ASSIGNMENT-15

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

#include<stdio.h>

int FindGreat (int b [], int n)

{

    int i, j, r;

    for (i=0; i<n; i++)

    {

        For (j=i+1; j<n; j++)

        {

            if(b[i]>b[j])

            {

                r = b[i];

                b[i] = b[j];

                b[j] = r;

            }

        }

    }

   return b[n-1];

}

int main ()

{

    int a [20], k, m, p;

    printf ("Enter Size of the array");

    scanf ("%d", &m);

    printf ("Enter %d Numbers:", m);

    for (k=0; k<m; k++)

    {

        scanf ("%d", &a[k]);

    }

    p= FindGreat (a, m);

    printf ("Greatest Number is: %d", p);

    return 0;

}

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

#include <stdio.h>

int FindSmallest (int b [], int m)

{

    int i, j, k;

    for (i=0; i<m; i++)

    {

        for (j=i+1; j<m; j++)

        {

            if (b[i]>b[j])

            {

                k = b[i];

                b[i] = b[j];

                b[j] = k;

            }

        }

    }

    return b [0];

}

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

void sortTheArr (int b [], int size)

{

    int m;

    for (int i=0; i<size; i++)

    {

        for (int j=i+1; j<size; j++)

        {

            if(b[i]>b[j])

                {

                    int temp;

                    temp = b[i];

                    b[i] = b[j];

                    b[j] = temp;

                }

        }

    }

    printf ("Sorted array is: ");

    for (int i=0; i<size; i++)

    {

        printf ("%d ", b[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 RotateLeft (int [], int, int);

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

void RotateLeft (int a [], int size, int d)

{

    while (d--)

    {

        int temp;

        temp = a [0];

        for (int i=0; i<size-1; i++)

        {

            a[i] = a[i+1];

        }

        a[size-1] = temp;

    }

    for (int i=0; i<= size-1; i++)

    {

        printf ("%d ", a[i]);

    }

}

void RotateRight (int b [], int size, int d)

{

    int temp;

    while (d--)

    {

        temp = b[size-1];

        for (int i= size-1; i>0; i--)

        {

            b[i] = b[i-1];

        }

        b [0] = temp;

    }

    for (int i=0; i<size; i++)

    {

        printf ("%d ", b[i]);

    }

}

int main ()

{

    int n, d, r;

    printf ("Enter size of array: ");

    scanf ("%d", &n);

    int arr[n];

    printf ("Enter %d elements: ", n);

    for (int i=0; i<n; i++)

    {

        scanf ("%d", &arr[i]);

    } // array input done

    printf ("Enter value of rotation: ");

    scanf ("%d", &d);

    printf ("Enter 0 for left and 1 for right: ");

    scanf ("%d", &r);

    if(r==0)

         RotateLeft (arr, n, d);

    if(r==1)

        RotateRight (arr, 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>

void OccOfAdjDuplicate (int [], int);

void OccOfAdjDuplicate (int a [], int n)

{

    int i;

    for (i=0; i<n; i++)

    {

        if (a[i] == a[i+1])

                break;

    }

    printf ("%d", a[i]);

}

int main ()

{

    int size;

    printf ("Enter size of array: ");

    scanf ("%d", &size);

    int arr[size];

    printf ("Enter %d elements: ", size);

    for (int i=0; i<size; i++)

    {

        scanf ("%d", &arr[i]);

    }

    OccOfAdjDuplicate (arr, size);

}

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 [], int);

void Reverse (int a[], int s)

{

    for (int i=s-1; i>=0; i--)

    {

        printf ("%d ", a[i]);

    }

}

int main ()

{

    int n;

    printf ("Enter Size of array: ");

    scanf ("%d", &n);

    int arr[n];

    printf ("Enter %d elements in the array: ");

    for (int i=0; i<n; i++)

    {

        Scanf ("%d", &arr[i]);

    }

    Reverse (arr, n);

    return 0;

}

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

#include<stdio.h>

void CountDuplicates (int [], int);

void CountDuplicates (int a [], int s)

{

    int hash [100] = {0};

    for (int i=0; i<s; i++)

    {

        hash[a[i]] ++;

    }

    for (int i=0; i<100; i++)

    {

        if (hash[i] > 1)

            printf ("No. of Dupli. of %d --> %d\n", i, hash[i]-1);

    }

}

int main ()

{

    int s;

    printf ("Enter Size of array: ");

    scanf ("%d", &s);

    int arr[s];

    printf ("Enter %d elements in the array: \n", s);

    for (int i=0; i<s; i++)

        scanf ("%d", &arr[i]);

    CountDuplicates (arr, s);

    return 0;

}

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

#include<stdio.h>

void Unique (int [], int);

void Unique (int arr [], int s)

{

    int hash [100] = {0};

    for (int i=0; i<s; i++)

    {

        hash [arr[i]] ++;

    }

    for (int i=0; i<100; i++)

    {

        if (hash[i] == 1)

            printf ("unique element is: %d\n", i);

    }

}

int main ()

{

    int n;

    printf ("Enter size of array: ");

    scanf ("%d", &n);

    int a[n];

    printf ("Enter %d elements: ", n);

    for (int i=0; i<n; i++)

    {

        scanf ("%d", &a[i]);

    }

    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>

#include<stdlib.h>

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

void MergeArrays (int a [], int b [], int s)

{

    int arr3[2\*s];

    for (int i=0; i<2\*s; i++)

    {

          if (i<s)

            arr3[i] = a[i];

          else

            arr3[i] = b[i-s];

    }

    /// Sorting array in descending order

    for (int i=0; i<(2\*s-1); i++)

    {

        for (int j=i+1; j<2\*s; j++)

        {

            if (arr3[i] < arr3[j])

            {

                int temp = arr3[i];

                arr3[i] = arr3[j];

                arr3[j] = temp;

            }

        }

    }

    // Printing the array 3

    printf ("Sorted array is: ");

    for (int i=0; i<2\*s; i++)

        printf ("%d ", arr3[i]);

}

int main ()

{

    int s;

    printf ("Enter Size of arrays: ");

    scanf ("%d", &s);

    int arr1[s], arr2[s];

    printf ("Enter elements of first array: ");

    for (int i=0; i<s; i++)

        scanf ("%d", &arr1[i]);

    printf ("Enter elements of second array: ");

    for (int i=0; i<s; i++)

        scanf ("%d", &arr2[i]);

    MergeArrays (arr1, arr2, s);

    return 0;

}

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

#include<stdio.h>

void CountFreq (int [], int);

void CountFreq (int a [], int s)

{

    int Freq [100] = {0};

    for (int i=0; i<s; i++)

    {

        Freq [a[i]] ++;

    }

    for (int i=0; i<100; i++)

    {

        if (Freq [i] > 0)

            printf ("%d --> %d \n", i, Freq[i]);

    }

}

int main ()

{

    int s;

    printf ("Enter size of array:");

    scanf ("%d", &s);

    int arr[s];

    printf ("Enter elements in array: ");

    for (int i=0; i<s; i++)

        scanf ("%d", &arr[i]);

    CountFreq (arr, s);

    return 0;

}