ASSIGNMENT-7

1. Read n number of values in an array and display it in reverse order.

PROGRAM:

```
#include<stdio.h>
void main()
{
   int n, i;
   printf("Enter the value of n: ");
   scanf("%d", &n);
   int a[n];

   for(i=0; i<n; i++)
   {
      printf("Enter a number: ");
      scanf("%d", &a[i]);
   }
   printf("\n");
   printf("The numbers in reverse order are:\n");
   for(i=n-1; i>=0; i--)
   {
      printf("%d\n", a[i]);
   }
}
```

OUTPUT:

```
Enter the value of n: 5
Enter a number: 2
Enter a number: 4
```

Enter a number: 6

Enter a number: 8

Enter a number: 10

The numbers in reverse order are:

10

8

6

4

2

2. Find the sum of all elements of the array.

PROGRAM:

```
#include<stdio.h>
void main()
{
   int a[5], i, sum=0;

   for(i=0; i<5; i++)
   {
      printf("Enter a number: ");
      scanf("%d", &a[i]);
   }
   for(i=0; i<5; i++)
   {
      sum = sum + a[i];
   }
```

```
    printf("The sum is: %d", sum);
}
```

OUTPUT:

```
Enter a number: 5
Enter a number: 10
Enter a number: 15
Enter a number: 20
Enter a number: 25
The sum is: 75
```

3. Copy the elements of one array into another array.

PROGRAM:

```
#include<stdio.h>
void main()
{
    int a[5], b[5], i;

    for(i=0; i<5; i++)
    {
        printf("Enter a number: ");
        scanf("%d", &a[i]);
    }

    printf("The elements after coping are:\n");
    for(i=0; i<5; i++)
    {
        b[i] = a[i];
```

```
        printf("%d\n", b[i]);
    }
}
```

OUTPUT:

```
Enter a number: 2
Enter a number: 4
Enter a number: 6
Enter a number: 8
Enter a number: 10
The elements after coping are:
2
4
6
8
10
```

4. Count a total number of duplicate elements in an array.

PROGRAM:

```
#include<stdio.h>
void main()
{
    int a[10], i, j, c=0;

    for(i=0; i<10; i++)
    {
        printf("Enter a number: ");
        scanf("%d", &a[i]);
```

```
    }
    for(i=0; i<10; i++)
    {
        for(j=i+1; j<10; j++)
        {
            if(a[i] == a[j])
            {
                c = c + 1;
                break;
            }
        }
    }
    printf("Total number of duplicate elements in the array are: %d\n", c);
}
```

OUTPUT:

```
Enter a number: 2
Enter a number: 10
Enter a number: 20
Enter a number: 40
Enter a number: 2
Enter a number: 10
Enter a number: 50
Enter a number: 20
Enter a number: 40
Enter a number: 65
Total number of duplicate elements in the array are: 4
```

5. Find the maximum and minimum element in an array.

PROGRAM:

```
#include<stdio.h>
void main()
{
   int a[5], i, max, min;

   for(i=0; i<5; i++)
   {
      printf("Enter a number: ");
      scanf("%d", &a[i]);
   }

   max = a[0];
   min = a[0];

   for(i=1; i<5; i++)
   {
      if(max > a[i])
      max = a[i];

      if(min < a[i])
      min = a[i];
   }
```

```
    printf("The maximum element in the array is:%d\n", max);
    printf("The minimum element in the array is:%d\n", min)     }
```

OUTPUT:

```
Enter a number: 2
Enter a number: 4
Enter a number: 6
Enter a number: 8
Enter a number: 10
The maximum element in the array is:2
The minimum element in the array is:10
```

