ASSIGNMENT-7

NAME-SAUBHAGYA RANJAN ROUT

Q1) read n number of values in an array and display it in reverse order.

```
#include <stdio.h>
void main()
{
 int i,n,a[100];
printf("The number of elements to store in the array :\n");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
 printf("a[%d] : ",i);
 scanf("%d",&a[i]);
 }
 printf("\nThe values store into the array are : \n");
 for(i=0;i<n;i++)
 {
printf(" %d",a[i]);
}
printf("\n\nThe values store into the array in reverse are :\n");
 for(i=n-1;i>=0;i--)
 {
```

```
Printf("% d",a[i]);
}
}
```

OUTPUT

```
The number of elements to store in the array : 3
a[0] : 1
a[1] : 2
a[2] : 3
The values store into the array are :
1 2 3
The values store into the array in reverse are :

3 2 1
```

Q2) find the sum of all elements of the array.

```
#include <stdio.h>
void main()
{
 int a[30];
 int i, n, sum=0;
 printf("Input the number of elements:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
```

```
 printf("a[%d] : ",i);
 scanf("%d",&a[i]);
 }
 for(i=0; i<n; i++)
 {
 sum += a[i];
 }
 printf("Sum of all elements is: %d", sum);
}
```

OUTPUT

```
Input the number of elements:4

a[0] : 5

a[1] : 6

a[2] : 9

a[3] : 0

Sum of all elements is: 20
```

Q3) copy the elements of one array into another array.

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

```
void main()
{
 int a[50], b[60];
 int i, n;
 printf("Input the number of elements:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
 printf("a[%d] : ",i);
 scanf("%d",&a[i]);
 }
 for(i=0; i<n; i++)
 {
 b[i] = a[i];
 }
 printf("copied elements are:\n");
 for(i=0; i<n; i++)
 {
 printf("% d", b[i]);
 }
}
```

OUTPUT

```
Input the number of elements:5
```

a[0] : 6

a[1] : 8

a[2] : 3

a[3] : 1

a[4] : 2

copied elements are:

6 8 3 1 2

Q4) count a total number of duplicate elements in an array.

```
#include <stdio.h>
int main()
{
 int arr[6];
 int i, j, size, count = 0;
 printf("Enter array size : ");
 scanf("%d", &size);
 printf("Enter elements in array : ");
 for(i=0; i<size; i++)
 {
```

```
 scanf("%d", &arr[i]);
 }
 for(i=0; i<size; i++)
 {
 for(j=i+1; j<size; j++)
 {
 if(arr[i] == arr[j])
{
count++;
 break;
 }
 }
 }
 printf("\nTotal number of duplicate elements found in array = %d", count);
 return 0;
}
```

Output

```
Enter array size : 5
Enter elements in array : 2 3 5 5 7 7
Total number of duplicate elements found in array = 1
```

Q5) find the maximum and minimum element in an array.

```
#include <stdio.h>
void main()
```

```
{
 int arr[100];
 int i,max,min,n;
 printf("Number of elements :");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
 printf("a[%d] : ",i);
 scanf("%d",&arr[i]);
 }
 max = arr[0];
 min = arr[0];
 for(i=1; i<n; i++)
 {
 if(arr[i]>max)
 {
 max = arr[i];
 }
 if(arr[i]<min)
 {
 min = arr[i];
 }
 }
 printf("Maximum element is : %d\n", max);
```

```
 printf("Minimum element is : %d", min);
}
```

Output

```
Number of elements :4

a[0] : 5555555

a[1] : 89098

a[2] : 0

a[3] : 8

Maximum element is : 5555555

Minimum element is : 0
```

