**Assignment-15 Solution Name : Om Pant**

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

Sol-

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

#include<stdio.h>

int Greatest(int a[], int size){

    int max,i;

    max = a[0];

    for(i=1;i<size;i++){

        if(a[i]>max){

            max = a[i];

        }

    }

    return max;

}

int main(){

    int num,i,greatest;

    printf("Enter array size\n");

    scanf("%d",&num);

    int arr[num];

    printf("Enter array elements\n");

    for(i=0;i<num;i++){

        scanf("%d",&arr[i]);

    }

    greatest = Greatest(arr,num);

    printf("Greatest Element in array is: %d\n",greatest);

    return 0;

}

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

Sol =

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

#include<stdio.h>

int smallest(int a[], int size){

    int min,i;

    min = a[0];

    for(i=1;i<size;i++){

        if(a[i]<min){

            min = a[i];

        }

    }

    return min;

}

int main(){

    int num,i,smallestElem;

    printf("Enter array size\n");

    scanf("%d",&num);

    int arr[num];

    printf("Enter array elements\n");

    for(i=0;i<num;i++){

        scanf("%d",&arr[i]);

    }

    smallestElem = smallest(arr,num);

    printf("smallest Element in array is: %d\n",smallestElem);

    return 0;

}

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

Sol-

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

#include<stdio.h>

void sortArray(int a[], int size){

    int temp,i,j;

    for(i=size-1;i>=0;i--){

        for(j=0;j<i;j++){

            if(a[j]>a[j+1]){

                temp = a[j];

                a[j] = a[j+1];

                a[j+1] = temp;

            }

        }

    }

    printf("Sorted Array: ");

    for(i=0;i<size;i++)

        printf("%d ",a[i]);

}

int main(){

    int num,i;

    printf("Enter array size\n");

    scanf("%d",&num);

    int arr[num];

    printf("Enter array elements\n");

    for(i=0;i<num;i++){

        scanf("%d",&arr[i]);

    }

    sortArray(arr , num);

    return 0;

}

1. 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] )

Sol-

// 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 a[], int length){

    int i,temp;

    temp = a[0];

    for(i=0;i<length;i++){

        a[i] = a[i+1];

    }

    a[length-1] = temp;

}

void rotateRight(int a[], int length){

    int i,temp;

    temp = a[length-1];

    for(i=length-1;i>0;i--){

        a[i] = a[i-1];

    }

    a[0] = temp;

}

int main(){

    int length,pos,dir;

    printf("Enter the lenght of array\n");

    scanf("%d",&length);

    int a[length];

    printf("Enter %d Elements\n",length);

    for(int i=0;i<length;i++){

        scanf("%d",&a[i]);

    }

    printf("Enter the Number of position to rotate\n");

    scanf("%d",&pos);

    printf("Enter Direction to Rotate \nPress 1 for - Left\nPress 2 for - Right\n");

    scanf("%d",&dir);

    switch (dir)

    {

    case 1:

        /\* Left \*/

        for(int i=0; i<pos;i++)

            rotateLeft(a,length);

        printf("Rotated array is ---\n");

        for(int j=0;j<length;j++){

            printf("%d ",a[j]);

        }

        break;

    case 2:

        /\* Right Rotation \*/

         for(int i=0; i<pos;i++)

            rotateRight(a,length);

        printf("Rotated array is ---\n");

        for(int j=0;j<length;j++){

            printf("%d ",a[j]);

        }

        break;

    default:

        printf("Please Enter Valid Choice");

        break;

    }

    return 0;

}

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

Sol-

// 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 adjacentDup(int a[],int length){

    for(int i=0;i<length-1;i++){

        if(a[i] == a[i+1])

            return a[i];

    }

    return 0;

}

int main(){

    int length,duplicate;

    printf("Enter length of array\n");

    scanf("%d",&length);

    int a[length];

    printf("Enter %d elements\n",length);

    for(int i=0;i<length;i++){

        scanf("%d",&a[i]);

