

## Arrays

★ :- An array in C is a collection of items stored at contiguous memory locations and elements can be accessed randomly using indices of an array. They are used to store similar type of elements as in the data type must be the same for all elements. They can be used to store collection of primitive data types such as int, float, double, char, etc. of any particular type. To add it, an array in C can store derived data types such as the structures, pointers, etc.

★ Why do we need Arrays?

⇒ We can use normal variables ( $v_1, v_2, v_3, \dots$ ) when we have a small number of objects, but if we want to store a large number of instances, it becomes difficult to manage them with normal variables. The idea of an array is to represent many instances in one variable.

~~int S1 = 10, S2 = 20, S3 = 40;~~

→ 100 values  $\times$  ~~100~~ this will not use if we want to declare 100 variables.

~~int S[3] = { 20, 40, 50 };~~

length of array  
Subscript

Creation cum  
initialization of  
an array

S → name
S[0]
S[1]
S[2]

Array

Array = body  
loop = soul

Now, if we want to input | ~~process~~ and output of  
Salary of  $(n)$  employ u.

```
int main()
{
    int n, i, sum=0;
    printf (" No. of Employee : ");
    scanf ("%d", &n); // 3
```

```
int s[n]; // Creation of array of n elements
for (i=0; i < n; i++)
```

```
{   printf (" Enter Salary at s[%d] pos : %d ", i);
    scanf ("%d", &s[i]); // Input
}
```

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

```
{   printf ("%d lac in", s[i]); // output
    sum = sum + s[i]; // processing
}
```

```
printf (" In total Salary = %d lac ", sum);
```

Enter No. of Emp :
Enter Sal of s[0], -
s[1] :-
s[2] :-
Salary slip :
20 Lac
40 Lac
50 Lac
Total Sum = 110 Lac

## || Greatest Element of Array

```
int main()
```

```
{ int N, i, max;
```

```
    printf ("Enter the value of N: ");
```

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

```
    int s[N];
```

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

```
    { printf ("Enter No. at s[%d] pos : ", i);
```

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

```
    max = s[0];
```

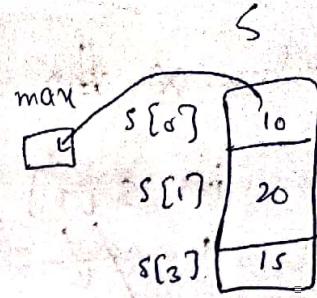
```
    for (i=0 ; i<N ; i++)
```

```
    { if (max < s[i])
```

```
        max = s[i];
```

```
    printf ("Max = %d", max);
```

```
}
```



## Some Technicals :-

int A [ ] = { 1, 2, 3, 4 }; ✓

1	2	3	4
---	---	---	---

int A [2] = { 1, 2, 3 }; X

int A [3] ;

garbage value	garbage. Value	garbage value
------------------	-------------------	------------------

int A [3] = { 2, 3 };

2	3	0
---	---	---

it will Take '0', if there is no value.

static int A [3];

0	0	0
---	---	---

# Home Work

(short question)

- ① || find error if any?

```
int main()
{
    int a[3] = { 6, 9, 5 };
    int b[] = { 7, 5, 3, 2 };
    int c[6] = { 4, 5, 6 };
}
```

⇒ no - Error

- ② || Tell output

```
int main()
{
    int d[3] = { 14, 16 };
    int i = 0;
    printf ("%d", d[i]);
}
```

⇒ Here,  $d[0] = 14$

- ③ int main ()

```
{
    int e [3] = { 14, 16 };
    int i;
    for (i=0; i<3; i++)
    {
        printf ("%d", e[i]);
    }
}
```

