

## 1. READ N NUMBER OF VALUES IN AN ARRAY AND DISPLAY IT IN REVERSE ORDER.

```
#include <stdio.h>
int main(){
    int i,n,a[100];
    printf("Input the number of elements: ");
    scanf("%d",&n);
    printf("Input %d number of elements in the array: ",n);
    for(i=0;i<n;i++)
    {
        printf("element - %d: ",i);
        scanf("%d",&a[i]);
    }

    printf("The values store into the array are : \n");
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }

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

### OUTPUT:

Input the number of elements: 5

Input 5 number of elements in the array: element - 0: 1

element - 1: 2

element - 2: 3

element - 3: 4

element - 4: 5

The values store into the array are :

1    2    3    4    5

The values stored in the array reversed are :

5    4    3    2    1

## **2. FIND THE SUM OF ALL ELEMENTS OF THE ARRAY.**

```
#include <stdio.h>
int main(){
    int a[100];
    int i, n, sum=0;
    printf("Input the number of elements to be stored in the
array :");
    scanf("%d",&n);
    printf("Input %d elements in the array :\n",n);
    for(i=0;i<n;i++)
    {
        printf("element - %d : ",i);
        scanf("%d",&a[i]);
    }

    for(i=0; i<n; i++)
    {
        sum += a[i];
    }

    printf("Sum of all elements stored in the array is : %d\n",
sum);
    return 0;
```

```
}
```

### OUTPUT:

Input the number of elements to be stored in the array :5

Input 5 elements in the array :

element - 0 : 1

element - 1 : 2

element - 2 : 3

element - 3 : 4

element - 4 : 5

Sum of all elements stored in the array is : 15

### 3. COPY THE ELEMENTS OF ONE ARRAY INTO ANOTHER ARRAY.

```
#include <stdio.h>
int main(){
    int arr1[100], arr2[100];
    int i, n;
    printf("Input the number of elements to be stored in the
array :");
    scanf("%d",&n);
    printf("Input %d elements in the array :\n",n);
    for(i=0;i<n;i++)
    {
        printf("element - %d : ",i);
        scanf("%d",&arr1[i]);
    }
    // Copy
    for(i=0; i<n; i++)
    {
        arr2[i] = arr1[i];
    }
}
```

```

printf("\nThe elements stored in the first array are :\n");
for(i=0; i<n; i++)
{
    printf("%d\t", arr1[i]);
}

printf("\nThe elements copied into the second array are
:\n");
for(i=0; i<n; i++)
{
    printf("%d\t", arr2[i]);
}
return 0;
}

```

### OUTPUT:

Input the number of elements to be stored in the array :4

Input 4 elements in the array :

element - 0 : 1

element - 1 : 2

element - 2 : 3

element - 3 : 4

The elements stored in the first array are :

1    2    3    4

The elements copied into the second array are :

1    2    3    4

#### 4. COUNT A TOTAL NUMBER OF DUPLICATE ELEMENTS IN AN ARRAY.

```
#include <stdio.h>
int main(){
    int arr[100];
    int i, j, size, count = 0;
    printf("Enter size of the array : ");
    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 size of the array : 4

Enter elements in array : 1

2

1

3

Total number of duplicate elements found in array = 1

### 5. FIND THE MAXIMUM AND MINIMUM ELEMENT IN AN ARRAY.

```
#include <stdio.h>  
int main(){  
    int arr1[100];  
    int i, max, min, n;  
    printf("Input the number of elements to be stored in the  
array :");  
    scanf("%d",&n);  
    printf("Input %d elements in the array :\n",n);  
    for(i=0;i<n;i++)  
    {  
        printf("element - %d : ",i);  
        scanf("%d",&arr1[i]);  
    }  
}
```

```
max = arr1[0];
min = arr1[0];

for(i=1; i<n; i++)
{
    if(arr1[i]>max)
    {
        max = arr1[i];
    }

    if(arr1[i]<min)
    {
        min = arr1[i];
    }
}
printf("Maximum element is : %d\n", max);
printf("Minimum element is : %d\n", min);
return 0;
}
```