Q6) separate odd and even integers in separate arrays.

```
#include <stdio.h>
void main()
{
 int a[10],b[10],c[10];
 int i,j=0,k=0,n;
printf("Number of elements :");
```

```
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
printf("a[%d] : ",i);
 scanf("%d",&a[i]);
 }
 for(i=0;i<n;i++)
 {
if (a[i]%2 == 0)
{
 b[j] = a[i];
 j++;
}
else
{
 c[k] = a[i];
 k++;
}
 }
 printf("\nThe Even elements are : \n");
 for(i=0;i<j;i++)
 {
printf("%d ",b[i]);
 }
```

```
 printf("\nThe Odd elements are :\n");
 for(i=0;i<k;i++)
 {
printf("%d ", c[i]);
 }
}
```

Output

Number of elements :8

a[0] : 6

a[1] : 9

a[2] : 8

a[3] : 4

a[4] : 99

a[5] : 81

a[6] : 77

a[7] : 90

The Even elements are :

6 8 4 90

The Odd elements are :

9 99 81 77

Q7) insert New value in the array.

```
#include <stdio.h>
void main()
{
 int arr1[50],i,n,p,inval;
 printf("Input the size of array : ");
 scanf("%d", &n);
 for(i=0;i<n;i++)
 {
 printf("a[%d] : ",i);
 scanf("%d",&arr1[i]);
 }
 printf("Input the value to be inserted : ");
 scanf("%d",&inval);
 printf("The exist array list is :\n ");
 for(i=0;i<n;i++)
 printf(" %d",arr1[i]);
 for(i=0;i<n;i++)
 if(inval<arr1[i])
```

```
{
p = i;
break;
}
for(i=n;i>=p;i--)
arr1[i]= arr1[i-1];
arr1[p]=inval;

printf("\n\nAfter Insert the list is :\n ");
for(i=0;i<=n;i++)
printf(" %d",arr1[i]);
}
```

Output

```
Input the size of array : 4

a[0] : 8

a[1] : 7

a[2] : 9

a[3] : 89
Input the value to be inserted : 100
After Insert the list is :
```

100 8 7 9 89

Q8) delete an element at desired position from an array.

```
#include <stdio.h>
void main(){
 int arr1[50],i,pos,n;
 printf("Input the size of array : ");
 scanf("%d", &n);
 for(i=0;i<n;i++)
 {
 printf("a[%d] : ",i);
 scanf("%d",&arr1[i]);
 }
 printf("\nInput the position where to delete: ");
 scanf("%d",&pos);
 i=0;
 while(i!=pos-1)
 i++;
 while(i<n)
 {
 arr1[i]=arr1[i+1];
 i++;
 }
 n--;
```

```
 printf("\nThe new list is : ");
 for(i=0;i<n;i++)
 {
 printf(" %d",arr1[i]);
}
 printf("\n\n");
}
```

Output

```
Input the size of array : 3
a[0] : 8
a[1] : 9
a[2] : 0
Input the position where to delete: 2
The new list is : 8 0
```

Q9) find the second largest element in an array.

```
#include <stdio.h>
int main() {
 int array[10];
 int size, i, largest, second;
 printf("enter the size of array:");
 scanf("%d",&size);
 printf("the value stored in the array is:\n");
 for(i=0;i<size;i++){
```

```
 printf("a[%d]:",i);
 scanf("%d",&array[i]);
}
 if(array[0] > array[1]) {
 largest = array[0];
 second = array[1];
 } else {
 largest = array[1];
 second = array[0];
 }
 for(i=2;i<size;i++) {
 if(largest<array[i] ) {
 second = largest;
 largest = array[i];
 } else if( second < array[i] ) {
 second = array[i];
 }
 }
 printf("Largest - %d \nSecond - %d \n", largest, second);
 return 0;
}
```