14	16	0
----	----	---

$$\Rightarrow e[1] = 14 \\ e[2] = 16 \\ e[3] = 0$$

④ int main()

```
{ int d[2] = {14, 16};  
    int i;  
    for (i=0; i<2; i++)  
    {  
        printf ("%d", d[i]);  
    }  
    printf ("%d", d[i]);  
}
```

$$\Rightarrow d[0] = 14$$

$$d[1] = 16 \quad \text{Common place } d[3] = 0$$

⑤ int main()

```
{ int a[2] = {50}, i=1;  
    a[i] = i * 10;  
    printf ("%d", a[i]);  
}
```

$$\Rightarrow 10 \quad [\because a[i] = a[1] = 1 * 10 = 10]$$

⑥ int main()

```
{ int a[2] = {5, 3, 6}, b[2] = {5, 3, 13}, c[2], i;  
    for (i=2; i >= 0; i--)  
    {  
        c[2-i] = a[i] * b[i];  
        printf ("%d", c[2-i]);  
    }
```

$$\Rightarrow c[0] = 6$$

$$c[1] = 9$$

$$c[2] = 25$$

$$\Rightarrow \underline{\underline{6 \ 9 \ 25}} \ c$$

Q- Write a program for smallest element in an array

A- int main()

{

int N, i;

cout << "Enter the value of N: ";

cin >> N;

int s[N];

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

{

cout << "Enter the value at s[" << i << " position: ";

cin >> s[i];

Min = s[0];

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

{

if (min > s[i])

min = s[i];

}

cout << "Smallest Element = " << min;

}

S	
s[0]	10
s[1]	15
s[2]	45
s[3]	20
s[4]	25
s[5]	4
s[6]	12
s[7]	16

Q - Input the price of n products in array and find total & avg price.

A - int main()

```
{  
    int N, i;  
    float sum = 0;  
    printf ("Enter the Value of n: ");  
    scanf ("%d", &N);  
    int s[N]; //  
    for (i=0; i<n; i++)  
    {  
        printf (" Enter value at s[%d] : ", i);  
        scanf ("%d", &s[i]);  
    }  
    for (i=0; i<n; i++)  
    {  
        printf (" Price of %d product : %d ", i+1, s[i]);  
        sum = sum + s[i];  
    }  
    printf (" Total amount : %f ", sum); // 2500.00  
    printf (" Average : %f ", sum/n); // 833.33  
}
```

S	s[0]	400
s[1]	700	
s[2]	800	
s[3]	100	
s[4]	500	

Q- Write a program to input marks of N students  
 And Count no. of students got  $\geq 80$  marks and  
 also Count who got marks b/w 60 & 70.

A- int main()

{

int N, i, Count1 = 0, Count2 = 0;

printf ("Enter the Value of N: ");

scanf ("%d", &N) || 6;

int s[N];

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

{

printf ("Enter the Value at s [%d] : ", i);

scanf ("%d", &s[i]); // Input

s[0]	85
s[1]	75
s[2]	68
s[3]	45
s[4]	90
s[5]	84

s[0]:

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

{

printf ("Marks of %d student : %d \n", i+1, s[i]);

if (s[i]  $\geq 80$ )

Count1 ++;

else if (s[i]  $\geq 60$  & s[i]  $\leq 70$ )

Count2 ++;

printf ("No. of Students who got  $\geq 80$  marks : %d \n", Count1);

printf ("No. of Students who got  $\geq 60$  &  $\leq 70$  marks : %d \n", Count2);

}

Q- Write a program to input N elements in 2 Arrays and do their sum in third array.