### **OUTPUT:**

Input the number of elements to be stored in the array :5

Input 5 elements in the array :

element - 0 : 16

element - 1 : 9

element - 2 : 15

element - 3 : 12

element - 4 : 1

Maximum element is : 16

Minimum element is : 1

## **6. SEPARATE ODD AND EVEN INTEGERS IN SEPARATE ARRAYS.**

```
#include <stdio.h>
int main(){
    int arr1[20], arr2[20], arr3[20];
    int i,j=0,k=0,n;
    printf("Input the number of elements to be stored in the
array: ");
    scanf("%d",&n);
    printf("Input %d elements in the array: ",n);
    for(i=0;i<n;i++){

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

    for(i=0;i<n;i++)
    {
        if (arr1[i]%2 == 0)
        {
            arr2[j] = arr1[i];
            j++;
        }
        else
        {
            arr3[k] = arr1[i];
            k++;
        }
    }

    printf("The Even elements are: ");
```



```

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

    printf("\nThe Odd elements are: ");
    for(i=0;i<k;i++)
    {
        printf("%d ", arr3[i]);
    }
    return 0;
}

```

### OUTPUT:

Input the number of elements to be stored in the array: 5

Input 5 elements in the array: 1

2

5

6

3

The Even elements are: 2 6

The Odd elements are: 1 5 3

### 7. INSERT NEW VALUE IN THE ARRAY.

```

#include <stdio.h>
int main(){
    int arr1[100],i,n,p,x;
    printf("Input the size of array : ");
    scanf("%d", &n);
    /* Stored values into the array*/
}

```

```

printf("Input %d elements in the array: \n",n);
for(i=0;i<n;i++){
    scanf("%d",&arr1[i]);
}

printf("Input the value to be inserted: ");
scanf("%d",&x);
printf("Input the Position, where the value to be inserted :");
scanf("%d",&p);

printf("The current list of the array :\n");
for(i=0;i<n;i++)
    printf("%d ",arr1[i]);
//Move all data at right side of the array
for(i=n;i>=p;i--)
    arr1[i]= arr1[i-1];
//insert
arr1[p-1]=x;

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

```

### OUTPUT:

Input the size of array : 3

Input 3 elements in the array:

1

2

6

Input the value to be inserted: 3

Input the Position, where the value to be inserted :2

The current list of the array :

1 2 6

After Insert the element the new list is :

1 3 2 6

## **8. DELETE AN ELEMENT AT DESIRED POSITION FROM AN ARRAY.**

```
#include <stdio.h>
int main(){
    int arr1[50],i,pos,n;
    printf("Input the size of array : ");
    scanf("%d", &n);
    /* Stored values into the array*/
    printf("Input %d elements in the array: \n",n);
    for(i=0;i<n;i++){
        scanf("%d",&arr1[i]);
    }

    printf("Input 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]);
    }
    return 0;

}

```

### OUTPUT:

Input the size of array : 4

Input 4 elements in the array:

1

5

2

3

Input the position where to delete: 2

The new list is : 1 2 3

### 9. FIND THE SECOND LARGEST ELEMENT IN AN ARRAY.

```

#include <stdio.h>
int main(){
    int arr1[50],n,i,j=0,large,seclarge;
    printf("Input the size of array : ");
    scanf("%d", &n);
    printf("Input %d elements in the array :\n",n);
    for(i=0;i<n;i++){

```

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

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

/\* ignore the largest element and find the 2nd largest element in the array \*/

```
    seclarge=0;
    for(i=0;i<n;i++)
    {
        if(i==j)
        {
            i++; //lrgr
            i--;
        }
        else
        {
            if(seclarge<arr1[i])
            {
                seclarge=arr1[i];
            }
        }
    }
}
```

```
printf("The Second largest element in the array is: %d \n",
seclarge);
}
```

