C-64 ⟹ Strings in C - Part 3

(printf() & puts() function in C)

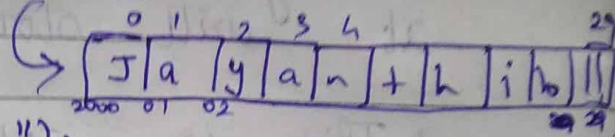name 200

Char name [30];
printf (" Enter name : ");
scanf (" %s ", name);
printf (" %s ", name);

⟶ O/P

```
Enter name : Jayanthi
Jayanthi
```

printf (" %.5S", name);

O/P
```
Enter name : Jayanthi

J a y a n
```

printf (" %10.5S, name);

```
Enter name : Jayanthi


_ _ _ _ _ _ _ _ _ _ J a y a n
  10 white spaces              5 character
```

```c
char name [30];
printf (" Enter name: ");
scanf ("%.s", name);
puts (name);
```

> Enter name: Jayanthi
>
> Jayanthi space

* puts() will put a empty space at end of string and new line is inbuilt within put().

* But in printf ("%s \n", name) new line is not inbuilt, we have to specify "\n".

Eg:- printf ("%.s ", name);   (Correct)
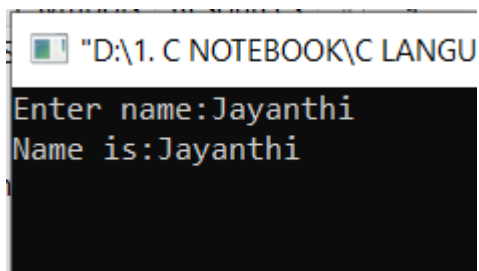
(Same)

Eg:- printf ("%.s", &name);   (Correct)

Eg:- printf ("%.s", &name[2]); ⟹ O/P   (Correct)
                                    → yanthi

Eg:- printf ("%.c", &c[2]); ⟹ O/P   (Correct)
                                 y

Eg:- printf ("%.s", name[2]); ⟹ Error   (wrong)
              ↓
         [No address]

CODE 1:

```c
1    #include <stdio.h>
2    #include <stdlib.h>
3    /** PRINT STRING USING printf() **/
4    int main()
5    {
6        char name[30];
7        printf("Enter name:");
8        scanf("%s",name);
9        printf("Name is:%s",name);
10       getch();
11   }
12
```

"D:\1. C NOTEBOOK\C LANGU

```
Enter name:Jayanthi
Name is:Jayanthi
```
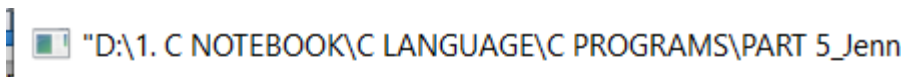
CODE 2:

```c
1    #include <stdio.h>
2    #include <stdlib.h>
3    /** PRINT STRING USING printf() **/
4    int main()
5    {
6        char name[30];
7        printf("Enter name:");
8        scanf("%s",name);
9        printf("Name is:%.5s",name);
10       /** prints only 5 characters **/
11       getch();
12   }
13
```

```
Enter name:Jayanthi
Name is:Jayan
```

CODE 3:

```c
1    #include <stdio.h>
2    #include <stdlib.h>
3    /** PRINT STRING USING printf() **/
4    int main()
5    {
6        char name[30];
7        printf("Enter name:");
8        scanf("%s",name);
9        printf("Name is:%10.5s",name);
10       /** prints 10 blank spaces and 5 characters **/
11       getch();
12   }
13
```
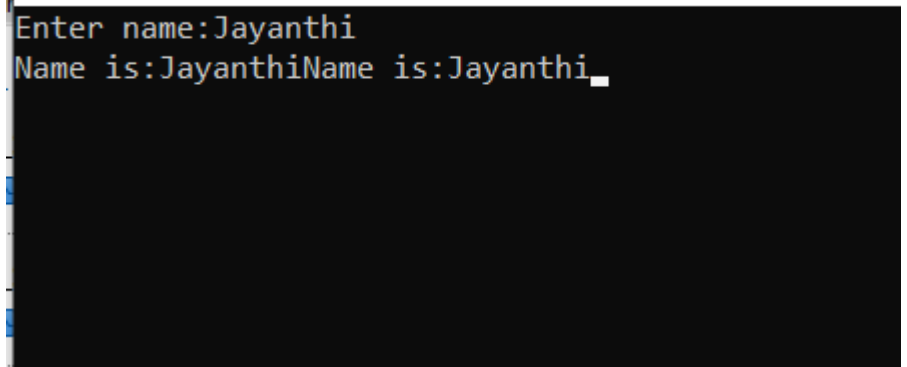
```
Enter name:Jayanthi
Name is:     Jayan
```

