ASSIGNMENT-7

1.**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("Input the number of elements of the array :");
  scanf("%d",&n);

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

  printf("\nThe  values 6of the array are : \n");
  for(i=0;i<n;i++)
   {
           printf("% 5d",a[i]);
          }

  printf("\n\nThe values of the array in reverse are :\n");
  for(i=n-1;i>=0;i--)
    {
           printf("% 5d",a[i]);
           }
  printf("\n\n");
}
```

OUTPUT-

```
Input the number of elements of the array :6
Input 6 number of elements in the array :
element - 0 : 7
element - 1 : 9
element - 2 : 76
element - 3 : 87
element - 4 : 67
element - 5 : 55

The  values of the array are :
    7    9   76   87   67   55

The values of the array in reverse are :
   55   67   87   76    9    7
```

2. **find the sum of all elements of the array.**

```
            #include <stdio.h>


int main()
{
   int a[1000],i,n,sum=0;

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

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


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

      sum+=a[i];
   }
    printf("sum of array is : %d",sum);

   return 0;
}
```

OUTPUT-

```
Enter size of the array : 4
Enter elements in array : 4 5 6 7
sum of array is : 22
```

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

```
#include <stdio.h>

int main()
{
   int source[100], dest[100];
   int i, size;
   printf("Enter the size of the array : ");
   scanf("%d", &size);
   printf("Enter elements of source array : ");
   for(i=0; i<size; i++)
   {
      scanf("%d", &source[i]);
   }

   for(i=0; i<size; i++)
   {
      dest[i] = source[i];
   }
   printf("\nElements of source array are : ");
   for(i=0; i<size; i++)
   {
      printf("%d\t", source[i]);
   }
   printf("\nElements of dest array are : ");
   for(i=0; i<size; i++)
   {
      printf("%d\t", dest[i]);
   }

   return 0;
}
```

OUTPUT-

```
Enter the size of the array : 4
Enter elements of  array : 5
7
8
88

Elements of source array are : 5       7       8       88
Elements of dest array are : 5  7       8       88
```

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

```
#include <stdio.h>

int main()
{
    int arr[10], i, j, Size, Count = 0;

    printf("\n Please Enter Number of  an array  :   ");
    scanf("%d", &Size);

    printf("\n Please Enter %d elements of an Array  :  ", Size);
    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("\n Total Number of Duplicate Elements in this Array  =  %d ", Count);

    return 0;
}
```

OUTPUT-

```
Please Enter Number of  an array  :   5

 Please Enter 5 elements of an Array  :  19
19
34
34
34

 Total Number of Duplicate Elements in this Array  =  3
```

5. find the maximum and minimum element in an array

```
#include <stdio.h>
#define MAX_SIZE 100
int main()
{
    int arr[MAX_SIZE];
    int i, max, min, size;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements in the array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    max = arr[0];
    min = arr[0];
    for(i=1; i<size; i++)
    {
        if(arr[i] > max)
        {
            max = arr[i];
        }
        if(arr[i] < min)
        {
            min = arr[i];
        }
    }
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);
```

```
   return 0;
}
```

OUTPUT-

```
Enter size of the array: 7
Enter elements in the array: 56
67
678
555
789
34
23
Maximum element = 789
Minimum element = 23
```

6. separate odd and even integers in separate arrays

```
#include <stdio.h>
int main()
{
   int arr[100],i,num;
   printf("Enter size of the array\n");
   scanf("%d",&num);
   printf("Enter the elements of the array\n");
   for(i=0; i<num; i++){
      scanf("%d",&arr[i]);
   }
   printf("\nEven numbers of the array are \n");
    for(i=0; i<num; i++){
      if(arr[i]%2==0){
         printf("%d \t",arr[i]);
      }
   }
   printf("\nOdd numbers of the array are \n");
    for(i=0; i<=num; i++){
      if (arr[i]%2==1){
         printf("%d \t",arr[i]);
      }
   }
   return 0;
}
```

OUTPUT-

```
Enter size of the array
6
Enter the elements of the array
56
67
78
45
76
45
```

```
Even numbers of the array are
56      78      76
Odd numbers of the array are
67      45      45
```

7. insert New value in the array.