⇒ int main()

{

int N, i

printf ("Enter the value of N : ");

scanf ("%d", &N);

int A[N], B[N], C[N];

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

{

printf ("Enter value at A[%d] : ", i);

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

}

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

{

printf ("Enter value at B[%d] : ", i);

}

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

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

{

printf ("C[%d] = A[%d] + B[%d]",

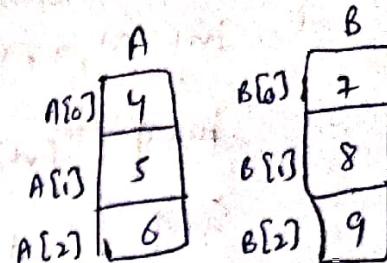
printf ("C[%d] = %d\n", i, A[i] + B[i]);

}

|| ||

|| 13 → Output

|| 15



Q. Write a program for no. of days b/w 2 months.

```
A. int main()
{
    int N, i, Sm, Em, Sum = 0;
    printf ("Enter value of N : \n");
    scanf ("%d", &N);
    int s[N]; int Days [N];
    for (i=0; i<N; i++)
    {
        printf ("Enter the value at %d : \n", i);
        scanf ("%d", &Days[i]);
    }
    printf ("Enter starting month and end month \n");
    scanf ("%d %d", &Sm, &Em); // (4 - 7) months
    for (i=Sm; i <= Em; i++)
    {
        printf ("%d", Days[i]);
        Sum = Sum + Days[i];
    }
    printf ("\n = %d", Sum);
    return 0;
}
```

5
30
31
30
31

## || Linear Searching :-

```
int main()
{
    int N, Count = 0, i, Wanted;
```

```
    printf ("Enter the value of N: \n");
    scanf ("%d", &N);
```

```
    int s[N];
```

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

```
{
```

```
        printf ("Enter Salary at s[%d]: \n", i);
        scanf ("%d", &s[i]);
```

```
}
```

```
    printf ("Enter Value Want to Search: \n");
    scanf ("%d", &Wanted);
```

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

```
{
```

```
    if (Wanted == s[i])
```

```
{
```

```
        printf ("found at %d location \n", i);
        Count++;
    }
```

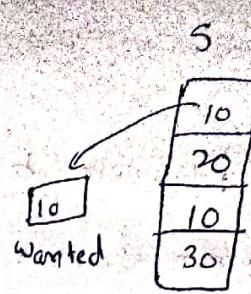
```
}
```

```
if (Count == 0)
```

```
    printf ("Not found");
```

```
else
```

```
    printf ("found %d times", Count);
```



## Home Work

Q- write a program to input elements in 2 arrays (sets) and find common element (intersection)

```
A - int main()
{
    int n, i, j, Count = 0;
    printf (" Enter the value of n : ");
    scanf ("%d", &n);
    int a[n], b[n];
    for (j=0 ; j<n ; j++)
    {
        printf (" Enter the value at a[%d] : ", j);
        scanf ("%d", &a[j]);
    }
    for (i=0 ; i<n ; i++)
    {
        printf (" Enter the value at b[%d] : ", i);
        scanf ("%d", &b[i]);
    }
    for (i=0 ; i<n ; i++)
    {
        for (j=0 ; j<n ; j++)
        {
            if (a[i]==b[j])
            {
                printf (" Common element : %d ", a[i]);
                Count++;
            }
        }
    }
    printf (" total common elements : %d ", Count);
}
```

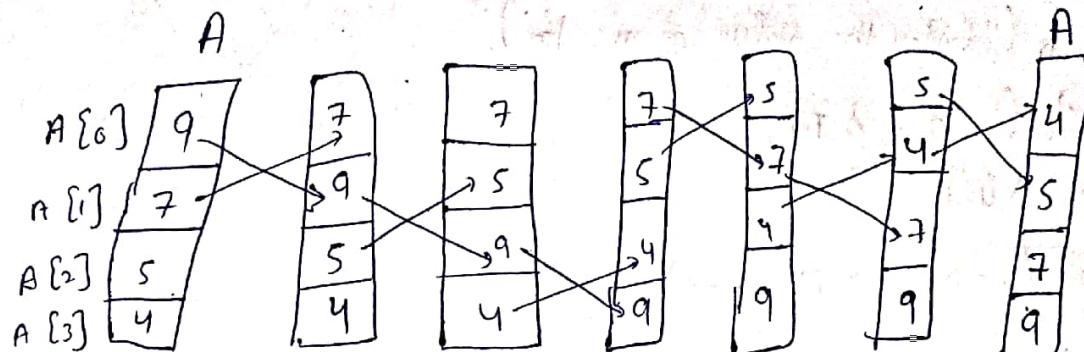
**★ Sorting of Arrays** :- Arrange Elements in order