### OUTPUT:

Input the size of array : 5

Input 5 elements in the array :

1

5

8

9

15

The Second largest element in the array is: 9

### 10. FIND THE MEDIAN OF TWO SORTED ARRAYS OF SAME SIZE.

```
#include <stdio.h>
int main(){
    int arr1[100], arr2[100], merge[100], i, j, k, m, n, temp;
    printf("enter the size of the array1: \n");
    scanf("%d", &m);
    printf("enter the size of the array2: \n");
    scanf("%d", &n);
    printf("enter the elements for array1: \n");
    for(i=0; i<m; i++){
        printf("enter arr[%d]: ", i);
        scanf("%d", &arr1[i]);
    }
    printf("enter the elements for array2: ");
    for(i=0; i<n; i++){
```

```
    printf("enter arr[%d]: ", i);
    scanf("%d",& arr2[i]);
}
i = 0;
j = 0;
k = 0;

// merge two arrays
while(i<m && j<n){
    if(arr1[i] <= arr2[j]){
        merge[k] = arr1[i];
        i++;
        k++;
    }
    else{
        merge[k] = arr2[j];
        k++;
        j++;
    }
}

while(i<m){
    merge[k] = arr1[i];
    i++;
    k++;
}

while(j<n){
    merge[k] = arr2[j];
    j++;
    k++;
}

// print the merged array
```

```

printf("the merged array is: \n");
for(i=0; i< (m+n); i++){
    printf("%d ", merge[i]);
}

// sort the merged array
for(i=0; i<(m+n); i++){
    for(j=i+1; j<(m+n); j++){
        if(merge[i] > merge[j]){
            temp = merge[i];
            merge[i] = merge[j];
            merge[j] = temp;
        }
    }
}

// the array after sorting
printf("\nthe merged array after sorting is: \n");
for(i=0; i< (m+n); i++){
    printf("%d ", merge[i]);
}

// finding median
float median=0;
int length = m+n;
if(length%2 == 0)
    median = (merge[(length-1)/2] + merge[length/2])/2.0;
else
    median = merge[length/2];

printf("\nthe median is %f ", median);
return 0;
}

```



## OUTPUT:

enter the size of the array1:

3

enter the size of the array2:

5

enter the elements for array1:

enter arr[0]: 1

enter arr[1]: 2

enter arr[2]: 3

enter the elements for array2: enter arr[0]: 5

enter arr[1]: 4

enter arr[2]: 1

enter arr[3]: 2

enter arr[4]: 3

the merged array is:

1 2 3 5 4 1 2 3

the merged array after sorting is:

1 1 2 2 3 3 4 5

the median is 2.500000

## 11. MULTIPLICATION OF TWO SQUARE MATRICES.

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

```
int main(){
```

```
    int
```

```
    arr1[50][50],arr2[50][50],arr[50][50],i,j,k,r1,c1,r2,c2,sum=0;
```

```

printf("Input the rows and columns of first matrix: ");
scanf("%d %d",&r1,&c1);
printf("\nInput the rows and columns of second matrix :
");
scanf("%d %d",&r2,&c2);
if(c1!=r2){
printf("Mutiplication of Matrix is not possible.");
printf("\nColumn of first matrix and row of second matrix
must be same.");
}
else
{
printf("Input elements in the first matrix :\n");
for(i=0;i<r1;i++){
for(j=0;j<c1;j++){

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

}
}
printf("Input elements in the second matrix :\n");
for(i=0;i<r2;i++){
for(j=0;j<c2;j++){

scanf("%d",&arr2[i][j]);

}
}
printf("\nThe First matrix is: ");
for(i=0;i<r1;i++){
printf("\n");
for(j=0;j<c1;j++)
printf("%d\t",arr1[i][j]);
}

