Programming for problem solving using C

123ES

Ms. Urvashi Gohil
Assistant Professor

Birla Vishvakarma Mahavidhyalaya (BVM) Engineering College
Department of Computer Engineering

Feb 25, 2025

Outline of the course :

Introduction

Array Syntax

One dimensional Array Problems

Multidimensional Array

Character Array

String

Errors

Introduction

Array user input

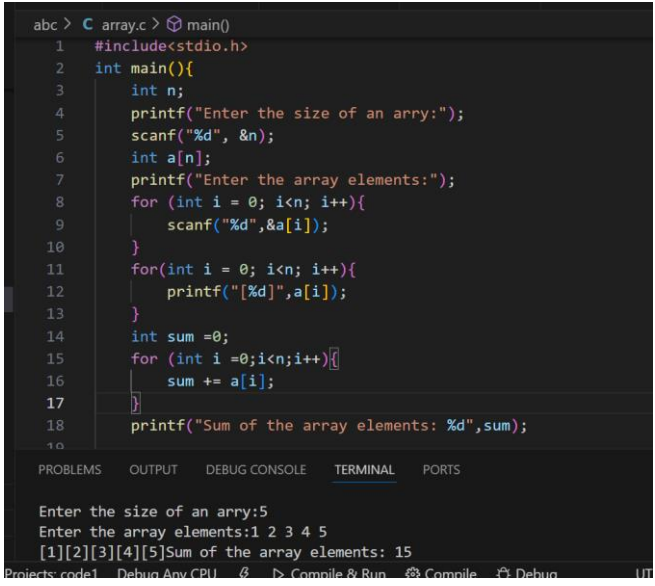
abc > C array.c > main()

```c
#include<stdio.h>
int main(){
    int n;
    printf("Enter the size of an arry:");
    scanf("%d", &n);
    int a[n];
    printf("Enter the array elements:");
    for (int i = 0; i<n; i++){
        scanf("%d",&a[i]);
    }
    for(int i = 0; i<n; i++){
        printf("[%d]",a[i]);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'array.exe'
Enter the size of an arry:5
Enter the array elements:2 3 4 5 5
[2][3][4][5][5]
PS F:\code1\abc\output>
```

Sum of array elements



min of array elements


```c
int main(){
    int n;
    printf("Enter the size of an arry:");
    scanf("%d", &n);
    int a[n];
    printf("Enter the array elements:");
    for (int i = 0; i<n; i++){
        scanf("%d",&a[i]);
    }
    for(int i = 0; i<n; i++){
        printf("[%d]",a[i]);
    }
    int min = a[0];
    for (int i =0;i<n;i++){
        if(min > a[i]){min = a[i]; }
    }
    printf("min of the array elements: %d",min);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Enter the size of an arry:3
Enter the array elements:1 2 4
[1][2][4]min of the array elements: 1
PS F:\code1\abc\output>
```

Check identical array or not


```c
#include <stdio.h>
int main() {
    int arr1[] = {1, 2, 3, 4, 5};
    int arr2[] = {1, 2, 0, 4, 5};
    int size = sizeof(arr1) / sizeof(arr1[0]);
    int i, isEqual = 1;

    for (i = 0; i < size; i++) {
        if (arr1[i] != arr2[i]) {
            isEqual = 0;  // Mark as different
            printf("Arrays differ at index %d: arr1[%d] = %d, arr2[%d] = %d\n", i, i, arr1[i], i, arr2[i]);
        }
    }
    if (isEqual)
        printf("Both arrays are identical.\n");
    else
        printf("Arrays are not identical.\n");
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Arrays differ at index 2: arr1[2] = 3, arr2[2] = 0
Arrays are not identical.
PS F:\code1\abc\output>
```

