

Array And Strings

Course Title :- Structured Programming Language Sessional

Course Code :- CSE-122 [SECTION-B]

Level Term: 1-II-A(G₁) & 1-II-B(G₃,G₄)

Outlines

- ✓ PART - 1 : 1D array taking input and print output
 - ❑ integer array
 - ❑ float array
 - ❑ double array
- ✓ PART - 2 : 2D array taking input and print output
 - ❑ integer array
 - ❑ float array
 - ❑ double array
- ✓ PART - 3 : Properties of array
- ✓ PART - 4 : List of C-program for 1D Array
- ✓ PART - 5 : List of C-program for 2D Array
- ✓ PART - 6 : Online judge problems
- ✓ PART - 7 : Exercise Problems

❑ PART - 1 : 1D array taking input and print output

1. Array declaration
2. Array initialization
3. Array access
4. Array update
5. Array traversal

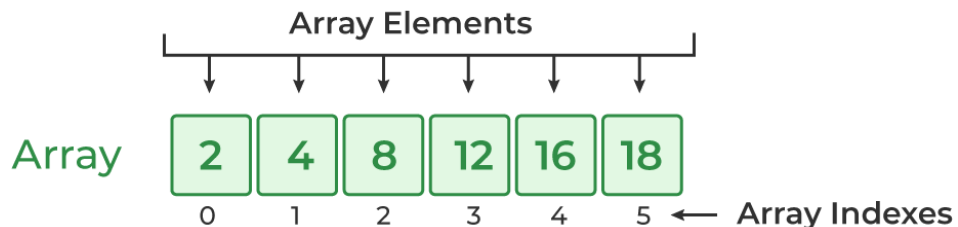
❑ 1D array taking input and print output - Integer

➤ Array in C What is Array in C?

- ✓ An array in C is a fixed-size collection of similar data items stored in contiguous memory locations.
- ✓ It can be used to store the collection of primitive data types such as int, char, float, etc.
- ✓ It is a simple and fast way of storing multiple values under a single name.

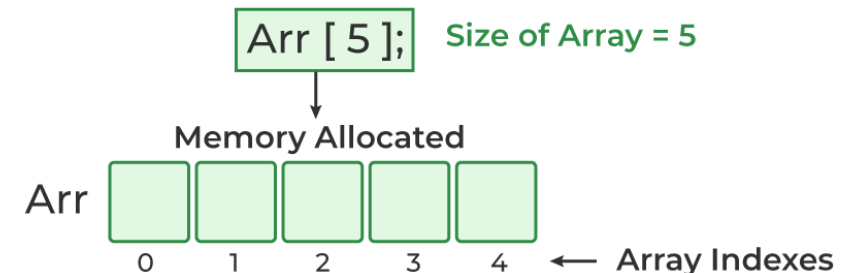
1. C Array Declaration

Array in C



```
char alphabets[5];  
double prices[5];  
float height[5];  
int age[5];
```

Array Declaration



```
int age[10];
```

0 - based indexing

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

1 - based indexing

Index	1	2	3	4	5	6	7	8	9	10
Values										

Array Initialization

```
Arr [ 5 ] = { 2, 4, 8, 12, 16 };
```



Memory Allocated and Initialized



2. Array initialization [How to put values in array?]

1. 0 - base indexing

```
int main(){
    int age[5];
    age[0] = 20;
    age[1] = 21;
    age[2] = 20;
    age[3] = 34;
    age[4] = 12;
}
```

2. 1 - base indexing

```
int main(){
    int age[5];
    age[1] = 21;
    age[2] = 20;
    age[3] = 34;
    age[4] = 12;
    age[5] = 20;
}
```

3. Array Initialization using user input

```
int main(){
    int n;
    scanf("%d", &n);
    int array[n];
    for(int i = 0; i <n; i++){
        scanf("%d ", &array[i] );
    }
}
```