OUTPUT

```
enter the size of array:3
```

the value stored in the array is:

a[0]5

a[1]4

a[2]8

Largest - 8

Second - 5

Q10) . find the median of two sorted arrays of same size.

```
#include <stdio.h>
int max(int a, int b)
{
 return ((a > b) ? a : b);
}
int min(int a, int b)
{
 return ((a < b) ? a : b);
}
int median(int arr[], int size)
{
 if (size % 2 == 0)
 return (arr[size/2] + arr[size/2-1])/2;
 else
```

```
 return arr[size/2];
}
int median2SortedArrays(int arr1[], int arr2[], int size)
{
 int med1;
 int med2;
 if(size <= 0) return -1;
 if(size == 1) return (arr1[0] + arr2[0])/2;
 if (size == 2) return (max(arr1[0], arr2[0]) + min(arr1[1], arr2[1])) / 2;
med1 = median(arr1, size);
 med2 = median(arr2, size);
if(med1 == med2) return med1;
if (med1 < med2)
 {
 return median2SortedArrays(arr1 + size/2, arr2, size - size/2);
 }
 else
 {
 return median2SortedArrays(arr2 + size/2, arr1, size - size/2);
 }
}
int main()
{
 int i,m,n;
```

```
 int arr1[] = {1, 5, 13, 24, 35};
 int arr2[] = {3, 8, 15, 17, 32};
 m = sizeof(arr1)
 n = sizeof(arr2)
printf("The given array - 1 is : ");
for(i = 0; i < m; i++)
{
printf("%d ", arr1[i]);
 }
printf("\n");
printf("The given array - 2 is : ");
for(i = 0; i < n; i++)
{
printf("%d ", arr2[i]);
 }
printf("\n");
 printf("\nThe Median of the 2 sorted arrays is: %d",median2SortedArrays(arr1, arr2, n));
 return 0;
}
```

OUTPUT

```
The given array - 1 is : 1 5 13 24 35
The given array - 2 is : 3 8 15 17 32
The Median of the 2 sorted arrays is: 14
```

11. multiplication of two square Matrices

```
#include <stdio.h>

#define N 4

void multiply(int mat1[][N], int mat2[][N], int res[][N])

{

 int i, j, k;

 for (i = 0; i < N; i++) {

 for (j = 0; j < N; j++) {

 res[i][j] = 0;

 for (k = 0; k < N; k++)

 res[i][j] += mat1[i][k] * mat2[k][j];

 }

 }

}

int main()

{

 int mat1[N][N] = { { 1, 1, 1, 1 },

 { 2, 2, 2, 2 },

 { 3, 3, 3, 3 },

 { 4, 4, 4, 4 } };

 int mat2[N][N] = { { 1, 1, 1, 1 },

 { 2, 2, 2, 2 },

 { 3, 3, 3, 3 },

 { 4, 4, 4, 4 } };
```

```
int res[N][N]; // To store result
int i, j;
multiply(mat1, mat2, res);
printf("Result matrix is \n");
for (i = 0; i < N; i++) {
for (j = 0; j < N; j++)
printf("%d ", res[i][j]);
printf("\n");
}
return 0;
}
```

OUTPUT

Result matrix is

10 10 10 10

20 20 20 20

30 30 30 30

40 40 40 40

12. find transpose of a given matrix.

```
#include <stdio.h>
void main()
```

```
{
int arr1[50][50],brr1[50][50],i,j,r,c;
printf("\nInput the rows and columns of the matrix : ");
scanf("%d %d",&r,&c);
printf("Input elements in the first matrix :\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
printf("element - [%d],[%d] : ",i,j);
scanf("%d",&arr1[i][j]);
}
}
printf("\nThe matrix is :\n");
for(i=0;i<r;i++)
{
printf("\n");
for(j=0;j<c;j++)
printf("%d\t",arr1[i][j]);
}
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
```

```
{
brr1[j][i]=arr1[i][j];
}
}
printf("\n\nThe transpose of a matrix is : ");
for(i=0;i<c;i++){
printf("\n");
for(j=0;j<r;j++){
printf("%d\t",brr1[i][j]);
}
}
}
```

OUTPUT

```
Input the rows and columns of the matrix : 2 3

Input elements in the first matrix :

element - [0],[0] : 1

element - [0],[1] : 2

element - [0],[2] : 3

element - [1],[0] : 4
```

element - [1],[1] : 5

element - [1],[2] : 6

The matrix is :