Assign arr2 in arr1

abc > C assingarr.c > main()

```c

int main() {
    int arr1[] = {1, 2, 3, 4, 5};
    int arr2[5];
    int size = sizeof(arr1) / sizeof(arr1[0]);

    for (int i = 0; i < size; i++) {
        arr2[i] = arr1[i];
    }
    printf("Copied array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr2[i]);
    }
    printf("\n");
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'assingarr.exe'
Copied array: 1 2 3 4 5
PS F:\code1\abc\output>
```

Reverse an array

abc > C reverse.c > main()

```c
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

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

    printf("Reversed array: ");
    for (int i = n - 1; i >= 0; i--) {
        printf("%d ", arr[i]);
    }
    printf("\n");

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> & .\'reverse.exe'
Original array: 1 2 3 4 5
Reversed array: 5 4 3 2 1
PS F:\code1\abc\output>
```

Reverse an array - 2

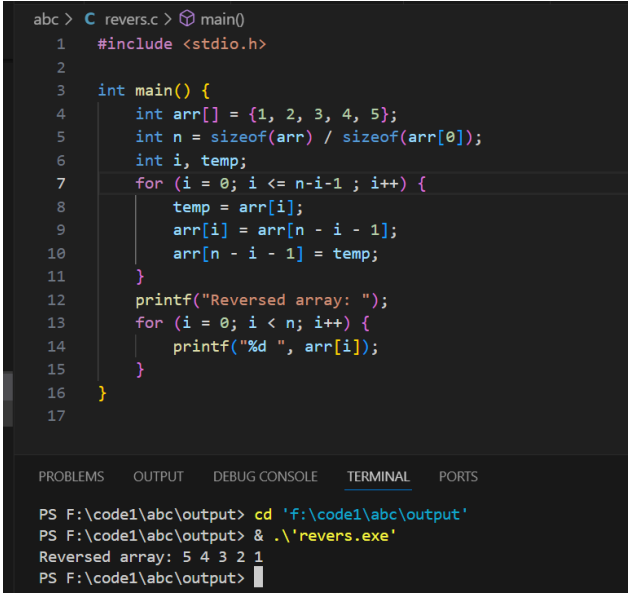
abc > C revarr.c > main()

```c
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int rev_arr[n];

    for (int i = 0; i < n; i++) {
        rev_arr[i] = arr[n - i - 1];
    }
    printf("Reversed array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", rev_arr[i]);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'revarr.exe'
Reversed array: 5 4 3 2 1
PS F:\code1\abc\output>
```

Reverse an array –with swapping



Linear search in array

abc > C linear search.c > main()

```c
#include <stdio.h>

int main() {
    int n, element;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int a[n];
    printf("Enter the array elements: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }
    printf("Enter the element to search: ");
    scanf("%d", &element);
    for (int i = 0; i < n; i++) {
        if (element == a[i]) {
            printf("Element found at index: %d\n", i);
        }
    }
    printf("Element not found.\n");

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Enter the number of elements: 5
Enter the array elements: 2 4 6 8 0
Enter the element to search: 6
Element found at index: 2
PS F:\code1\abc\output>
```

Sorting in array: Insertion sort

Insertion sort is a sorting algorithm that places an unsorted element at its suitable place in each iteration.

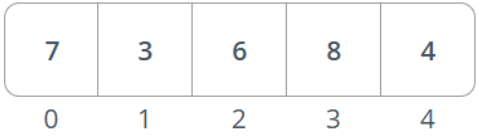
Step-1 First element is considered as a sorted, take a second element and stored it separately in a key variable

Step-2 compare first element with key, if it is grater than key then key will be fall first before the first element.

Step-3 Now first 2 elements are sorted (smaller placed first) , now 3rd element is compared with left (first 2). Placed it just behind the element smaller than it. If there is no element smaller than it, then place it at the beginning of the array.

Step-4 repeatedly follow these steps place every unsorted element at its correct position

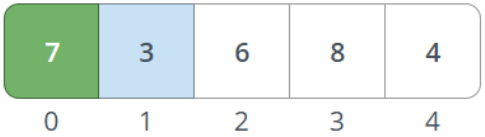
Sorting in array: Insertion sort



Steps:

Starting Insertion Sort.

