

*Tutorial*

Programming & Data Structure: CS 11001

*Section - 4/D*

Department of Computer Science and  
Engineering

I.I.T. Kharagpur

*Spring Semester: 2013 - 2014 (06.03.2014)*

**Download**

**Download the file tut060314.pdf from  
Programming & Data Structures ... of**

<http://cse.iitkgp.ac.in/~goutam>

**View the file using the command `acroread` & or  
`xpdf` &**

## Character String

A character string is a sequence of ASCII values stored in consecutive memory locations (1-D array of type `char`) and terminated by a *null* character `'\0'` (value zero (0)).



## String I/O

```
#include <stdio.h>
#define LEN 100
int main() // stringIO.c
{
    char a[LEN];

    printf("Enter the 1st string: ");
    scanf("%s", a);
    printf("a: %s\n", a);
```

```
printf("Enter the 2nd string: ");  
scanf("%[^\n]", a);  
printf("a: %s\n", a);  
  
printf("Enter the 3rd string: ");  
scanf(" %[^\n]", a);  
printf("a: %s\n", a);  
  
return 0;  
}
```

## string.h

The prototypes of the C library string functions are available in `string.h`. A few examples are, `strcmp()` (string compare), `strcat()` (string concatenation), `strcpy()` (string copy), `strchr` (search for a character), `strlen()` etc.

## String I/O

```
#include <stdio.h>
#include <string.h>
#define SIZE 1000
int main()          // stringExpl.c
{
    char s1[SIZE], s2[SIZE];

    scanf("%[^\n]", s1);
    printf("|%s| = %d\n", s1, strlen(s1));
```

```
strcpy(s2, s1);  
printf("s2: %s\n", s2);  
printf("s1+s2: %s\n", strcat(s1,s2));  
return 0;  
}
```

## Tutorial VIII.1

Write a function

`int strlen(char *)` that returns the length of a string. [Do not use any string function.]

## Tutorial VIII.2

Write a recursive function

`int strLenR(char *)` that returns the length of a string. [Do not use any string function.]

### Tutorial VIII.3

Write a string function `int strMNcat(char *s1, char *s2, int m, int n, char *s)`. The string `s` is formed by concatenating  $k$  length initial portion of `s1` and  $l$  length initial portion of `s2`, where  $k \leftarrow \min(m, \text{len}(s1))$  and  $l \leftarrow \min(n, \text{len}(s2))$ . It returns the length of `s`. [Do not use any library function.]

## 2-D Array

Consider the following declaration of a 2-D array.

```
#define ROW 10
#define COL 20
int main(){
    double data[ROW][COL];
    .....
} // 2DarrayRead.c
```

## Tutorial VIII.4

1. What will be the total number of bytes allocated for the array?

Assume that `sizeof(double)` is 8 bytes.

2. What is the value of `sizeof data`?

## Tutorial VIII.4 (cont.)

1. What is the interpretation of the value of `data`?
2. What are the values of `data+1` and `data+2` if the value of `data` is `x`.
3. What are the values of `*data`, `*(data+1)`, `*data+1`, `*data + 2`, and `*(data+2)`.
4. What are the types of `data` and `*data`?

## Initialisation & Type of 2-D Array **name**

Consider the following declarations.

```
#include <stdio.h>
int main() tutVIII.5.c
{
    double data[10][20]={{1.0,2.0,3.0,4.0},
                        {10.0,20.0,30.0,40.0},
                        {100.0, 200.0, 300.0, 400.0}},
        (*dtwP)[20], *dP ;
```

## Tutorial VIII.5

Assume that the memory address is 32-bit or 4-bytes and answer the following questions.

1. What is the value of `data[2][5]`?
2. What are the sizes of `dtwP` and `dP`?
3. Is it valid to write `dtwP = data+2`?
4. Can we access elements of `data[][20]` using `dtwP`?

## Tutorial VIII.6

Write a function

```
void transposeInPlace(int x[][MAX], int r,  
int c);
```

where  $r$  and  $c$  are the row and column numbers of the matrix stored in the 2-D array. The function transposes the matrix in the same array.

**Example**

$$\begin{bmatrix} 1 & 2 & 3 & - & - & - \\ 4 & 5 & 6 & - & - & - \\ 7 & 8 & 9 & - & - & - \\ 10 & 11 & 12 & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix}^T_{4 \times 3} = \begin{bmatrix} 1 & 4 & 7 & 10 & - & - \\ 2 & 5 & 8 & 11 & - & - \\ 3 & 6 & 9 & 12 & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix}_{3 \times 4}$$