array elements in descending order is:-
10 9 8 6 4 3 2 1
```

---

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

Program -

```
#include<stdio.h>

void countF(int[],int);

int main()
{
    int N , i;

    printf("Enter the number of elements to store: ");
    scanf("%d",&N);

    int a[N];

    printf("Enter %d elements\n",N);
    for(i = 0 ; i < N ; i++)
        scanf("%d",&a[i]);

    countF(a,N);

    return 0;
}

void countF(int arr[],int size)
{
    int i , j , f;
    int G = -1e6;

    for(i = 0 ; i < size ; i++)
    {
        f = 1;
```

```
         for(j = 0 ; j < size ; j++)
         {
            if(i != j && arr[i] == arr[j] && arr[j] != G)
            {
                arr[j] = G;
                f++;
            }
        }
        if(arr[i] != G)
        {
            if(f == 1)
                printf("\nFrequency of %d = 1",arr[i]);
            else
                printf("\nFrequency of %d = %d",arr[i],f);
        }
    }
}
```

Output -

```
Enter the number of elements to store: 10
Enter 10 elements
1 2 2 1 3 4 4 3 1 0
Frequency of 1 = 3
Frequency of 2 = 2
Frequency of 3 = 2
Frequency of 4 = 2
Frequency of 0 = 1
```