4. Array Initialization with Declaration without Size

```
int age[ ] = {11, 21, 32, 43, 56};
```

5. Array Initialization with Declaration

```
int age[5] = {20, 21, 22, 23, 12};
```

3. Array access [How to print array?]

```
int main(){
    int age[5];
    age[0] = 20;
    age[1] = 21;
    age[2] = 20;
    age[3] = 34;
    age[4] = 12;
    printf("%d ", age[4]);
    printf("%d ", age[1]);
    printf("%d ", age[3]);
    printf("%d %d", age[0], age[2]);
}
```

4. Update Array Element

```
int n;
scanf("%d", &n);

int array[n];
for(int i = 0; i <n; i++){
    scanf("%d ", &array[i] );
}
array[0] = - 555;
for(int i=0; i<n; i++){
    printf("%d ", array[i]);
}
```

Assignment:

1D array taking input and print output - Float

1D array taking input and print output - Double

```
int n;
scanf("%d", &n);

int array[n];
for(int i = 0; i <n; i++){
    scanf("%d ", &array[i] );
}

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

5. C Array Traversal

```
int n;
scanf("%d", &n);

int array[n];
for(int i = 0; i <n; i++){
    scanf("%d ", &array[i] );
}
for(int i=0; i<n; i++){
    printf("%d ", array[i]);
}
```

❑ PART – 2 : 2D array taking input and print output

▪ integer array

1. Array declaration
2. Array initialization
3. Array access
4. Array update
5. Array traversal

❑ 2D array taking input and print output – Integer

[1] Array declaration

//method-1

```
row = 3;
```

```
col = 4;
```

```
int matrix[row][col] ;
```

//method-2

```
int array[2][3];
```

//method-3

```
int n,m;
```

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

```
int matrix[ n ][ m ];
```

0-based indexing 2D array

i / j	j=0	j=1	j=2	j=3
i=0	10 [0, 0]	20 [0, 1]	30 [0, 2]	40 [0, 3]
i=1	10 [1, 0]	20 [1, 1]	30 [1, 2]	40 [1, 3]
i=2	10 [2, 0]	20 [2, 1]	30 [2, 2]	40 [2, 3]
i=3	10 [3, 0]	20 [3, 1]	30 [3, 2]	40 [3, 3]

1-based indexing 2D array

i / j	j=1	j=2	j=3	j=4
i=1	10 [1, 1]	20 [1, 2]	30 [1, 3]	40 [1, 4]
i=2	10 [2, 1]	20 [2, 2]	30 [2, 3]	40 [2, 4]
i=3	10 [3, 1]	20 [3, 2]	30 [3, 3]	40 [3, 4]
i=4	10 [4, 1]	20 [4, 2]	30 [4, 3]	40 [4, 4]

//method-4: 0-based indexing

```
int row, col;
scanf("%d %d", &row, &col);
int matrix[ row ] [ col ] ;
for(i = 0; i<row; i++){
    for(j=0; j<col; j++){
        scanf("%d ", &matrix[i][j]);
    }
}
```

[2] Array Initialization

//method-1

```
int row=3;
int col=4;
int matrix[row][col] = {{5, 5,5,5},    {1,2,3,4},    {4,3,2,1} }
```

//method-2

```
int col=4;
int matrix[ ][col] = {{5, 5,5,5},    {1,2,3,4},    {4,3,2,1} }
```

//method-3

```
int mx[3][4] = {0, 1 ,2 ,3 ,4 , 5 , 6 , 7 , 8 , 9 , 10 , 11}
```

//method-4: 1-based indexing

```
int row, col;
scanf("%d %d", &row, &col);
int matrix[ row ] [ col ] ;
for(i = 1; i≤row; i++)
{
    for(j=1; j≤col; j++)
    {
        scanf("%d ", &matrix[ i ][ j] );
    }
}
```

3. Array access [How to print array?]

