The %s used in `printf()` is a format specification for printing out a string. The same specification can be used to receive a string from the keyboard.

Sample-1:

```c
#include <stdio.h>
#include <conio.h>
void main()
{
    char name[25];
    clrscr();
    printf("Enter your name ":
    scanf("%s",name);
    printf("Hi %s!
",name);
    getch();
}
```

Note that the declaration `char name[25]` sets aside 25 bytes under the array `name[ ]`, whereas the `scanf()` function fills in the characters typed at keyboard into this array until the enter key is hit. Once enter is hit, `scanf()` places a ‘0’ in the array.

Using `scanf()` we must be cautious about two things:
- The length of the string should not exceed the dimension of the character array. This is because the C compiler doesn’t perform bounds checking on character arrays.
- if you carelessly exceed the bounds there is always a danger of overwriting something important.
- `scanf()` is not capable of receiving multi-word strings.
- Therefore names such as ‘Amit Kumar’ would be unacceptable.

Sample-2: If we try to enter multi-word string with `scanf()` then

```c
#include <stdio.h>
#include <conio.h>
void main()
{
    char name[25];
    clrscr();
    printf("Enter your name ":
    scanf("%s",name);
    printf("Hi %s!
",name);
    getch();
}
```

To overcome this problem `scanf()` accept multi-word strings by writing it in this manner:

```c
#include <stdio.h>
#include <conio.h>
void main()
{
    char name[25];
    clrscr();
    printf("Enter your name ");
    scanf("%s",name);
    printf("Hi %s!
",name);
    getch();
}
```

But it is very irregular type of coding, to avoid this the way to get around this limitation is by using the function `gets()` its counterpart `puts()`.
Sample-3:

The program and the output are self-explanatory, except for the fact that, `puts()` can display only one string at a time (use of two `puts()` in the program above). On displaying a string, unlike `printf()`, `puts()` places the cursor on the next line. Though `gets()` is capable of receiving only one string at a time, the plus point with `gets()` is that it can receive a multi-word string.

Example 1: **Write a program to input any string and print each character in new line.**

```c
#include<stdio.h>
#include<conio.h>
void main()
{
 int i;
 char x[10];
 clrscr();
 printf("Enter any string\n");
 gets(x);
 for(i=0;x[i]!="\0";i++)
 printf("%c\n",x[i]);
 getch();
}
```

Example 2: **Program to find the Length of a string.**

```c
#include<stdio.h>
#include<conio.h>
void main()
{
 int i,j=0;
 char x[10];
 clrscr();
 printf("Enter any string\n");
 gets(x);
 for(i=0;x[i]!="\0";i++)
 j++;
 printf("Length of string is== %d",j);
 getch();
}
```