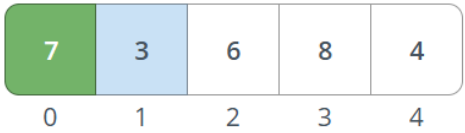
Sorting in array: Insertion sort



Steps:

Highlighted green records to the left are always sorted. We begin with the record in position 0 in the sorted portion, and we will be moving the record in position 1 (in blue) to the left until it is sorted.

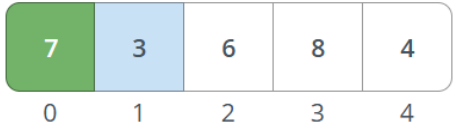
Sorting in array: Insertion sort



Steps:

Processing record in position 1

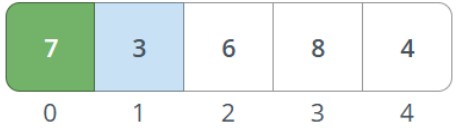
Sorting in array: Insertion sort



Steps:

Move the blue record to the left until it reaches the correct position.

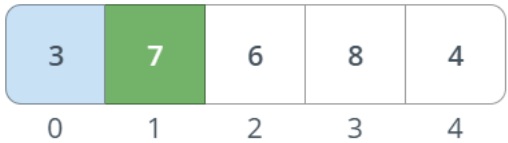
Sorting in array: Insertion sort



Steps:

Move the blue record to the left until it reaches the correct position.

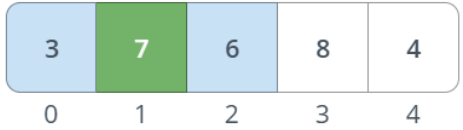
Sorting in array: Insertion sort



Steps:

Swap.

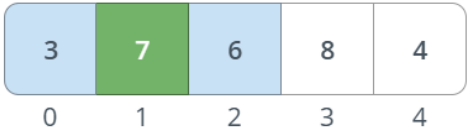
Sorting in array: Insertion sort



Steps:

Processing record in position 2

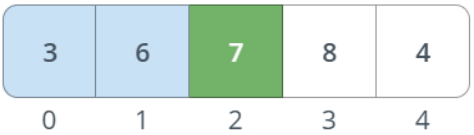
Sorting in array: Insertion sort



Steps:

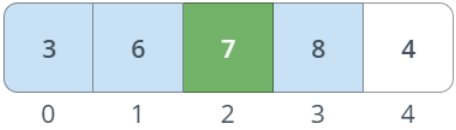
Move the blue record to the left until it reaches the correct position.

Sorting in array: Insertion sort



> **Steps:**
> Swap.

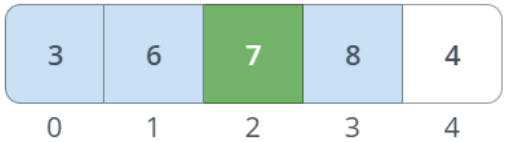
Sorting in array: Insertion sort



Steps:

Processing record in position 3

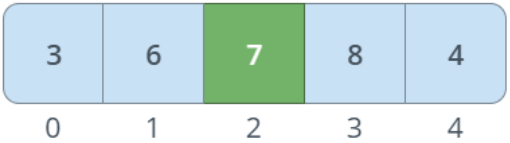
Sorting in array: Insertion sort



Steps:

Move the blue record to the left until it reaches the correct position.

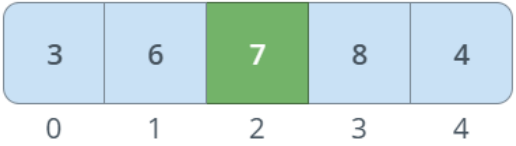
Sorting in array: Insertion sort



Steps:

Processing record in position 4

Sorting in array: Insertion sort



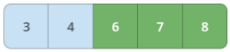
Steps:

Move the blue record to the left until it reaches the correct position.