6. Separate odd and even integers in separate arrays.

PROGRAM:

```
#include <stdio.h>
void main()
{
   int n,i,j=0,k=0,c=0;
   printf("Enter size of array: ");
   scanf("%d",&n);
   int arr[n],odd[n],even[n];

    for(i=0;i<n;i++)
    {
       printf("Input number in array: ");
       scanf("%d",&arr[i]);
    }
```

```
    for(i=0;i<n;i++)
    {
        if(arr[i] % 2 == 0)
        {
            even[j] = arr[i];
            j ++;
            c ++;
        }
        else
        {
            odd[k] = arr[i];
            k++;
        }
    }

        printf("Even numbers: ");
        for(i=0;i<c;i++)
        {
            printf("%d ",even[i]);
        }

        printf("Odd numbers: ");
        for(i=0;i<n-c;i++)
        {
            printf("%d ",odd[i]);
        }
}
```

OUTPUT:

Enter size of array: 5

Input number in array: 2

Input number in array: 3

Input number in array: 5

Input number in array: 4

Input number in array: 1

Even numbers: 2 4

Odd numbers: 3 5 1

7. Insert New value in the array.

PROGRAM:

```
#include <stdio.h>

int main()
{
  int location, i, n, value;
  printf("Enter number of elements in array\n");
  scanf("%d", &n);
  int arr[n];
  for (i = 0; i < n; i++)
  {
    printf("Enter a number: ");
    scanf("%d", &arr[i]);
  }

  printf("Enter the location where you wish to insert an element\n");
```

```
  scanf("%d", &location);

  printf("Enter the value to insert\n");
  scanf("%d", &value);

  for (i = n - 1; i >= location - 1; i--)
    arr[i+1] = arr[i];

  arr[location-1] = value;

  printf("Resultant array is\n");

  for (i = 0; i <= n; i++)
    printf("%d\n", arr[i]);

  return 0;
}
```

OUTPUT:

```
Enter number of elements in array
5
Enter a number: 2
Enter a number: 4
Enter a number: 6
Enter a number: 8
Enter a number: 10
Enter the location where you wish to insert an element
4
```

Enter the value to insert

9

Resultant array is

2

4

6

9

8

10

8. Delete an element at desired position from an array.

PROGRAM:

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

  printf("Enter number of elements in array\n");
  scanf("%d", &n);
  int arr[n];
  for (i= 0; i < n; i++)
  {
    printf("Enter a number: ");
    scanf("%d", &arr[i]);
  }

  printf("Enter the location where you wish to delete element\n");
```

```
  scanf("%d", &position);

  if (position >= n+1)
    printf("Deletion not possible.\n");
  else
  {
    for (i = position - 1; i < n - 1; i++)
      arr[i] = arr[i+1];

    printf("Resultant array:\n");

    for (i = 0; i < n - 1; i++)
      printf("%d\n", arr[i]);
  }

  return 0;
}
```

OUTPUT:

```
Enter number of elements in array
5
Enter a number: 2
Enter a number: 4
Enter a number: 6
Enter a number: 8
Enter a number: 10
Enter the location where you wish to delete element
4
```

Resultant array:

2

4

6

10

9. Find the second largest element in an array.

PROGRAM:

```
#include <stdio.h>
int main(){
int n,i,j,temp=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++){
   printf("Input number in array :: ");
   scanf("%d",&arr[i]);
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      if(arr[i] < arr[j]){
         temp = arr[i];
         arr[i] = arr[j];
         arr[j] = temp;
      }
   }
```

```
}
printf("Second largest value = %d",arr[n-2]);
return 0;
}
```

OUTPUT:

```
enter size of array :: 6
Input number in array :: 3
Input number in array :: 4
Input number in array :: 8
Input number in array :: 9
Input number in array :: 7
Input number in array :: 2
Second largest value = 8
```

10.Multiplication of two square Matrices.

PROGRAM:

```
#include <stdio.h>
int main(){
int n,i=0,j=0,k;
printf("enter size of array :: ");
scanf("%d",&n);
int arr1[n][n],arr2[n][n],mul[n][n];

printf("Enter elements in first array -->\n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("Enter a number :: ");
```

```
      scanf("%d",&arr1[i][j]);

   }
}

printf("Enter elements in second array -->\n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      printf("Enter a number :: ");
      scanf("%d",&arr2[i][j]);

   }
}

for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      mul[i][j]=0;
      for(k=0;k<n;k++){
      mul[i][j] += arr1[i][k] * arr2[k][j] ;
      }