```
int row, col;
scanf("%d %d", &row, &col);
int matrix[ row ] [ col ] ;
for(i = 0; i<row; i++){
    for(j=0; j<col; j++){
        scanf("%d ", &matrix[i][j]);
    }
}
printf("%d ", matrix[1][2]);
for(i = 0; i<row; i++){
    for(j=0; j<col; j++){
        printf("%d ", matrix[i][j]);
    }
}
```

5. C Array Traversal

```
int row, col;
scanf("%d %d", &row, &col);
int matrix[row][col];
for(i = 0; i<row; i++){
    for(j=0; j<col; j++){
        scanf("%d ", &matrix[i][j]);
    }
}
for(i = 0; i<row; i++){
    for(j=0; j<col; j++){
        printf("%d ", matrix[i][j]);
    }
}
```

4. Update Array Element

```
int row, col;
scanf("%d %d", &row, &col);
int matrix[row][col] ;
for(i = 0; i<row; i++){
    for(j=0; j<col; j++){
        scanf("%d ", &matrix[i][j]);
    }
}
matrix[0][0] = -30;
for(i = 0; i<row; i++){
    for(j=0; j<col; j++){
        printf("%d ", matrix[i][j]);
    }
}
```

There Can be Multi-dimensional Array!

`int array[2][2][3]`

Assignment:

2D array taking input and print output – Float
2D array taking input and print output – Double

PART – 3: C Array Properties

1. Fixed Size of an Array

```
int main(){
    int array[5] = { 1, 2, 3, 4, 5 };
    printf("Size of Array Before: %d\n", sizeof(array) / sizeof(int));
    array[6];
    printf("Size of Array After: %d", sizeof(array) / sizeof(int));
}
```

Output

Size of Array Before: 5

Size of Array After: 5

2. Homogeneous Collection

```
int main(){
    int arr[3] = { 1, 2 };
    arr[2] = "Geeks";
    printf("Array[1]: %d\n", arr[0]);
    printf("Array[2]: %d\n", arr[1]);
    printf("Array[3]: %s", arr[2]);
}
```

Output

Array[1]: 1

Array[2]: 2

3. Indexing in an Array: By default index of the first element will be 0 and last element will be (size – 1) where size is the size of the array.

4. Random Access: we can get to a random element at any index of the array in constant time complexity just by using its index number.

5. No Index Out of Bounds:

// This C program compiles fine as index out of bound is not checked in C.

```
int main(){
    int arr[2];
    printf("%d ", arr[3]);
    printf("%d ", arr[-2]);
}
```

Output:

211343841

4195777

In C, it is not a compiler error to initialize an array with more elements than the specified size. For example, the below program compiles fine and shows just a Warning.

```
#include <stdio.h>
int main(){
    // Array declaration by initializing it
    // with more elements than specified size.
    int arr[2] = { 10, 20, 30, 40, 50 };
}
```

PART – 4 : List of C-program for 1D Array

Array	H. W	Recommended tutorials
1.Taking input & output		Array Input/Output Bangla Tutorial
2.reverse an array	Assignment	Print Elements of Array In Reverse Order
3.Copy array1 to array2	Assignment	copy all elements of an array to another array
4.concatenate two array into one	Assignment	Concatenate two array
5.Linear search		Searching a number (Linear search)
6.Sort an array [ascending]- bubble sort		1. Bubble Sort Algorithm 2. Bubble Sort 3. Bubble Sort Algorithm and Program
7.Sort an array [descending]	Assignment	Sort Array Elements in Descending Order
8.Replace a number with P in position X of array		Replace a number
9.Count even, odd, positive, negative in an array	Assignment	Count even, odd, positive, negative
10.Count specific number in an array	Assignment	Count specific number
11.Sum of digits of an integer value		1. sum of digits (Theory) 2. sum of digits (practical)
12.Sum and average of array		Sum and Average of an Array
13.Check two array is same or not	Assignment	compare two arrays equality
14.Check array is Palindrome or not		check if an Array is Palindrome or not
15.Check an integer value is Palindrome or not/reverse of a number	Assignment	1. How To Reverse A Number (Theory) 2. Reverse a number (practical) 1) Palindrome number (Theory) 2) Palindrome number (practical)
16.Find maximum number in an array		Maximum and Minimum of Array
17.Find minimum number in an array	Assignment	Maximum and Minimum of Array
18.Fibonacci series using array		Fibonacci series using array

1. Taking input & output