(Bubble Sort)

$N=4$

Ascending

Descending



$A[0] > A[1]$

$A[0] > A[1] \Rightarrow \text{swap}$

$A[1] > A[2] \Rightarrow \text{swap}$

$A[2] > A[3] \Rightarrow \text{swap}$

3 Comparisons

$(N-1)$

$A[0] > A[1] \Rightarrow \text{swap}$

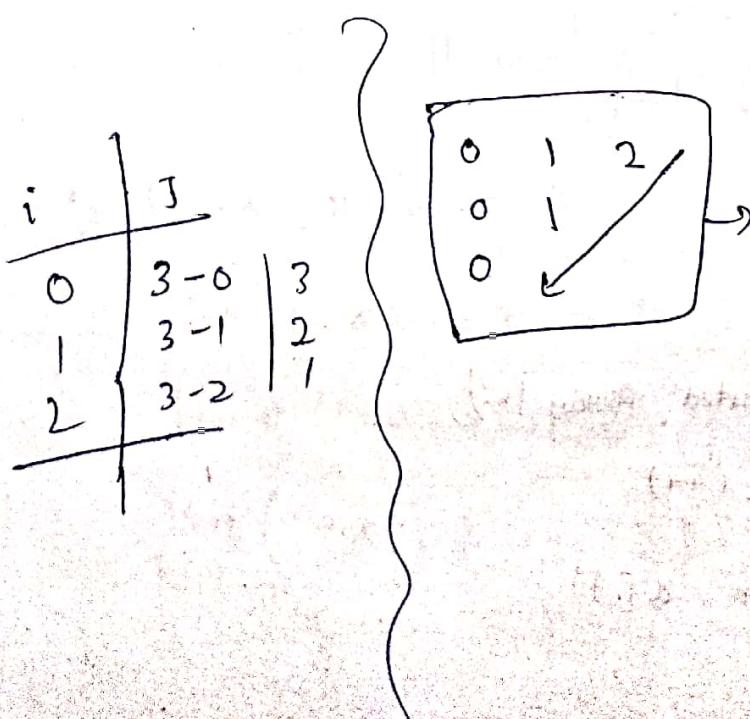
2 Comparisons

$(N-2)$

$A[0] > A[1] \Rightarrow \text{swap}$

1 Comparison

$(N-3)$



it shows running of loop  
which we will discuss on  
next page, OR it also show  
for the program how works

We see in diagram above,  
that swapping of positions.

## Program :

```
int main()
{
    int N, i, j, temp;
    printf("Enter the value of n : ");
    scanf("%d", &N);
    int A[N];
    for (i=0; i< N; i++)
    {
        printf("Enter the value at A[%d] : ", i);
        scanf("%d", &A[i]);
    }
    for (i=0; i< n-1; i++)
    {
        for (j=0; j< n-i-1; j++)
        {
            if (a[j] > a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
    printf("In sorted Array %n");
    for (i=0; i< n; i++)
    {
        printf("%d %n", a[i]);
    }
}
```

## ★ 2-Dimensional Array :-

1) Input / process / output of a 2-D array of m x n elements

```
int A [2] [3] = { { 10, 20, 30 },
                  { 40, 6, 15 } }
```

↕      ↕  
 Rows    Columns

Hmm isko aise Padhenge, Lets suppose ki ye value hai

Temperatures ki, Jaise 1 dim mein 3 bar Temperature

Record kuna hai aur ye 2 dim tak Record kuna, To

We will use 2-D array, Jaha aisa kam kuna ho.