CODE 4:

```c
#include <stdio.h>
#include <stdlib.h>
/** PRINT STRING USING puts() **/
/** puts() has inbuilt new line **/
int main()
{
  char name[30];
    printf("Enter name:");
    scanf("%s",name);
    printf("Name is:%s",name);
    printf("Name is:%s",name);

    getch();
}
```

```
Enter name:Jayanthi
Name is:JayanthiName is:Jayanthi_
```

CODE 5:

```c
#include <stdio.h>
#include <stdlib.h>
/** PRINT STRING USING puts() **/
/** puts() has inbuilt new line **/
int main()
{
  char name[30];
    printf("Enter name:");
    scanf("%s",name);
    //printf("Name is:%s",name);
    //printf("Name is:%s",name);
    puts(name);
    puts(name);

    getch();
}
```
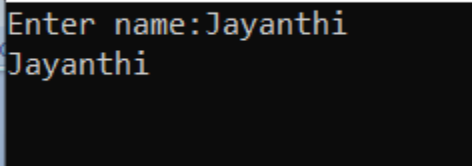
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART

```
Enter name:Jayanthi
Jayanthi
Jayanthi
```

CODE 6:

```c
1    #include <stdio.h>
2    #include <stdlib.h>
3    /** printf() in strings **/
4    int main()
5    {
6        char name[30];
7        printf("Enter name:");
8        scanf("%s",name);
9        printf("%s",&name); /** Both same -> printf("%s",name); **/
10
11       getch();
12   }
13
```

"D:\1. C NOTEBOOK\C LANGU,
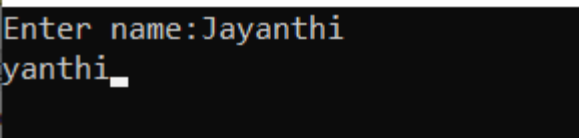
```
Enter name:Jayanthi
Jayanthi
```

CODE 7:

```c
1    #include <stdio.h>
2    #include <stdlib.h>
3    /** printf() in strings **/
4    int main()
5    {
6        char name[30];
7        printf("Enter name:");
8        scanf("%s",name);
9        printf("%s",&name[2]);
10       /** It will printf from index address name[2] till before null character **/
11       /** We read string Jayanthi and prints from yanthi **/
12       getch();
13   }
14
```

"D:\1. C NOTEBOOK\C LANGUAGE\C P|

```
Enter name:Jayanthi
yanthi_
```

CODE 8:

```c
#include <stdio.h>
#include <stdlib.h>
/** printf() in strings **/
int main()
{
    char name[30];
    printf("Enter name:");
    scanf("%s",name);
    //printf("%s",&name[2]);
/** It will print from index address name[2] till before null character here we
    read string Jayanthi and prints from yanthi **/
    //printf("%s",name[2]);
/** It will show no output because string format specifier needs address of index
2 to start form name[2], but here no (&) hence we get no output **/
    printf("%c",name[2]);
/** This is character format specifier, single character no need (&),
hence it gives output for index name[2] **/

    getch();
}
```

■ "D:\1. C NOTEBOOK\C LANGUAGE\C PROGRA

```
Enter name:Jayanthi
y
```