```
int main(){
    int n,i;
    printf("Enter Array Size: ");
    scanf("%d", &n);

    int array[n];
    printf("Enter Array Elements: ");
    for(i=0; i<n ; i++){
        scanf("%d", &array[i]);
    }
    printf("Output: ");
    for(i=0; i<n ; i++){
        printf("%d ", array[i]);
    }
}
```

Output:

Enter Array Size: 5

Enter Array Elements: 1 2 3 4 5

Output: 1 2 3 4 5

C-Program 2,3,4 → Assignment!

5. Searching a number (Linear search)

```
int main(){
    int N,i;
    scanf("%d", &N);
    int array[N];
    for(i=1; i≤ N; i++){
        scanf("%d", &array[i]);
    }
    printf("Enter element to Search: ");
    int value, check=0, position = 0;
    scanf("%d", &value);
    for(i=1; i≤ N; i++){
        if(array[i] == value){
            check=1;
            position = i;
        }
    }
    if(check == 1){
        printf("%d found at position %d\n", value, position);
    }
    else{
        printf("Value Not found\n");
    }
}
```

Output:

5

10 2 8 4 0

Enter element to Search: 8

8 found at position 3

6. Sort Array in Ascending order – Bubble sort

```
int main(){
    int n, i, j, temp;
    printf("Enter Array Size: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d elements: ", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    printf("Sorted array:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}
```

Output:

```
Enter Array Size: 4
Enter 4 elements:
7 6 4 3
Sorted array:
3 4 6 7
```

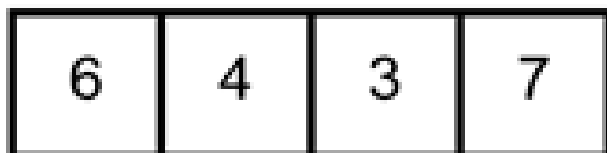
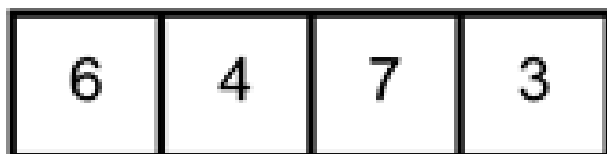
How this worked?

```
Enter Array Size: 4
Enter 4 elements:
7 6 4 3
6 4 3 7
4 3 6 7
3 4 6 7
Sorted array:
3 4 6 7
```

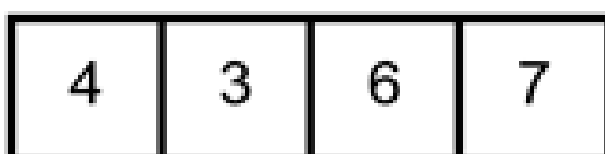
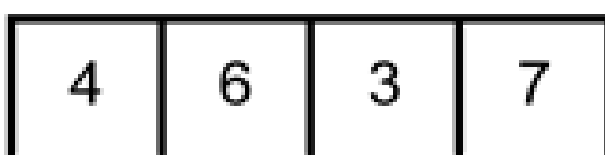
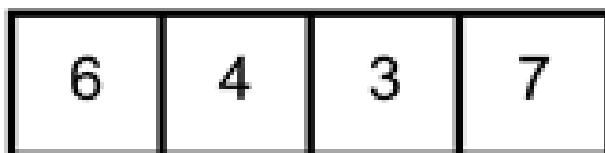
C-Program 7 → Assignment!

□ Visual Representation of Bubble Sort

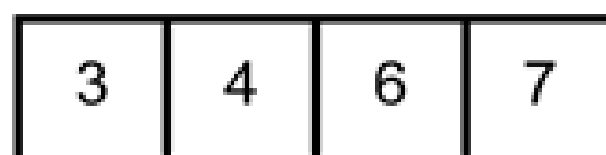
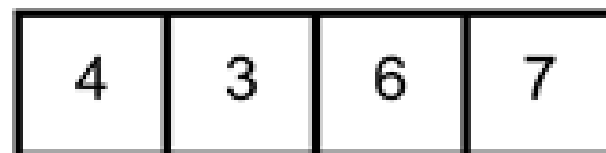
First pass



Second pass



Third pass



8. Replace a number with P in position X of array

