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

```
#include <stdio.h>
int main()
{
    int arr[10];
    int size, i;
    printf("Enter size of the array: ");
    scanf("%d", &size);

    printf("Enter elements in array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("\nArray in reverse order: ");
    for(i = size-1; i>=0; i--)
    {
        printf("%d\t", arr[i]);
    }

    return 0;
}
```

OUTPUT-

```
Enter size of the array: 4
Enter elements in array: 1 4 6 7

Array in reverse order: 7       6       4       1
```

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

```
#include <stdio.h>
 int main()
{
    int a[10],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 : 8
1 2 3 4 5 6 7 8
sum of array is : 14
```

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

```
 #include<stdio.h>
    int main() {
        int arr1[20], arr2[20];
        int i, j, n1;

printf("\nEnter size of array :");
        scanf("%d", &n1);
   printf("Enter the 1st Array elements one by one \n");
  for (i = 1; i<=n1; i++)
        scanf("%d", &arr1[i]);
        for (i = 1; i<=n1; i++)
        arr2[i] = arr1[i];

   printf("The Coppied Array elements in the 2nd Array : \n");
        for (i = 1; i<=n1; i++)
        printf(" %d", arr2[i]);
   return 0;
   }

  OUTPUT-
Enter size of array :4
Enter the 1st Array elements one by one
1 7 8 5
The Coppied Array elements in the 2nd Array :
 1 7 8 5
```

Q4.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 Enter Number of elements in an array  :   ");
      scanf("%d", &Size);

      printf("\n 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-

```
Enter Number of elements in an array  :   6

 Enter 6 elements of an Array  :  12 5 7 5 15 7

 Total Number of Duplicate Elements in this Array  =  2
```

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

```
#include <stdio.h>
int main(){
 int a[1000],i,n,min,max;

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

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

    min=max=a[0];
    for(i=1; i<n; i++)
    {
         if(min>a[i])
             min=a[i];
              if(max<a[i])
               max=a[i];
    }
     printf("minimum of array is : %d",min);
          printf("\nmaximum of array is : %d",max);
return 0;
}
```

OUTPUT-

```
Enter size of the array : 5
Enter elements in array : 8 9 4 6 10
```

```
minimum of array is : 4
maximum of array is : 10
```

Q6.separate odd and even integers in separate arrays.

```
#include<stdio.h>

int main()
{
 int i,j,k;
int num[10] = {1,2,3,4,5,6,7,8,9,10};
    int odd[10];
    int even[10];
    j = 0;
k = 0;
for(i = 0; i<10 ; i++){
        if(num[i]%2 == 0){
            even[j] = num[i];
            j++;
        }else{
            odd[k] = num[i];
            k++;
        }
    }
    printf("even numbers : ");
    for(i = 0;i<j; i++){
        printf("%d ",even[i]);
    }

    printf("\nodd numbers : ");
    for(i = 0;i<k; i++){
        printf("%d ",odd[i]);
    }
printf("\n");
return 0;
}
```

```
OUTPUT-
even numbers : 2 4 6 8 10
odd numbers : 1 3 5 7 9
```

Q7. insert New value in the array.

```
#include <stdio.h>
int main()
{
  int array[100], position, c, n, value;
   printf("Enter number of elements in array:");
   scanf("%d", &n);
```

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

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

   printf("Enter the location where you wish to insert an
element:");
   scanf("%d", &position);

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

   for (c = n - 1; c >= position - 1; c--)
   array[c+1] = array[c];
 array[position-1] = value;
   printf("Resultant array is:");
for (c = 0; c <= n; c++)
      printf("%d\t", array[c]);

   return 0;
}
```

```
OUTPUT-
Enter number of elements in array:5
Enter 5 elements:1 2 34 5
9
Enter the location where you wish to insert an element:4
Enter the value to insert:8
Resultant array is:1    2       34      8       5       9
```

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

```
#include <stdio.h>
int main()
{
   int array[100], z, c, n;
    printf("Enter number of elements in array:");
   scanf("%d", &n);
   printf("Enter %d elements\n", n);
   for (c = 0; c < n; c++)
      scanf("%d", &array[c]);
   printf("Enter the location where you wish to delete
element:");
   scanf("%d", &z);
  if (z >= n+1)
      printf("Deletion not possible.\n");
   else
```

```
   {
      for (c = z - 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 2 5 4 6 7 Enter the location where you wish to
delete element:2 Resultant array: 2
4
 6
 7
```

9. find the second largest element in an array.

