

## Programming and Problem Solving through C Language

### O Level / A Level

## Chapter - 5 : Array

---

### String

- C implements the string data structure using arrays of type char.
- The declaration of a string is the same as declaring a char array.

```
char string_var[30];  
char string_var[20] = "Initial value";
```

### Declare string

```
char s[5];
```

| s[0] | s[1] | s[2] | s[3] | s[4] |
|------|------|------|------|------|
|      |      |      |      |      |

```
char greeting[] = "Hello";
```

| Index    | 0       | 1       | 2       | 3       | 4       | 5       |
|----------|---------|---------|---------|---------|---------|---------|
| Variable | H       | e       | l       | l       | o       | \0      |
| Address  | 0x23451 | 0x23452 | 0x23453 | 0x23454 | 0x23455 | 0x23456 |

### initialize strings

```
char c[] = "abcd";  
char c[50] = "abcd";  
char c[] = {'a', 'b', 'c', 'd', '\0'};  
char c[5] = {'a', 'b', 'c', 'd', '\0'};
```

| c[0] | c[1] | c[2] | c[3] | c[4] |
|------|------|------|------|------|
| a    | b    | c    | d    | \0   |

**Example : scanf() to read a string**

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter name: ");
    scanf("%s", name);
    printf("Your name is %s.", name);
    return 0;
}
```

**Example : Program to read a line of text**

You can use the fgets() function to read a line of string. And, you can use puts() to display the string.

```
#include <stdio.h>
int main()
{
    char name[30];
    printf("Enter name: ");
    fgets(name, sizeof(name), stdin); // read string

    printf("Name: ");
    puts(name); // display string
    return 0;
}
```

**Example : Read & Write Strings in C using gets() and puts() functions**

```
#include <stdio.h>
#include <string.h>
int main()
{
    /* String Declaration*/
    char nickname[20];

    /* Console display using puts */
    puts("Enter your Nick name:");

    /*Input using gets*/
    gets(nickname);

    puts(nickname);

    return 0;
}
```

## Memory Storage for a String

- The string is always ended with a null character '\0'.
- The characters after the null character are ignored.
  - E.g., `char str[20] = "Initial value";`



## Arrays of Strings

- An array of strings is a two-dimensional array of characters in which each row is one string.
  - `char names[People][Length];`
  - `char month[5][10] = {"January", "February", "March", "April", "May"};`

## Input/Output of a String

- The placeholder `%s` is used to represent string arguments in `printf` and `scanf`.
  - `printf("Topic: %s\n", string_var);`
- The string can be right-justified by placing a positive number in the placeholder.
  - `printf("%8s", str);`
- The string can be left-justified by placing a negative number in the placeholder.
  - `Printf("%-8s", str);`

## Manipulating String with `scanf` and `printf`

```
#define STRING_LEN 10

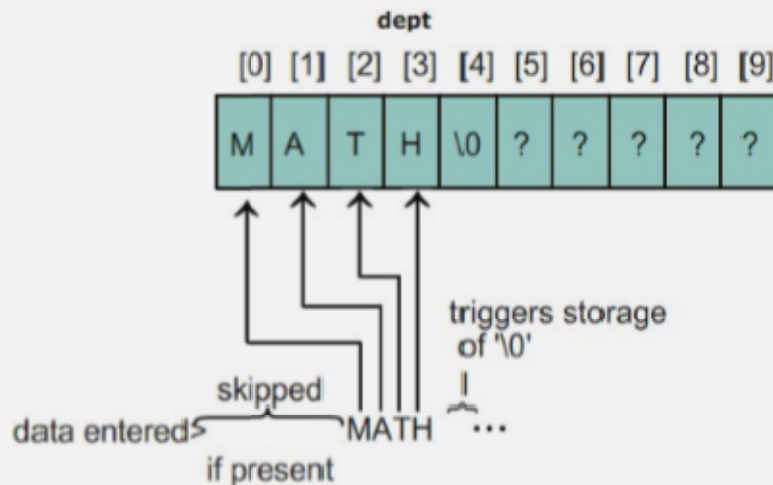
int
main(void)
{
    Char dept[STRING_LEN];
    int Course_num;
    Char days[STRING_LEN];
    int time;

    Printf("Enter department Code, Course number, days and");
    printf("time like this:\n> COSC 2060 MWF 1410\n>");
    Scanf("%s%d%s%d", dept, &Course_num, days, &time);
    Printf("%s %dmeets %s at %d\n", course_num, days, time);

    return(0);
}
```

### Execution of scanf ("%s", dept);

- Whenever encountering a white space, the scanning stops and scanf places the null character at the end of the string.
- E.g., if the user types "MATH 1234 TR 1800," the string "MATH" along with '\0' is stored into dept.



### Program -1 /\*C program to print string one by one characters using loop.\*/

```
#include<stdio.h>

int main()
{
    char text[100];
    int i;

    printf("Enter any string: ");
    gets(text);

    printf("Entered string is: ");
    for(i=0;text[i]!='\0';i++)
    {
        printf("%c",text[i]);
    }
    printf("\n");
    return 0;
}
```

**Program - /\*C program to count upper case, lower case and special characters in a string.\*/**

```
#include<stdio.h>
int main()
{
    char text[100];
    int i;
    int countL,countU,countS;

    printf("Enter any string: ");
    gets(text);

    //here, we are printing string using printf
    //without using loop
    printf("Entered string is: %s\n",text);

    //count lower case, upper case and special characters
    //assign 0 to counter variables
    countL=countU=countS=0;

    for(i=0;text[i]!='\0';i++)
    {
        //check for alphabet
        if((text[i]>='A' && text[i]<='Z') || (text[i]>='a' && text[i]<='z'))
        {
            if((text[i]>='A' && text[i]<='Z'))
            {
                //it is upper case alphabet
                countU++;
            }
            else
            {
                //it is lower case character
                countL++;
            }
        }
        else
        {
            //character is not an alphabet
            countS++; //it is special character
        }
    }

    //print values
    printf("Upper case characters: %d\n",countU);
    printf("Lower case characters: %d\n",countL);
    printf("Special characters: %d\n",countS);

    return 0;
}
```

**Program -/\*C program to convert string in upper case and lower case.\*/**

```
#include<stdio.h>

int main()
{
    char text[100];
    int i;

    printf("Enter any string: ");
    gets(text);

    printf("Entered string is: %s\n",text);

    //convert into upper case
    for(i=0;text[i]!='\0';i++)
    {
        if(text[i]>='a' && text[i]<='z')
            text[i]=text[i]-0x20;
    }
    printf("String in Upper case is: %s\n", text);

    //convert into lower case
    for(i=0;text[i]!='\0';i++)
    {
        if(text[i]>='A' && text[i]<='Z')
            text[i]=text[i]+0x20;
    }
    printf("String in Lower case is: %s\n",text);

    return 0;
}
```

**Assignment**

1. Write a program to read a line of string and print the length of the string.
2. Write a program to read two strings and concatenate them.
3. Write a program to read a line and print the numbers of characters.
4. Write a program to read a line and print the numbers of words.
5. Write a program to read a word and print it reverse.