```
int main() {
    int n, i, position, newValue;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int arr[n];
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter the position to replace: ");
    scanf("%d", &position);
    if (position < 1 || position > n) {
        printf("Invalid position!\n"); return 0;
    }
    printf("Enter the new value: ");
    scanf("%d", &newValue);

    arr[position - 1] = newValue;
    printf("Updated array:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

Output:

Enter the size of the array: 5
30 5 6 9 8

Enter the position to replace: 3
Enter the new value: -100

Updated array:
30 5 -100 9 8

C-Program 9, 10 → Assignment!

11. Sum of digits of number N

```
int main(){
    int N, sum = 0;
    scanf("%d", &N);

    while(N > 0){
        int digit = N % 10;
        sum = sum + digit;
        N = N / 10;
    }
    printf("Sum : %d", sum);
}
```

Output:

```
123
Sum : 6
```

12. Sum of all elements of array

```
int main(){

    int N, sum = 0;
    scanf("%d", &N);
    int array[N];
    for(int i=0; i<N; i++){
        scanf("%d", &array[i]);
    }
    for(int i=0; i<N; i++){
        sum = sum + array[i];
    }
    printf("Sum : %d", sum);
}
```

Output:

```
5
4 9 -3 2 5
Sum : 17
```

C-Program 13 → Assignment!

14. Palindromic array

```
int main(){
    int N, sum = 0;
    scanf("%d", &N);
    int array[N];
    for(int i=0; i<N; i++){
        scanf("%d", &array[i]);
    }
    int i=0, j = N-1;
    while(i < j){
        if(array[i] != array[j]){
            printf("Array is not a palindrome!");
            return 0;
        }
        i++;
        j--;
    }
    printf("Array is palindrome");
}
```

Output:

```
5
1 2 3 2 1
Array is palindrome
```

15. Palindromic number

```
int main(){
    int N, sum = 0, M = 0, P;
    scanf("%d", &N);

    P = N; /// save main number
    while(N > 0){
        int digit = N % 10; ///find last digit
        M = (M*10) + digit;
        N = N/10;
    }
    printf("%d\n", M);
    if(M == P){
        printf("Palindrome Number!");
    }
    else{
        printf("Not a Palindrome Number!");
    }
}
```

Output:

```
1221
1221
Palindrome Number!
```


16+17. Maximum and Minimum of Array

```
int main(){
    int N;
    scanf("%d", &N);
    int array[N];
    for(int i=1; i≤ N; i++){
        scanf("%d", &array[i]);
    }
    int ans1 = -1;
    for(int i=1; i≤N; i++){
        if(array[i] ≥ ans1){
            ans1 = array[i];
        }
    }
    printf("max = %d\n", ans1);
    int ans2 = 9999999999;
    for(int i=1; i≤N; i++){
        if(array[i] ≤ ans1){
            ans1 = array[i];
        }
    }
    printf("min = %d\n", ans1);
}
```

Output:

5

1 2 3 4 5

Max = 5

Min = 1

18. Fibonacci series using array

```
int main(){
    int N;
    scanf("%d", &N);
    ///Fibonacci array: 0, 1, 1, 2, 3, 5, 8

    int array[N];
    array[0] = 0;
    array[1] = 1;

    for(int i = 2; i ≤ N; i++)
    {
        array[i] = array[i-1] + array[i-2];
    }

    for(int i = 0; i ≤ N; i++)
    {
        printf("%d ", array[i]);
    }
}
```

Output:

7

0 1 1 2 3 5 8 13

PART – 5: List of C-program for 2D Array

Array	H. W	Recommended tutorials
1. Taking input & output		<ul style="list-style-type: none">• Introduction to 2D Array• Getting input for 2D Array• Simple Matrix• code
2. Reverse an array	Assignment	<ul style="list-style-type: none">• Print Elements of 2D Array In Reverse Order
3. Copy array1 to array2	Assignment	<ul style="list-style-type: none">• copy all elements of an 2D array to another array•
4. Linear search		<ul style="list-style-type: none">• Slide•
5. Count specific number in an array	Assignment	<ul style="list-style-type: none">• Count specific number 2D array
6. Count even, odd, positive, negative in an array	Assignment	<ul style="list-style-type: none">• Count even, odd, positive, negative of 2D array
7. Sum of all elements in 2D array		<ul style="list-style-type: none">• slide
8. Sum of diagonal elements		<ul style="list-style-type: none">• Sum of diagonal elements of a matrix
9. Check two array is same or not	Assignment	<ul style="list-style-type: none">• compare two arrays equality
10. Find maximum number in an array		<ul style="list-style-type: none">• slide
11. Find minimum number in an array	Assignment	<ul style="list-style-type: none">• Minimum of 2D Array
12. matrix addition and subtraction		<ul style="list-style-type: none">• Matrix Addition & Subtraction

1. 2D array Input & Output

```
int main(){
    int n,m;
    scanf("%d %d", &n, &m);
    int array[n][m];

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            scanf("%d", &array[i][j]);
        }
    }
    printf("Output:\n");
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            printf("%d ", array[i][j]);
        }
        printf("\n");
    }
}
```

Output:

```
2 3
1 2 3
4 5 6
```

Output:

```
1 2 3
4 5 6
```

C-Program 2,3 → Assignment!

4. Linear search

```
int main(){
    int n,m;
    scanf("%d %d", &n, &m);
    int array[n][m];
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            scanf("%d", &array[i][j]);
        }
    }
    int p;
    printf("Enter value of search: ");
    scanf("%d", &p);
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            if(array[i][j] == p){
                printf("Value found at array[%d][%d]", i, j);
                return 0;
            }
        }
    }
    printf("Value not found!");
}
```

Output:

```
3 3
10 20 30
40 50 60
70 80 90
```

Enter value of search: 80

Value found at array[2][1]

C-Program 5,6 → Assignment!

7. Sum of all elements of a 2D array

```
int main(){
    int n,m;
    scanf("%d %d", &n, &m);
    int array[n][m];

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            scanf("%d", &array[i][j]);
        }
    }
    int sum = 0;
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            sum = sum + array[i][j];
        }
    }
    printf("sum: %d\n", sum);
}
```

Output:

```
2 3
0 0 1
1 1 1
sum: 4
```

8. Sum of diagonal elements in a 2D array

```
int main(){
    int n,m;
    scanf("%d %d", &n, &m);
    int array[n][m];

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            scanf("%d", &array[i][j]);
        }
    }
    int sum = 0;
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            if(i == j){
                sum = sum + array[i][j];
            }
        }
    }
    printf("sum: %d\n", sum);
}
```

Output:

```
3 3
10 20 30
40 50 60
70 80 90
sum: 150
```

10. Find maximum number in an array

```
int main(){
    int n,m;
    scanf("%d %d", &n, &m);
    printf("Enter values of matrix: \n");
    int array[n][m];
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            scanf("%d", &array[i][j]);
        }
    }
    int ans = -1;
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            if(ans ≤ array[i][j])
            {
                ans = array[i][j];
            }
        }
    }
    printf("maximum value: %d", ans);
}
```

Output:

```
3 3
Enter values of matrix:
10 -9 150
-5 34 75
4 5 2
maximum value: 150
```

C-Program 11 → Assignment!

12. Matrix Addition & Subtraction

```
int main(){
    int n,m;
    scanf("%d %d", &n, &m);
    printf("Enter values of matrix 1: \n");
    int array1[n][m];
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            scanf("%d", &array1[i][j]);
        }
    }
    printf("Enter values of matrix 2: \n");
    int array2[n][m];
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            scanf("%d", &array2[i][j]);
        }
    }
    printf("Addition of Two Matrix: \n");
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            printf("%d ", array1[i][j]+array2[i][j]);
        }
        printf("\n");
    }
    printf("Subtraction of Matrix: \n");
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            printf("%d ", array1[i][j]-array2[i][j]);
        }
        printf("\n");
    }
}
```