```
#include <stdio.h>
int main()
{
     int array[10] = {50, 15, 80, 100, 70, 30, 75, 150, 170, 40};
   int a, largest, second_largest;
   if(array[0] > array[1]) {
      largest = array[0];
      second_largest  = array[1];
   } else {
      largest = array[1];
      second_largest = array[0];
   }
   for(a = 2; a < 10; a++) {
      if( largest < array[a] ) {
        second_largest = largest;
         largest = array[a];
      }
else if( second_largest < array[a] ) {
         second_largest=  array[a];
      }
   }
   printf("second_largest: %d \n",second_largest);
   return 0;
}
```

```
OUTPUT-
second_large=150
```

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,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[] = {1, 12, 15, 26, 38};
    int ar2[] = {2, 13, 17, 30, 45};
    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");
     return 0;
}
```

OUTPUT:

```
Median is 16
```

11. multiplication of two square Matrices

```
#include <stdio.h>
int main(){
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
printf("enter the first matrix element=\n");
for(i=0;i<r;i++){
for(j=0;j<c;j++){
scanf("%d",&a[i][j]);
}}
printf("enter the second matrix element=\n");
for(i=0;i<r;i++){
for(j=0;j<c;j++){
scanf("%d",&b[i][j]);
}}
printf("multiply of the matrix=\n");
for(i=0;i<r;i++){
for(j=0;j<c;j++){
mul[i][j]=0;
for(k=0;k<c;k++){
mul[i][j]+=a[i][k]*b[k][j];
}}}
for(i=0;i<r;i++){
for(j=0;j<c;j++){
printf("%d\t",mul[i][j]);
}
printf("\n");
}
return 0;
}
```

```
OUTPUT-
enter the number of row=2
enter the number of column=2
 enter the first matrix element= 9 8 7 6
enter the second matrix element= 10 11 2 8 multiply of the
matrix=
106   163
82    125
```

12. find transpose of a given matrix.

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

```
int main(){
  int a[10][10], transpose[10][10], r, c, i, j;
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);
    printf("\nEnter matrix elements:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    printf("\nEntered matrix: \n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("%d  ", a[i][j]);
            if (j == c - 1)
                printf("\n");
        }
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            transpose[j][i] = a[i][j];
        }
    printf("\nTranspose of the matrix:\n");
    for (i = 0; i < c; ++i)
        for (j = 0; j < r; ++j) {
            printf("%d  ", transpose[i][j]);
  if (j == r - 1)
                printf("\n");
        }
    return 0;
}

OUTPUT-
Enter rows and columns: 2
2

Enter matrix elements:
Enter element a11: 5 10
Enter element a12: Enter element a21: 15 8
Enter element a22:
Entered matrix:
5  10
15  8

Transpose of the matrix:
5  15
10  8
```

13. find the sum of left diagonals of a matrix

```
#include <stdio.h>
int 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);
    return 0;
}
```

```
OUTPUT-
Input the size of the square matrix : 2
Input elements in the first matrix :
element - [0],[0] : 10 20
element - [0],[1] : element - [1],[0] : 30 40
element - [1],[1] : The matrix is :
  10  20
  30  40
Addition of the  left Diagonal elements is :50
```

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

```
#include <stdio.h>
int main(){
   int arr1[10][10];
   int r1,c1;
```

```
   int i, j, yn =1;
   printf("Input number of Rows for the matrix :");
   scanf("%d", &r1);
   printf("Input number of Columns for the matrix :");
   scanf("%d",&c1);
       printf("Input elements in the first matrix :\n");
        for(i=0;i<r1;i++)
         {
             for(j=0;j<c1;j++)
  {
                  printf("element - [%d],[%d] : ",i,j);
                  scanf("%d",&arr1[i][j]);
             }
         }
       printf("The matrix is :\n");
       for(i=0;i<r1;i++)
       {
         for(j=0;j<c1 ;j++)
           printf("% 4d",arr1[i][j]);
          printf("\n");
       }
    for(i=0; i<r1; i++)
    {
     for(j=0; j<c1; j++)
     {
 if(arr1[i][j] != 1 && arr1[j][i] !=0)
       {
         yn = 0;
         break;
       }
      }
   }
   if(yn == 1 )
printf(" The matrix is an identity matrix.\n\n");
   else
      printf(" The matrix is not an identity matrix.\n\n");
      return 0;
}
```

```
OUTPUT-
Input number of Rows for the matrix :2
Input number of Columns for the matrix :2
Input elements in the first matrix :
element - [0],[0] : 2
element - [0],[1] : 4
element - [1],[0] : 5
element - [1],[1] : 6
The matrix is :
   2   4
   5   6
 The matrix is not an identity matrix.
```

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

```
#include <stdio.h>
int search(int mat[4][4], int n, int x)
{
int i = 0, j = n-1;
while ( i < n && j >= 0 )
{
    if ( mat[i][j] == x )
    {
        printf("The given value for searching is  %d, %d", i, j);
        return 1;
    }
    if ( mat[i][j] > x )
        j--;
    else
        i++;
}
printf("n Element not found");
return 0;
}
int main()
{
int mat[4][4] = { {10, 20, 30, 40},
              {15, 25, 35, 45},
              {27, 29, 37, 48},
              {32, 33, 39, 50},
                };
search(mat, 4, 29);
return 0;
}
```

OUTPUT-

```
The given value for searching is 2, 1
```