   }
}
printf("Multiplication of the given matrix --> \n");
for(i=0;i<n;i++){
   for(j=0;j<n;j++){
      printf("%d ",mul[i][j]);
```

```
    }
    printf("\n");
}
return 0;
}
```

OUTPUT:

```
enter size of array :: 2
Enter elements in first array -->
Enter a number :: 4
Enter a number :: 2
Enter a number :: 2
Enter a number :: 4
Enter elements in second array -->
Enter a number :: 2
Enter a number :: 2
Enter a number :: 2
Enter a number :: 2
Multiplication of the given matrix -->
12 12
    12 2
```

12.Find transpose of a given matrix.

PROGRAM:

```
#include <stdio.h>
int main(){
int n,i=0,j=0,k;
```

```
printf("enter size of array :: ");
scanf("%d",&n);
int arr1[n][n],arr2[n][n];

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("Enter a number :: ");
       scanf("%d",&arr1[i][j]);

    }
}

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        arr2[i][j] = arr1[j][i];
    }
}

printf("Inserted matrix -->\n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("%d ",arr1[i][j]);
      }
      printf("\n");
}

printf("Transpose of this given matrix is -->\n");
```

```
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("%d ",arr2[i][j]);
      }
      printf("\n");
}
return 0;
}
```

OUTPUT:

```
enter size of array :: 3
Enter a number :: 1
Enter a number :: 2
Enter a number :: 3
Enter a number :: 4
Enter a number :: 5
Enter a number :: 6
Enter a number :: 7
Enter a number :: 8
Enter a number :: 9
Inserted matrix -->
1 2 3
4 5 6
7 8 9
Transpose of this given matrix is -->
1 4 7
2 5 8
```

3 6 9

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

PROGRAM:

```
#include <stdio.h>
int main(){
int n,i,j,sum=0;
printf("enter size of array :: ");
scanf("%d",&n);
int arr[n][n];

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("Enter a number :: ");
       scanf("%d",&arr[i][j]);

    }
}
printf("Inserted matrix -->\n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("%d ",arr[i][j]);
       }
       printf("\n");
}

printf("sum of left diagonals ");
```

```
for(i=0;i<n;i++){
    j=i;
    printf("%d ",arr[i][j]);
}
printf("is :: ");

for(i=0;i<n;i++){
    j=i;
    sum += arr[i][j];
}
printf("%d",sum);

return 0;
}
```

OUTPUT:

```
enter size of array :: 3
Enter a number :: 1
Enter a number :: 2
Enter a number :: 3
Enter a number :: 4
Enter a number :: 5
Enter a number :: 6
Enter a number :: 7
Enter a number :: 8
Enter a number :: 9
Inserted matrix -->
1 2 3
```

4 5 6

7 8 9

sum of left diagonals 1 5 9 is :: 15

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

PROGRAM:

```
#include<stdio.h>
int main()
{
   int i, j, rows, columns, a[10][10], Flag = 1;

   printf("\n enter the Number of rows and columns  :  ");
   scanf("%d %d", &i, &j);

   printf("\n enter the Matrix Elements \n");
   for(rows = 0; rows < i; rows++)
   {
      for(columns = 0; columns < j; columns++)
      {
         scanf("%d", &a[rows][columns]);
      }
   }

   for(rows = 0; rows < i; rows++)
   {
      for(columns = 0; columns < j; columns++)
      {
```

```
            if(a[rows][columns] != 1 && a[columns][rows] != 0)
            {
               Flag = 0;
               break;
            }
         }
      }
      if(Flag == 1)
      {
         printf("\n the matrix that you entered is an Identity Matrix ");
      }
      else
      {
         printf("\n the matrix that you entered is Not an Identity Matrix ");
      }

      return 0;
}
```

OUTPUT:

Output:-

enter the Number of rows and columns : 2

2

enter the Matrix Elements

1

2

3

4

the matrix that you entered is Not an Identity Matrix

enter the Number of rows and columns : 2

2

enter the Matrix Elements

1

0

0

1

the matrix that you entered is an Identity Matrix