Jaise 5 Student ke 6 Subjects mein marks, like that.

```
int main()
{
    int R, C, H, T, Sum = 0;
    cout { "Enter the value of Hours & times : \n" };
    cin { ".d .d", &H, &T };
    int A [H] [T]; // creation of 2-D
                    // Array
    for (R = 0 ; R < H ; R++)
    {
        for (C = 0 ; C < T ; C++)
        {
            cout { "Enter temperature at A [".d ][ ".d ] : " R );
            cin { ".d", &A [R] [C] };
        }
    }
}
```

Enter Hours & times : 2,3

Enter temp. at A [0][0] :

\_\_\_\_\_ A[0][0]

\_\_\_\_\_ !  
A[2][3]

Output :-

10	20	30
40	6	15

Average temp. :-

```
        printf ("In output In");
```

```
    for (R = 0; R < H; R++)
```

```
{   for (c = 0; c < T; c++)
```

```
    {   printf ("%d ", A[R][c]);
```

$$Sum = Sum + A[R][c];$$

```
    printf ("In");
```

```
}
```

```
    printf ("In Average Temperature : %d In", Sum/(R*C));
```

# Home Work

Q- Input a matrix of  $m \times n$  and print Diagonal Elements.

```
#include <iostream.h>
int main()
{
    int R, C, M, N;
    cout << "Enter the value of m & n : ";
    cin << M << N;
    int A[M][N];
    for (R=0; R<M; R++)
    {
        for (C=0; C<N; C++)
        {
            cout << "Enter the value at A[" << R << "][" << C << "]; ";
            cin << A[R][C];
        }
    }
    for (R=0; R<M; R++)
    {
        for (C=0; C<N; C++)
        {
            if (R==C)
                cout << A[R][C];
            else if (R!=C)
                cout << "0";
        }
    }
    cout << "In";
}
```

A- Input a matrix  $m \times N$  and Print user diagonal elements

A- int main()

{  
int R, C, M, N;

Printf ("Enter the value of m & n: ");

scanf ("%d %d", &M, &N);  
for (R=0; R<M; R++)

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

{  
printf ("Enter the value at A[%d][%d]: ", R, C);

scanf ("%d", &A[R][C]);

}

for (R=0; R<M; R++)

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

{  
if (R == C)

printf ("%d ", A[R][C]);

else if (R+C == m-1)

printf ("%d ", A[R][C]);

else

printf ("0");

}

printf ("\n");

}

(0,0)	(0,1)	(0,2)
1	2	3
(1,0)	(1,1)	(1,2)
4	5	6
(2,0)	(2,1)	(2,2)
7	8	9

2+0=2

Q-1. Write a program for sum of two Matrices of order  $m \times n$

```
A- int main()
{
    int m, n, R, C;
    printf ("Enter m & n: ");
    scanf ("%d %d", &m, &n);
    int A[m][n], B[m][n];
    for (R=0; R<m; R++)
    {
        for (C=0; C<n; C++)
        {
            printf ("Enter value at A[%d][%d]: ", R, C);
            scanf ("%d", &A[R][C]);
        }
    }
}
```

```
for (R=0; R<m; R++)
{
    for (C=0; C<n; C++)
    {
        printf ("Enter value at B[%d][%d]: ", R, C);
        scanf ("%d", &B[R][C]);
    }
}
```

```
int P[m][n];
for (R=0; R<m; R++)
{
    for (C=0; C<n; C++)
    {
        P[R][C] = A[R][C] + B[R][C];
        printf ("%d ", P[R][C]);
    }
    printf ("\n");
}
```

Result :-      Output :-

A[m][n]	<table border="1"><tr><td>4</td><td>5</td><td>7</td></tr><tr><td>2</td><td>3</td><td>9</td></tr><tr><td>0</td><td>1</td><td>4</td></tr></table>	4	5	7	2	3	9	0	1	4
4	5	7								
2	3	9								
0	1	4								