```

```

        printf("\nThe Second matrix is: ");
        for(i=0;i<r2;i++){
            printf("\n");
            for(j=0;j<c2;j++)
                printf("%d\t",arr2[i][j]);
        }
//multiplication
    for(i=0;i<r1;i++)
        for(j=0;j<c2;j++)
            arr[i][j]=0;
        for(i=0;i<r1;i++) //row of first matrix
        {
            for(j=0;j<c2;j++) //column of second matrix
            {
                sum=0;
                for(k=0;k<c1;k++)
                    sum=sum+arr1[i][k]*arr2[k][j];
                arr[i][j]=sum;
            }
        }
    printf("\nThe multiplication of two matrices is: ");
    for(i=0;i<r1;i++)
    {
        printf("\n");
        for(j=0;j<c2;j++)
        {
            printf("%d\t",arr[i][j]);
        }
    }
}
return 0;
}

```

## OUTPUT:

Input the rows and columns of first matrix: 2

2

Input the rows and columns of second matrix : 2

2

Input elements in the first matrix :

1

3

2

4

Input elements in the second matrix :

1

2

1

1

The First matrix is:

1    3

2    4

The Second matrix is:

1    2

1    1

The multiplication of two matrices is:

4     5

6     8

## 12.     **FIND TRANSPOSE OF A GIVEN MATRIX.**

```
#include <stdio.h>
int main(){
    int arr1[50][50],arr2[50][50],i,j,r,c;
    printf("Input the rows and columns of the matrix: ");
    scanf("%d %d",&r,&c);
    printf("Input elements in the matrix :\n");
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            scanf("%d",&arr1[i][j]);
        }
    }

    printf("\nThe matrix is: ");
    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++){
            arr2[j][i]=arr1[i][j];
        }
    }

    printf("\nThe transpose of a matrix is: ");
    for(i=0;i<c;i++){
        printf("\n");
```

```

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

```

### OUTPUT:

Input the rows and columns of the matrix: 2

2

Input elements in the matrix :

1

2

3

4

The matrix is:

1    2

3    4

The transpose of a matrix is:

1    3

2    4

### 13.      FIND THE SUM OF LEFT DIAGONALS OF A MATRIX.

```

#include <stdio.h>
int main(){
    int i,j,arr[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++){
            scanf("%d",&arr[i][j]);
        }
    }
    printf("The matrix is: \n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n ;j++)
            printf("% 4d",arr[i][j]);
        printf("\n");
    }
// sum of left diagonals
    for(i=0;i<n;i++){
        m=m-1;
        for(j=0;j<n ;j++){
            if (j==m)
            {
                sum= sum+arr[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 matrix:

1

1

2

4

The matrix is:

1 1

2 4

Addition of the left Diagonal elements is: 3

#### **14. CHECK WHETHER A GIVEN MATRIX IS AN IDENTITY MATRIX.**

```
#include <stdio.h>
int main(){
    int arr[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 matrix: \n");
    for(i=0;i<r1;i++){
        for(j=0;j<c1;j++){
            scanf("%d",&arr[i][j]);
        }
    }
    printf("The matrix is :\n");
```



```

        for(i=0;i<r1;i++)
        {
            for(j=0;j<c1 ;j++)
                printf("% 4d",arr[i][j]);
            printf("\n");
        }

for(i=0; i<r1; i++)
{
    for(j=0; j<c1; j++)
    {
        if(arr[i][j] != 1 && arr[j][i] !=0)
        {
            yn = 0;
            break;
        }
    }
}

if(yn == 1 )
    printf("The matrix is an identity matrix.\n");
else
    printf("The matrix is not an identity matrix.\n");
return 0;
}

```

### **OUTPUT:**

Input number of Rows for the matrix: 2

Input number of Columns for the matrix: 2

Input elements in the matrix:

1

0

0

1

The matrix is :

```
1  0
0  1
```

The matrix is an 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=37;

    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: 37

The element Found at the position in the matrix is: 3,3