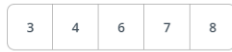
Sorting in array: Insertion sort

Steps:
Swap.

Steps:
Swap.



Steps:
Swap.



Steps:
Done sorting!

Sorting in array: Insertion sort

abc > C insert.c > main()

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter the array elements: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    // Insertion Sort
    for (int i = 1; i < n; i++) {
        int key = arr[i];  // key--> second element
        int j = i - 1; // another variable for comarision (left)
        // Shift elements to right if they are greater than key
        while (j >= 0 && arr[j] > key) { // compare key(2nd element and j=0 left element)
            arr[j + 1] = arr[j]; // change place of 2nd elemt to right side (9-5)
            j--;  // Move left
        }
        arr[j + 1] = key;  // Insert element at that position
    }
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

```

Projects: code1 Debug Any CPU Compile & Run Compile Debug Ln 13, Col 34 Spaces: 4 UTF-8

```
Enter the number of elements: 5
Enter the array elements: 8 7 4 9 2
Sorted array: 2 4 7 8 9
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'insert.exe'
Enter the number of elements:
```

Sub Array

abc > C subarry.c > main()

```c
#include <stdio.h>
int main() {
    int arr[] = {1,2,3,4,5,6,7,8};
    int start = 2, end = 5; // Subarray range
    int subArray[100]; // Array to store subarray
    int j = 0; // Index for subArray
    // Copy elements from arr[start] to arr[end] into subArray
    for (int i = start; i <= end; i++) {
        subArray[j] = arr[i];
        j++;
    }
    printf("Subarray of the given array: ");
    for (int i = 0; i < j; i++) {
        printf("%d ", subArray[i]);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'subarry.exe'
Subarray of the given array: 3 4 5 6
PS F:\code1\abc\output>
```

Sub Array of even numbers

abc > C subarreven.c > main()

```c
int main() {
    int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // Original array
    int size = 10;
    int subArray[100]; // Subarray storage
    int j = 0; // Index for subArray
    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 == 0) {
            subArray[j] = arr[i];
            j++;
        }
    }
    printf("Subarray (Even Numbers): ");
    for (int i = 0; i < j; i++) {
        printf("%d ", subArray[i]);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'subarreven.exe'
Subarray (Even Numbers): 2 4 6 8 10
PS F:\code1\abc\output>
```

Deletion in array

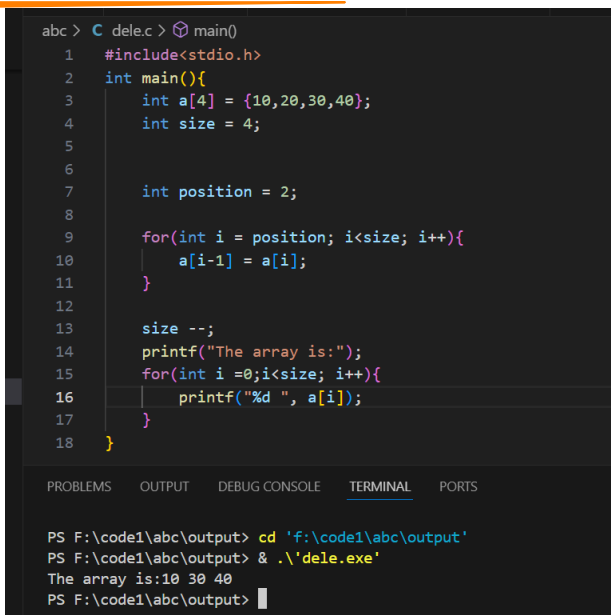
Algorithm

1. Find the given element in the given array and note the index.
2. If the element found,

   Shift all the elements from index + 1 by 1 position to the left.

   Reduce the array size by 1.
3. Otherwise, print "Element Not Found"

Deletion in array



2D array

abc > C matrix2d.c > main()