1 2 3

4 5 6

The transpose of a matrix is :

1 4

2 5

3 6

13. find the sum of left diagonals of a matrix.

```
#include <stdio.h>
void main()
 {
 int i,j,arr1[50][50],sum=0,n,m=0;
printf("Input the size of the square matrix : ");
 scanf("%d", &n);
 m=n;
printf("Input elements in the first matrix :\n");
 for(i=0;i<n;i++)
```

```
 {
 for(j=0;j<n;j++)
 {
 printf("element - [%d],[%d] : ",i,j);
 scanf("%d",&arr1[i][j]);
 }
 }
printf("The matrix is :\n");
for(i=0;i<n;i++)
{
 for(j=0;j<n ;j++)
 printf("% 4d",arr1[i][j]);
 printf("\n");
}
for(i=0;i<n;i++)
{
 m=m-1;
 for(j=0;j<n ;j++)
 {
 if (j==m)
 {
 sum= sum+arr1[i][j];
 }
```

```
 }
}
 printf("Addition of the left Diagonal elements is :%d\n",sum);
 }
```

OUTPUT

```
Input the size of the square matrix : 2

Input elements in the first matrix :

element - [0],[0] : 2

element - [0],[1] : 5

element - [1],[0] : 8

element - [1],[1] : 9

The matrix is :
2 5

8 9

Addition of the left Diagonal elements is :13
```

14. check whether a given matrix is an identity matrix.

```
#include <stdio.h>
void main()
{
 int a[10][10];
 int i, j, row, column, count = 1;
 printf("Enter the order of the matrix A \n");
 scanf("%d %d", &row, &column);
 printf("Enter the elements of matrix A \n");
 for (i = 0; i < row; i++)
 {
 for (j = 0; j < column; j++)
 {
 scanf("%d", &a[i][j]);
 }
 }
 printf("MATRIX A is \n");
 for (i = 0; i < row; i++)
 {
 for (j = 0; j < column; j++)
 {
 printf(" %d", a[i][j]);
 }
 printf("\n");
```

```
}
for (i = 0; i < row; i++)
{
for (j = 0; j < column; j++)
{
if (a[i][j] != 1 && a[j][i] != 0)
{
count = 0;
break;
}
}
}
if (count== 1 )
printf("It is identity matrix \n");
else
printf("It is not a identity matrix \n");
}
```

OUTPUT

```
Enter the order of the matrix A

2

2
```

Enter the elements of matrix A

1

0

0

MATRIX A is

1 0

0 1

It is identity matrix

15. search an element in a row wise and column wise sorted matrix.

```
#include <stdio.h>
int searchElement(int arr2D[4][4], int n, int x)
{
 int i = 0, j = n-1;
 while ( i < n && j >= 0 )
 {
 if ( arr2D[i][j] == x )
 {
 printf("\nThe element Found at the position in the matrix is: %d, %d", i, j);
```

```
 return 1;
 }
 if ( arr2D[i][j] < x )
 j--;
 else
 i++;
 }
 printf("\nThe given element not found in the 2D array.");
 return 0;
}
int main()
{
 int arr2D[4][4] = { {15, 23, 31, 39},
 {18, 26, 36, 43},
 {25, 28, 37, 48},
 {30, 34, 39, 50},
 };
int i,j,v;
v=51;
printf("The given array in matrix form is : \n");
for(i = 0; i < 4; i++)
{
for (j=0;j<4;j++)
{
```

```
printf("%d ", arr2D[i][j]);
 }
printf("\n");
}
 printf("The given value for searching is: %d",v);
 searchElement(arr2D, 4, v);
 return 0;
}
```

OUTPUT

```
The given array in matrix form is :

15 23 31 39

18 26 36 43

25 28 37 48

30 34 39 50

The given value for searching is: 51

The given element not found in the 2D array.
```