C_63 ⇒ String in C

Part - 2 (Runtime String Initialization)

Read a string using scanf & gets function

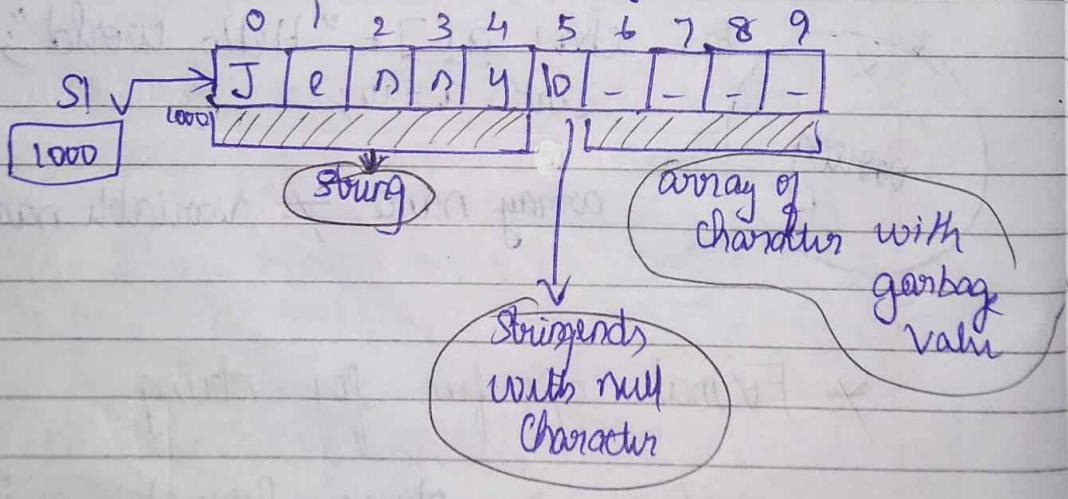
* When we read a string during run time we use scanf() & gets()

* Why Null character :-

↳ Because character array is a array of characters ; and when we store a string ; Compiler will automatically put null character at the end of string to note that string ends and remaing memory would be considered as array of character.

```
char s1[10] = "Jenny";
```

10 bytes



* Format specifier for string is %s

a[10] -

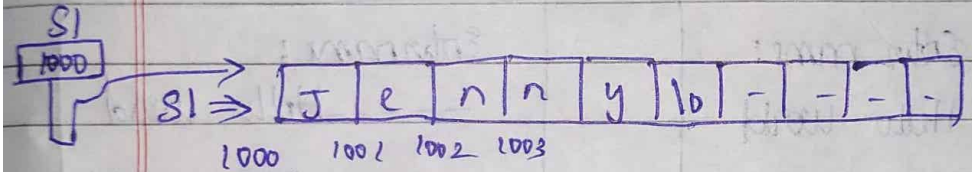
*In array, we use for loop to read a value specific. a[0,1] &a[i,i]

*But string is not a value, it is array of character; a[ ], so we dont use for loop, and address of operator (&).

a[ ]

Eg: ① Char s1[10]; printf ("Enter name:");
scanf ("%s", s1);
printf ("%s", s1);

O/P
Enter name:
Jenny.
Jenny

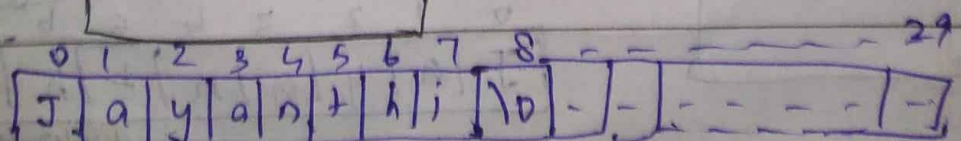


*Drawback in scanf()

Eg ②: Char name[30];
printf ("Enter name:");
scanf ("%s", name);
printf ("%s", name);

O/P
Enter name:
Jagarthi Khatri
white space.
Jayarthi

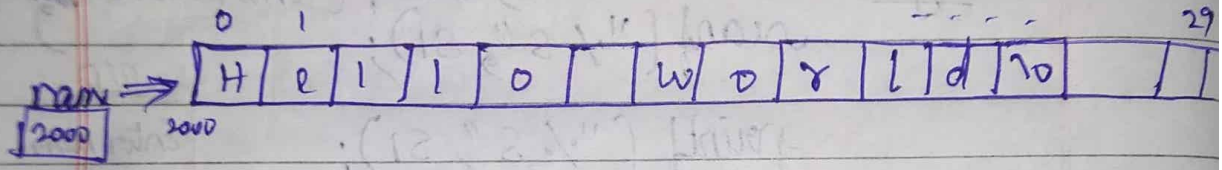
* We get only Jayarthi
* Bcoz scanf() won't consider white space.
* It will end with null.