```c
#include <stdio.h>

int main() {
    int matrix[3][3];

    printf("Enter elements for the 3x3 matrix:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("Enter element at position [%d][%d]: ", i, j);
            scanf("%d", &matrix[i][j]);
        }
    }
    printf("\nThe 3x3 Matrix is:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n"); // New line after each row
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Enter element at position [0][0]: 1
Enter element at position [0][1]: 2
Enter element at position [0][2]: 3
Enter element at position [1][0]: 4
Enter element at position [1][1]: 5
Enter element at position [1][2]: 6
Enter element at position [2][0]: 7
Enter element at position [2][1]: 8
Enter element at position [2][2]: 9

The 3x3 Matrix is:
1 2 3
4 5 6
7 8 9
```

2D array- sum


```c
int main() {
    int matrix1[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int matrix2[3][3] = {
        {9, 8, 7},
        {6, 5, 4},
        {3, 2, 1}
    };
    int sum[3][3];
    // Calculate the sum of both matrices
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            sum[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
    printf("Sum of the two matrices:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", sum[i][j]);
        }
        printf("\n");
```

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'matrixsum.exe'
Sum of the two matrices:
10 10 10
10 10 10
10 10 10
PS F:\code1\abc\output>
```

2D array- Matrix multiplication

abc > C matrixsum.c > main()

```c
#include <stdio.h>
int main() {
    int matrix1[2][3] = {
        {1, 2, 3},
        {4, 5, 6}
    };
    int matrix2[3][2] = {
        {7, 8},
        {9, 10},
        {11, 12}
    };
    int product[2][2] = {0} ;
    for (int i = 0; i < 2; i++) { //Loop through rows of matrix1
        for (int j = 0; j < 2; j++) { //Loop through cols of matrix2
            for (int k = 0; k < 3; k++) { // Loop through shared dimension (columns of matrix1 / rows of matrix2)
                product[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }
    printf("Product of the matrices:\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            printf("%d ", product[i][j]);
        }
        printf("\n");
```

```
Product of the matrices:
58 64
139 154
PS F:\code1\abc\output>
```

String

%s
Sizeof
strlen

abc > C char.c > main()

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

int main(){
    char alphabet[] = {"ABCDEFGHIJKLMNOPQRSTUVWXYZ"};

    printf("%s",alphabet);

    printf("\n%d",sizeof(alphabet));

    printf("\n%d", strlen(alphabet));
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'char.exe'
ABCDEFGHIJKLMNOPQRSTUVWXYZ
27
26
PS F:\code1\abc\output>
```

String –user input

abc > C chariput.c > main()

```c
#include <stdio.h>

int main() {
    char clgname[50];

    printf("Enter your clg name: ");
    scanf("%s", clgname);
    printf("Hello, %s!\n", clgname);
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'chariput.exe'
Enter your clg name: BVM Engineering college
Hello, BVM!
PS F:\code1\abc\output>
```

clgname automatically points to &clgname[0], so **scanf() already knows where to store input**.

String – Handle space [^\n]

abc > C chariput.c > ...

```c
#include <stdio.h>

int main() {
    char clgname[50];

    printf("Enter your clg name: ");
    scanf("%[^\n]s", clgname);
    printf("Hello, %s!\n", clgname);
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'chariput.exe'
Enter your clg name: BVM Engineering college
Hello, BVM Engineering college!
PS F:\code1\abc\output>
```

Read a string containing any character

String operations- Strcpy()

abc > C strcpy.c > ...