    }

    duplicate = adjacentDup(a,length);

    if(duplicate != 0)

        printf("%d is adjacent Duplicate\n",duplicate);

    else

        printf("No Adjacent Duplicate Present\n");

    return 0;

}

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

Sol-

// 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 readAnDisplay(int a[], int length){

    printf("Enter %d elements\n",length);

    for(int i=0;i<length;i++){

        scanf("%d",&a[i]);

    }

    printf("Array Element In Reverse Order are\n");

    for(int i=length-1;i>=0;i--){

        printf("%d ",a[i]);

    }

}

int main(){

    int length;

    printf("Enter length of array\n");

    scanf("%d",&length);

    int a[length];

    readAnDisplay(a,length);

   return 0;

}

1. 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)

Sol-

// 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 countDuplicates(int a[],int length){

    int count = 0;

    for(int i=0;i<length-1;i++){

        if(a[i] == a[i+1] && a[i] != a[i-1])

            count +=1;

    }

    return count;

}

int main(){

    int length;

    printf("Enter length of array\n");

    scanf("%d",&length);

    int a[length];

    printf("Enter %d elements\n",length);

    for(int i=0;i<length;i++){

        scanf("%d",&a[i]);

    }

    printf("Duplicates in Array are: %d\n",countDuplicates(a,length));

    return 0;

}

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

Sol-

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

#include<stdio.h>

void PrintUnique(int a[], int length){

    int i,j,flag ;

    for(i = 0;i<length;i++){

        for(j=1,flag=0;j<length;j++){

            if(a[i] == a[j] && j != i){

                flag = 1;

                break;

            }

        }

        if(flag != 1){

            printf("%d ",a[i]);

        }

    }

}

int main(){

    int length;

    printf("Enter length of array\n");

    scanf("%d",&length);

    int a[length];

    printf("Enter %d elements\n",length);

    for(int i=0;i<length;i++){

        scanf("%d",&a[i]);

    }

    PrintUnique(a,length);

    return 0;

}

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

Sol-

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

#include<stdio.h>

void mergeArray(int a[], int b[], int length){

    int res[2\*length],i,j,k=0;

    //Copy a[] elements in res[]

    for(i=0;i<length;i++){

        res[k] = a[i];

        k++;

    }

    //Copy b[] elements in res[]

    for(i=0;i<length;i++){

        res[k] = b[i];

        k++;

    }

    //sorting in decreasing order

    int temp;

    for(i=0;i<2\*length;i++){

        for(j=i;j<2\*length;j++){

            if(res[i] < res[j]){

                temp = res[i];

                res[i] = res[j];

                res[j] = temp;

            }

        }

    }

    //printing Merged array

    for(int x=0;x <length\*2;x++){

        printf("%d ",res[x]);

    }

}

int main(){

    int length;

    printf("Enter length of array's\n");

    scanf("%d",&length);

    //Array Definition

    int a[length], b[length];

    printf("Enter 1st array Elements\n");

    for(int i=0;i<length;i++){

        scanf("%d",&a[i]);

    }

    printf("\nEnter 2nd array Elements\n");

    for(int i=0;i<length;i++){

        scanf("%d",&b[i]);

    }

    mergeArray(a , b, length);

    return 0;

}

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

Sol-

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

#include<stdio.h>

void count(int a[], int length){

    int i,greatest=-1;

    for(i=0;i<length;i++){

        if(a[i]>greatest){

            greatest = a[i];

        }

    }

    int arr[greatest+1];

    for(i=0;i<greatest+1;i++){

            arr[i] = 0;

        }

    for(i=0;i<length;i++){

        arr[a[i]] += 1;

    }

    for(i=0;i<greatest+1;i++){

        if(arr[i] != 0){

            printf("%d - %d Times\n",i,arr[i]);

        }

    }

}

int main(){

    int length;

    printf("Enter the size of array\n");

    scanf("%d",&length);

    int array[length];

    printf("Enter %d elements of array\n",length);

    for(int i=0;i<length;i++){

        scanf("%d",&array[i]);

    }

    // counting Frequnecy

    count(array , length);

    return 0;

}