```
#include <stdio.h>

int main()
{
   int array[50], position, c, n, value;

   printf("Enter  elements in the array\n");
   scanf("%d", &n);

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

   for (c = 0; c < n; c++)
   scanf("%d", &array[c]);

   printf("Please enter the location where you want to insert an new element\n");
   scanf("%d", &position);

   printf("Please enter the value\n");
   scanf("%d", &value);

   for (c = n - 1; c >= position - 1; c--)
array[c+1] = array[c];

   array[position-1] = value;

   printf("final array is\n");

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

   return 0;
}
```

OUTPUT-

```
Enter  elements in the array
5
Enter 5 elements
6
7
5
7
5
Please enter the location where you want to insert an new element
```

```
3
Please enter the value
44
final array is
6
7
44
5
7
5
```

8. delete an element at desired position from an array

```
#include <stdio.h>
int main()
{
  int array[100], position, c, n;

  printf("Enter number of elements in array\n");
  scanf("%d", &n);

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

  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);

  printf("Enter the location where the delete element\n");
  scanf("%d", &position);

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

    printf("Resultant array:\n");

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

  return 0;
}
```

```
OUTPUT-Enter number of elements in array
5
Enter 5 elements
5
6
7
```

```
8
9
Enter the location where the delete element
3
Resultant array:
5
6
8
9
```

9. find the second largest element in an array.

```
#include <stdio.h>
#include <limits.h>
#define MAX_SIZE 1000
int main()
{
    int arr[MAX_SIZE], size, i;
    int max1, max2;
    printf("Enter size of the array : ");
    scanf("%d", &size);
    printf("Enter elements in the array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    max1 = max2 = INT_MIN;
    for(i=0; i<size; i++)
    {
        if(arr[i] > max1)
        {
            max2 = max1;
            max1 = arr[i];
        }
        else if(arr[i] > max2 && arr[i] < max1)
        {
            max2 = arr[i];
        }
    }

    printf("First largest = %d\n", max1);
    printf("Second largest = %d", max2);
    return 0;
}
```

OUTPUT-

```
Enter size of the array : 5
Enter elements in the array: 4
34
34
55
```

```
23
First largest = 55
Second largest = 34
```

10. find the median of two sorted arrays of same size.

```
#include <stdio.h>
int getMedian(int ar1[], int ar2[], int n)
{
   int i = 0;
   int j = 0;
   int count;
   int m1 = -1, m2 = -1;
   for (count = 0; count <= n; count++)
   {
      if (i == n)
      {
         m1 = m2;
         m2 = ar2[0];
         break;
      }
      else if (j == n)
      {
         m1 = m2;
         m2 = ar1[0];
         break;
      }
      if (ar1[i] <= ar2[j])
      {
         m1 = m2;
         m2 = ar1[i];
         i++;
      }
      else
      {
         m1 = m2;
         m2 = ar2[j];
         j++;
      }
   }

   return (m1 + m2)/2;
}
int main()
{
   int ar1[] = {2,45,16,34,67};
   int ar2[] = {3,34,54,65,78};

   int n1 = sizeof(ar1)/sizeof(ar1[0]);
   int n2 = sizeof(ar2)/sizeof(ar2[0]);
```

```
  if (n1 == n2)
    printf("Median is %d", getMedian(ar1, ar2, n1));
  else
    printf("Doesn't work for arrays of unequal size");
  getchar();
  return 0;
}
```

OUTPUT-Median is 25

11. multiplication of two square Matrices