```c
#include <stdio.h>
#include <string.h>
int main() {
    char city[] = "Colcutta";
    char mycity[] = "Mumbai";

    printf("city before copy: %s\n", city);

    strcpy(city, mycity);  // Copy 'mycity' into 'city'

    printf("City after copying mycity into city, new city: %s\n", city);

    printf("Len of city after copying: %d ", strlen(city));

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'strcpy.exe'
city before copy: Colcutta
City after copying mycity into city, new city: Mumbai
Len of city after copying: 6
PS F:\code1\abc\output>
```

String – strcpy()

abc > C strcpy.c > main()

```c
#include <stdio.h>
#include <string.h>
int main() {
    //char city[] = "Colcutta";
    //char mycity[] = "Mumbai";

    char city[10];
    char mycity[10];

    strcpy(city, "colcutta");
    strcpy(mycity, "mumbai");

    printf("city before copy: %s\n", city);

    strcpy(city, mycity);  // Copy 'mycity' into 'city'

    printf("City after copying mycity into city, new city: %s\n", city);

    printf("Len of city after copying: %d ", strlen(city));

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
City after copying mycity into city, new city: Mumbai
Len of city after copying: 6
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'strcpy.exe'
city before copy: colcutta
City after copying mycity into city, new city: mumbai
Len of city after copying: 6
```

String – strcat()

abc > C strcpy.c > main()

```
3   int main() {
7       char city[20];
8       char mycity[20];
9
10      strcpy(city, "colcutta");// 8
11      strcpy(mycity, "mumbai");//6
12
13      printf("city before copy: %s\n", city);
14
15      strcat(city, mycity);
16
17      printf("City after appending mycity into city, new city: %s\n", city);
18
19      printf("%d",strlen(city));
20      printf("\n%d", sizeof(city));
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'strcpy.exe'
city before copy: colcutta
City after appending mycity into city, new city: colcuttamumbai
14
20
PS F:\code1\abc\output>
```

String – Find uppercase in string


```c
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main(){

    char str[20];
    int count=0, n;
    scanf("%[^\n]", str);

    n = strlen(str);

    for(int i = 0;i<n;i++){
        if(isupper(str[i])){
            printf("%c\n", str[i]);
            count++;
        }
    }
    printf("\nThe number of count is : %d", count);
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'upper.exe'
Hello World
H
W

The number of count is : 2
PS F:\code1\abc\output>
```

String – strcmp

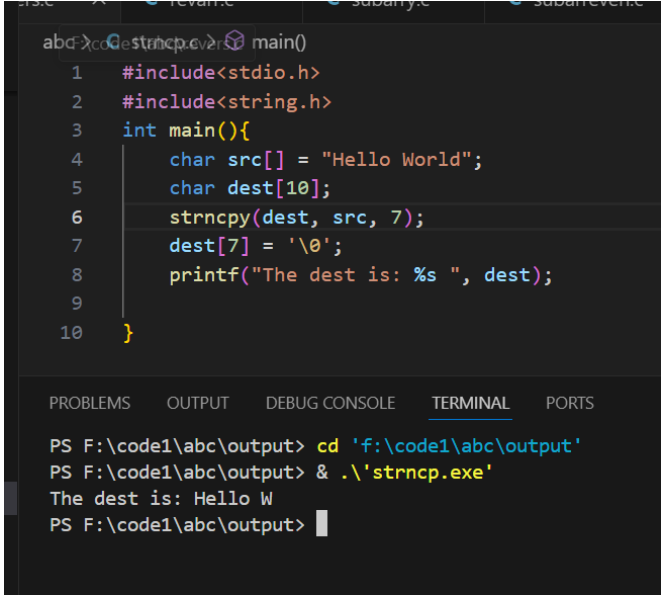
abc > C stcmp.c > main()

```c
#include<string.h>
int main(){
    char a[] = "hello";
    char b[] = "hello";
    char c[] = "hi";

    printf("The compared string is : %d", strcmp(a,b));
    printf("\nThe compared string is : %d", strcmp(a,c));
    printf("\nThe compared string is : %d", strcmp(c,a));