B[m][n]	<table border="1"><tr><td>7</td><td>2</td><td>9</td></tr><tr><td>8</td><td>6</td><td>2</td></tr><tr><td>3</td><td>2</td><td>1</td></tr></table>	7	2	9	8	6	2	3	2	1
7	2	9								
8	6	2								
3	2	1								

P[m][n]

11	7	16
10	9	11
3	3	5

~~Q- Write a program for multiplication of 2 matrices of m x n.~~

~~int main ()~~

~~{~~

~~int m, n, R, C;~~

~~printf (" Enter value of m & n: ");~~

~~scanf ("%d %d", &m, &n);~~

~~int A[m][n], B[m][n];~~

~~for (R=0; R<m; R++)~~

~~for (C=0; C<n; C++)~~

~~{ printf (" Enter value at A[%d][%d]: ",~~

~~scanf ("%d", &A[R][C]);~~

~~for (R=0; R<m; R++)~~

~~for (C=0; C<n; C++)~~

~~printf (" Enter value of B[%d][%d]: ",~~

~~scanf ("%d", &B[R][C]);~~

~~int P[m][n];~~

~~for (R=0; R<m; R++)~~

~~for (C=0; C<n; C++)~~

~~P[R][C] = A[R][C] \* B[R][C];~~

~~printf ("%d - ", P[R][C]);~~

~~printf ("\n");~~

Output :-

$A[m][n] =$

1	3	2	4
5	3	6	7
4	8	7	9
0	4	2	5

$B[m][n] =$

3	6	1	2
4	8	3	7
9	6	4	5
8	6	3	4

$P[m][n]$

3	18	2	8
20	24	48	49
36	48	28	45

## II Matrix Multiplication of m x n orders

```
int main()
{
    int R, C, k, RA, CA, RB, CB, Sum;
    printf (" Enter order of matrix A[ ][ ] : m");
    scanf ("%d %d", &RA, &CA);
    printf (" Enter order of matrix B[ ][ ] : n");
    scanf ("%d %d", &RB, &CB);
    if (CA != RB)
        printf (" Invalid order ! ");
    else
        int A [RA] [CA], B [RB] [CB], AB [RA] [CB];
        for (R=0 ; R<RA ; R++)
        {
            for (C=0; C<CA; C++)
            {
                printf (" Enter value in A[%d][%d] : ", R, C);
                scanf ("%d", &A [R] [C]);
            }
        }
        for (R=0 ; R<RB ; R++)
        {
            for (C=0; C<CB; C++)
            {
                printf (" Enter value in B[%d][%d] : ", R, C);
                scanf ("%d", &B [R] [C]);
            }
        }
}
```

AB	0	1
0	10	14
1	14	20

AB	0	1	(A) c C 1
0	1	3	
1	2	4	

AB	0	1
0	1	2
1	3	4

$$\text{Sum} = \text{Sum} + A[R][n] \times B[n][C]$$

for ( $R=0$ ;  $R < RA$ ;  $R++$ )

{ for ( $c=0$ ,  $C < CB$ ;  $c++$ )

{ Sum = 0;

for ( $n=0$ ;  $k < CA$ ;  $n++$ )

{

$$\text{Sum} = \text{Sum} + A[R][n] \times B[n][C];$$

}

$$AB[R][C] = \text{Sum};$$

}

printf ("In multiplied matrix: %m");

for ( $R=0$ ;  $R < RA$ ;  $R++$ )

{

for ( $c=0$ ;  $C < CB$ ;  $c++$ )

{

printf ("%d ", AB[R][C]);

}

printf ("\n %m");

}

} // else

} // int main()

# Homework

# Repetition of

## previous programs

11 Write a program for union of elements in two arrays.

⇒ int main()

```
{ int N, i, j;
```

```
printf ("Enter the value of n: ");
```

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

```
int A[n], B[m];
```

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

```
{ printf (" Enter the value at A[%d]: ", i);
```