Output:

```
3 3
Enter values of matrix 1:
3 3 3
2 2 2
4 4 4
Enter values of matrix 2:
0 0 0
1 1 1
5 6 7
Addition of Two Matrix:
3 3 3
3 3 3
9 10 11
Subtraction of Matrix:
3 3 3
1 1 1
-1 -2 -3
```

PART – 6: Online judge problems

Codeforces:

1. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/A>
2. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/B>
3. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/C>
4. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/G>
5. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/H>
6. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/F>
7. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/K>
8. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/O>
9. <https://codeforces.com/group/MWSDmqGsZm/contest/219774/problem/S>

BeeCrowd

10. <https://judge.beecrowd.com/en/problems/view/1180>

PART – 7: Exercise Problems – 1

<pre>int main(){ int a[5]={5,4,3,2,1},n=5,m=0,i; for(i=0;i<n;i++){ printf("%d ",a[i]); } printf("%d\n",m); }</pre>	<pre>int main(){ int arr[5] = { 10, 20, 30, 40, 50 }; arr[2] = 100; printf("Elements in Array: "); for (int i = 0; i < 5; i++) { printf("%d ", arr[i]); } }</pre>	<pre>int main(){ int arr[5] = { 10, 20, 30, 40, 50 }; int arr1[] = { 1, 2, 3, 4, 5 }; float arr2[5]; for (int i = 0; i < 5; i++) { arr2[i] = (float)i * 2.1; } }</pre>
<pre>int main(){ int arr[5]; for (int i = 0; i < 5; i++) { arr[i] = i * i - 2 * i + 1; } printf("Elements of Array: "); for (int i = 0; i < 5; i++) { printf("%d ", arr[i]); } }</pre>	<pre>int main() { int i; int arr[5] = {1}; for (i = 0; i < 5; i++) printf("%d ", arr[i]); return 0; }</pre>	<pre>int main() { int i; int arr[5] = {0}; for (i = 0; i <= 5; i++) printf("%d ", arr[i]); return 0; }</pre>
<pre>int main(){ int a[][] = {{1,2},{3,4}}; int i, j; for (i = 0; i < 2; i++) for (j = 0; j < 2; j++) printf("%d ", a[i][j]); }</pre>	<pre>int main(){ int arr[5]; arr++; printf("%u", arr); }</pre>	<pre>int arr[] = {2,3,4,5}; printf("%d", 2[arr]);</pre>

Exercise Problems – 2

```
int main ()
{
    int i, j;
    int a [8] = {1, 2, 3, 4, 5, 6, 7, 8};
    for(i = 0; i < 3; i++) {
        a[i] = a[i] + 1;
        i++;
    }
    i--;
    for (j = 7; j > 4; j--) {
        int i = j/2;
        a[i] = a[i] - 1;
    }
    printf ("%d, %d", i, a[i]);
}
```

```
void main() {
    int a[] = {22, 19, 17, 36, 12, 15, 28, 35, 66, 43};
    int i, j, n = sizeof(a)/sizeof(int);
    for(i = 0; i < n; ++i)
        for(j = 0; j < i; ++j)
            if (a[i] > a[j]) {
                a[i] = a[i] + a[j];
                a[j] = a[i] - a[j];
                a[i] = a[i] - a[j];
            }

    for(i = 0; i < n; ++i)
        printf("%d ", a[i]);
}
```

```
int main () {
    int sum = 0, maxsum = 0, i, n = 6;
    int a [] = {2, -2, -1, 3, 4, 2};
    for (i = 0; i < n; i++) {
        if (i == 0 || a [i] < 0 || a [i] < a [i - 1])
        {
            if (sum > maxsum) maxsum = sum;
            sum = (a [i] > 0) ? a [i] : 0;
        }
        else sum += a [i];
    }
    if (sum > maxsum) maxsum = sum ;
    printf ("%d\\n", maxsum);
}
```

```
int main(){
    int arr[2][3] = { 10, 20, 30, 40, 50, 60 };
    printf("2D Array:\\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ",arr[i][j]);
        }
        printf("\\n");
    }
}
```