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'stcmp.exe'
The compared string is : 0
The compared string is : -1
The compared string is : 1
PS F:\code1\abc\output>
```

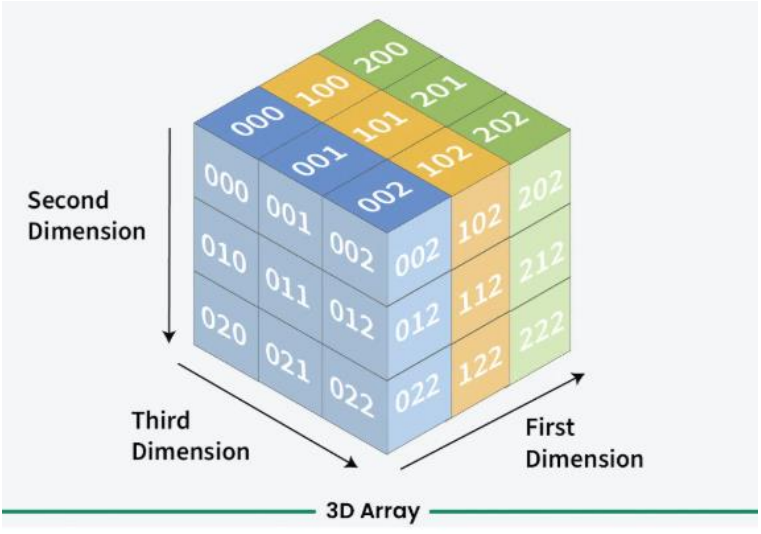
String – strncpy



2D array- Matrix multiplication

	Memory Address	Data	Array Index
Base Address →	40000	24	arr[0][0]
	40004	15	arr[0][1]
	40008	34	arr[0][2]
	40012	26	arr[1][0]
	40016	134	arr[1][1]
	40020	194	arr[1][2]
	40024	67	arr[2][0]
	40028	23	arr[2][1]
	40032	345	arr[2][2]

3D array


Second
Dimension
Third
Dimension
First
Dimension
000
001
002
010
011
012
020
021
022
100
101
102
200
201
202
002
012
022
102
112
122
202
212
222
3D Array

3D array

3d.c > main()

```c
int main() {
    // Create and Initialize the 3-dimensional array
    int arr[2][3][2] = { { { 1, 1 }, { 2, 3 }, { 4, 5 } },
                         { { 6, 7 }, { 8, 9 }, { 10, 11 } } };

    // Loop through the depth
    for (int i = 0; i < 2; ++i) {
        // Loop through the rows of each depth
        for (int j = 0; j < 3; ++j) {
            // Loop through the columns of each row
            for (int k = 0; k < 2; ++k)
                printf("arr[%i][%i][%i] = %d   ", i, j, k,
                       arr[i][j][k]);
            printf("\n");
        }
        printf("\n\n");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\output> cd 'f:\code1\output'
PS F:\code1\output> & .\'3d.exe'
arr[0][0][0] = 1   arr[0][0][1] = 1
arr[0][1][0] = 2   arr[0][1][1] = 3
arr[0][2][0] = 4   arr[0][2][1] = 5


arr[1][0][0] = 6   arr[1][0][1] = 7
arr[1][1][0] = 8   arr[1][1][1] = 9
arr[1][2][0] = 10   arr[1][2][1] = 11
```

[2008 : 1 Mark]

1.57 What is the output printed by the following C code?

```
# include <stdio.h>
int main( )
{
    char a[6] = "world";
    int i, j;
    for (i = 0, j = 5; i < j; a[i++] = a [j – –]);
    printf ("%s\n", a);
}
```

(a) dlrow
(b) Null string
(c) dlrld
(d) worow

[2008 : 2 Marks]

```c
#include <stdio.h>
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]);
}
```

A. 2, 3
B. 2, 4
C. 3, 2
D. 3, 3

Gate 2008

abc > C gate.c > main()

```c
#include <stdio.h>
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++;
        }
        printf("%d", i);
        i--;
        for (j = 7; j > 4; j--) {
            int i = j/2;
            a[i] = a[i] - 1;
        }
        printf ("\n%d, %d", i, a[i]);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\code1\abc\output> cd 'f:\code1\abc\output'
PS F:\code1\abc\output> & .\'gate.exe'
4
3, 2
PS F:\code1\abc\output>
```