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

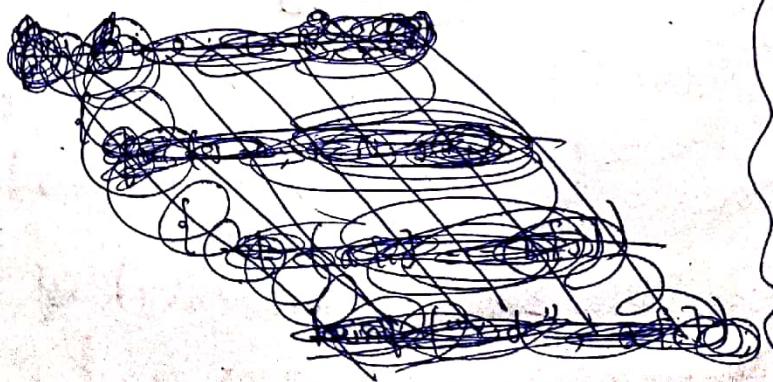
```
}
```

```
for (i=0; i<m; i++)
```

```
{ printf (" Enter the value at B[%d]: ", i);
```

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

```
}
```



A

1
2
3
6
7
9

B

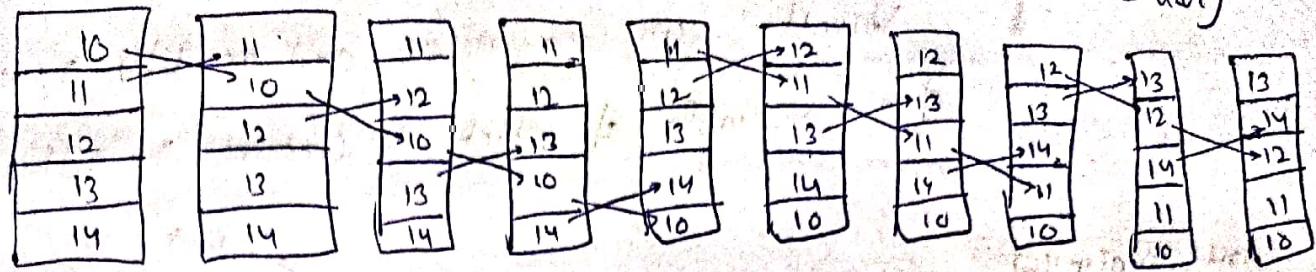
10
12
2
3
6
11
33
32

```
for (i=0; i<n; i++)
    point ("l+", d - ")", a[i]);
for (i=0; i<n; i++)
{
    for (j=d; j<m; j++)
    {
        if (b[i] == a[j])
            bunge = 1;
        else if (b[i] != a[j])
            bunge = 0;
        if (bunge == 1)
            continue;
        else if (bunge == 0)
            cout << "l+", d - ")", b[i]);
```

```
for (i=0; i<n; i++)
{
    cout << "l+", d - ")", a[i]);
}

for (i=0; i<n; i++)
{
    for (j=0; j<m; j++)
    {
        if (b[i] == a[j])
        {
            bunge = 1;
            break;
        }
        else if (b[i] != a[j])
        {
            bunge = 0;
        }
        if (bunge == 1)
            continue;
        else if (bunge == 0)
            cout << "l+", d - ")", b[i]);
    }
}
```

Sorting of two arrays → For Practice [ output in Descending Order ]



```

int main()
{
    int n, i, j, temp;
    printf("Enter value of n: ");
    scanf("%d", &n);
    int A[n];
    for (i=0; i<n; i++)
    {
        printf("Enter A[%d]: ", i);
        scanf("%d", &A[i]);
    }
    for (i=0; i<n-1; i++)
    {
        for (j=0; j<n-1-i; j++)
        {
            if (a[j] < a[j+1])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
}

```