* gets()

* gets() function will read the entire line until we give 'enter'.

Eg:
```
char name[30];
printf("Enter name:");
gets(name);
printf("%s", name);
```



O/P ①

```
Enter name:
Hello World
Hello word.
```

O/P ②

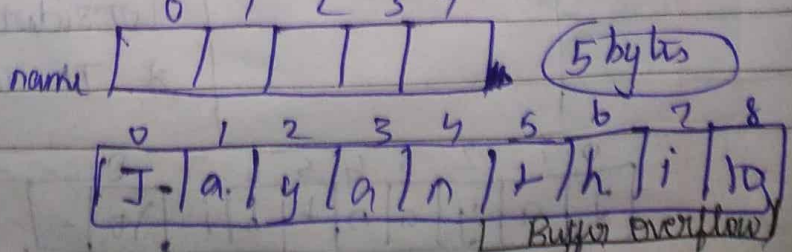
```
Enter name:
-----Hello World
---- Hello World
```

* Drawback in scanf() & gets() :-

Eg:
```
char name[5];
printf("Name:");
scanf("%s", name);
(or)
gets(name);
```

(5 bytes)

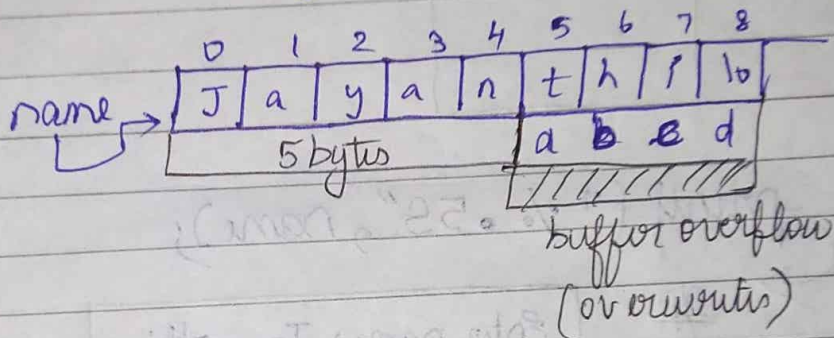
```
Name:
Jayanthi
```



* Eventhough only 5 bytes allocated; both scanf() and gets for string will try to read the maximum we give more than 5 bytes and it will store all the values and also points

* This is called buffer overflow.

* If some other values are stored beyond the 5 bytes; this will overwrite those values and our information will be changed & problem arises.



* Alternative → scanf("%.4s", name);

char name[5]; name → 0 1 2 3 4 | J | a | y | a | \0 | (5 bytes)

```
Enter name : Jayanthi
Jaya
```

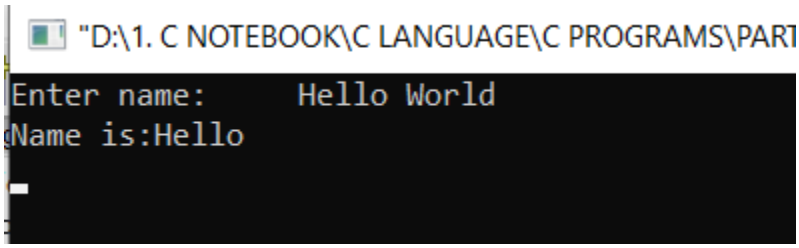
Reads only 4 characters
Stored in byte
+1 character for null
5 bytes.

CODE 1:

```c
#include <stdio.h>
#include <stdlib.h>
/** Read string using scanf() **/
int main()
{
    char name[30];
    printf("Enter name:");
    scanf("%s",name);
    printf("Name is:%s\n",name);

    getch();
}
/** scanf() wont consider white spaces, it will get only single string name
    which is entered first and after that if any space it will terminate
    after reading one string name **/

    /**     Hello World -> Hello **/
```



CODE 2:

```c
#include <stdio.h>
#include <stdlib.h>
/** Read string using gets() **/
int main()
{
    char name[30];
    printf("Enter name:");
    gets(name);
    printf("Name is:%s\n",name);

    getch();
}
/** gets() will read entire line until we give ENTER and it will also
    print the white spaces inside the string, it will read array of string.. **/

    /**     Hello World ->      Hello World**/

```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Le
Enter name:     Hello World
Name is:     Hello World
```

CODE 3:

```c
#include <stdio.h>
#include <stdlib.h>
/** Alternative for buffer overflow **/
int main()
{
    char name[5];
    printf("Enter name:");
    scanf("%4s",name); /** prints only 4 characters **/
    printf("%s",name);
    getch();
}

/** only 5 bytes allocated but i am giving input more than 5 bytes, then specify
literal for 4 characters in formate specifier, it will print only 4 characters
are printed....**/
```