```
#include <stdio.h>

int main()
{
 int m, n, p, q, c, d, k, sum = 0;
 int first[10][10], second[10][10], multiply[10][10];

 printf("Enter number of rows and columns of first matrix\n");
 scanf("%d%d", &m, &n);
 printf("Enter elements of first matrix\n");

 for (c = 0; c < m; c++)
  for (d = 0; d < n; d++)
   scanf("%d", &first[c][d]);

 printf("Enter number of rows and columns of second matrix\n");
 scanf("%d%d", &p, &q);

 if (n != p)
  printf("The multiplication isn't possible.\n");
 else
 {
  printf("Enter elements of second matrix\n");

  for (c = 0; c < p; c++)
   for (d = 0; d < q; d++)
    scanf("%d", &second[c][d]);

  for (c = 0; c < m; c++) {
   for (d = 0; d < q; d++) {
    for (k = 0; k < p; k++) {
     sum = sum + first[c][k]*second[k][d];
    }

    multiply[c][d] = sum;
    sum = 0;
   }
  }
```

```
    printf(" the matrices is :\n");

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++)
        printf("%d\t", multiply[c][d]);

      printf("\n");
    }
  }

  return 0;
}
```

OUTPUT-

```
Enter number of rows and columns of first matrix
3
3
Enter elements of first matrix
2 3 4
4 5 6
1 9 5
Enter number of rows and columns of second matrix
3
3
Enter elements of second matrix
4 6 7
3 6 8
9 9 6
 the matrices is :
53      66      62
85      108     104
76      105     109
```

12. find transpose of a given matrix.

```
#include <stdio.h>

int main()
{
  int m, n, c, d, matrix[10][10], transpose[10][10];

  printf("Enter the  rows and columns of a matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter elements of the matrix\n");

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
      scanf("%d", &matrix[c][d]);

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
```

```
   transpose[d][c] = matrix[c][d];

 printf("Transpose of the matrix:\n");

 for (c = 0; c < n; c++) {
  for (d = 0; d < m; d++)
   printf("%d\t", transpose[c][d]);
  printf("\n");
 }

 return 0;
}
```

```
OUTPUT-Enter the  rows and columns of a matrix
3
3
Enter elements of the matrix
2 3 4
5 6 7
8 9 3
Transpose of the matrix:
2	5	8
3	6	9
4	7	3
```

13. find the sum of left diagonals of a matrix

```
#include<stdio.h>

int main()
{
	int i, j, rows, columns, a[10][10], Sum = 0;

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

	printf("\n Please 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++)
	{
		Sum = Sum + a[rows][rows];
	}

	printf("\n The Sum of Diagonal Elements of a Matrix =  %d", Sum );
```

```
            return 0;
}
OUTPUT-
Please Enter Number of rows and columns  :  3 3
 Please Enter the Matrix Elements
3 5 6
5 6 7
5 7 8
 The Sum of Diagonal Elements of a Matrix =  17
```

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

```
#include <stdio.h>

int main ( )
{
            int a[10][10];
            int i = 0, j = 0, row = 0, col = 0;

            printf ("Enter the order of the matrix :\n");
            scanf ("%d %d", &row, &col);

            int flag = 0;

            printf ("Enter the elements of the matrix\n");
            for (i = 0; i < row; i++)
            {
                        for (j = 0; j < col; j++)
                        {
                                    scanf ("%d", &a[i][j]);
                        }
            }

            for (i = 0; i < row; i++)
            {
                        for (j = 0; j < col; j++)
                        {
                                    if (i == j && a[i][j] != 1)
                                    {
                                                flag = -1;
                                                break;
                                    }
                                    else if (i != j && a[i][j] != 0)
                                    {
                                                flag = -1;
                                                break;
                                    }
                        }
            }
```

```
        if (flag == 0)
        {
                printf ("It is a identity matrix\n");
        }
        else
        {
                printf ("It is not an identity matrix\n");
        }

        return 0;
}
```

OUTPUT-

```
Enter the order of the matrix :
3 3
Enter the elements of the matrix
2 3 4
5 4 2
1 5 9
It is NOT an identity matrix
```