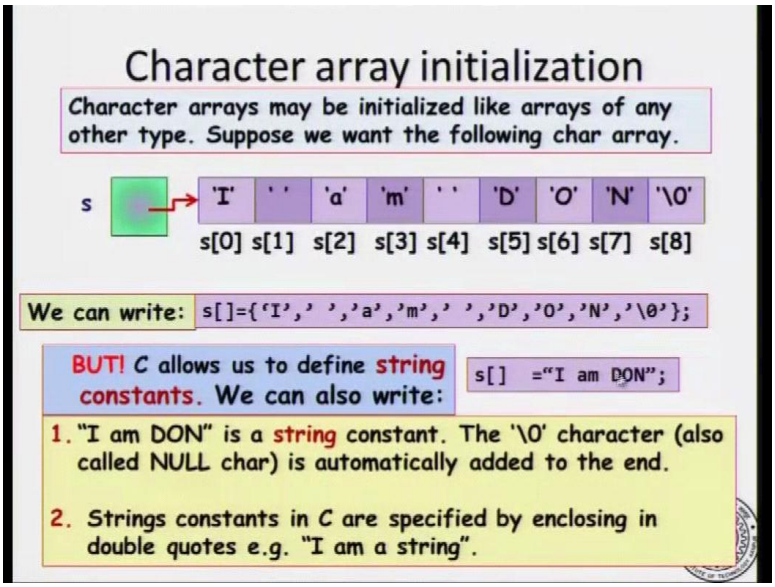
Introduction to Programming in C
Prof. Satyadev Nandakumar
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 31

(Refer Slide Time: 00:05)

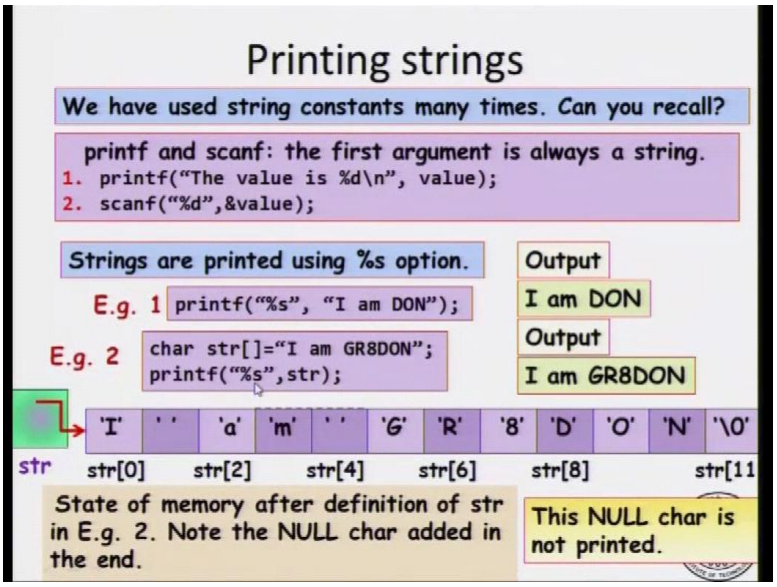


In this video, will talk about initializing character arrays which are a special kind of arrays and has more features then, when it comes to initialization as compared to arrays of other type. So, they may be initialized just like any other array and suppose I want to initialize an array to the following values, the first is I second is a space character then a then m and so on. And the final character is a null character which will be given as back slash 0 within single quotes now we can write s. So, character s equal to and with just the empty pair of square brackets without specifying a size and then followed by the list of characters. Notice that each character is enclosed in single quotes right.

So, the space character is a space enclosed in a single quote and so on until the last which is a null character, which is back slash 0 enclosed in a pair of single quotes. But C also allows us to define what are known as string constants. So, in order to initialize an array a character array I can also write character s square brackets is equal to I am don, but this time within, a pair of double quotes. So, I am don is a string constant in a every string constant that is a string enclosed in double quotes the null character is automatically added to the end. So, I want you to note this difference that, here we explicitly gave a

null character at the end here we do not have to give that. Now C; C string constant are specified by enclosing some text with in double quotes for a example, I am a string with in double quotes.

(Refer Slide Time: 02:18)

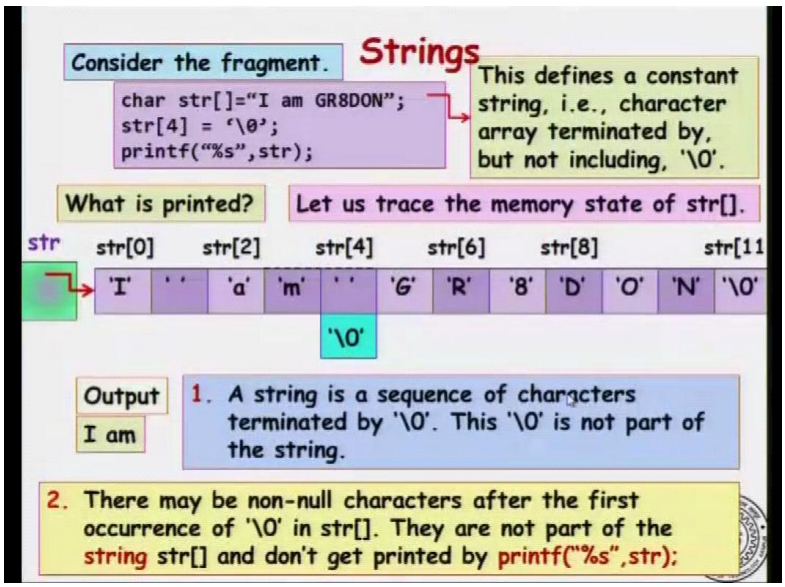


Now, we will talk about how do we prints strings we have used sting constants many times. So, just take a moment to think back to see whether you can figure out where we have used string constants. So, we have used for them for example, in printf and scanf the first argument of a printf or a scanf was always a string constant, because if you recall we had some text in which involved special characters like new line. It involves formats specifiers as like percentage d, but whatever it was, it was a bunch of characters. So, it was a text inside a pair of double quotes that is a string constant. So, the first argument is the string constant followed by what all arguments we want to print. Similarly, even for scanf we had some say formats specifier enclosed in double brac double quotes, so, that is a string constant and then you say and value. So, strings are printed using the percentage s option. So, any of the basic data types in C can be easily printed using the print f statement if you give the correct format specifier.

So, if you have a string constant you can print it using the percentage s option. For example, if I want to print the string I am don with in double quotes then what I can do is I can say printf percentage s I am don. And this will print I space am space don which is exactly what I wanted to print. Now, what if I initialize a character array character str

open close square bracket without specifying the size. I initialize it using a string constant I am great don within double quotes. Then I print it using printf percentage s str, will this work? And the answer is yes it will work, because C will consider this as a string constant and it will print it using percentage s and you will get the correct output. So, state of memory after definition of this string in example two is that, it has a list of all these characters I space a m space and so on and note the implicit null at the end. So, even though the double quotes ended just after n when you stored it in an array there is an implicit null that is inserted at the end of the array. So, when you print it will print until the null character. So, null character itself at the end of the string is not printed when you print it using percentage s.

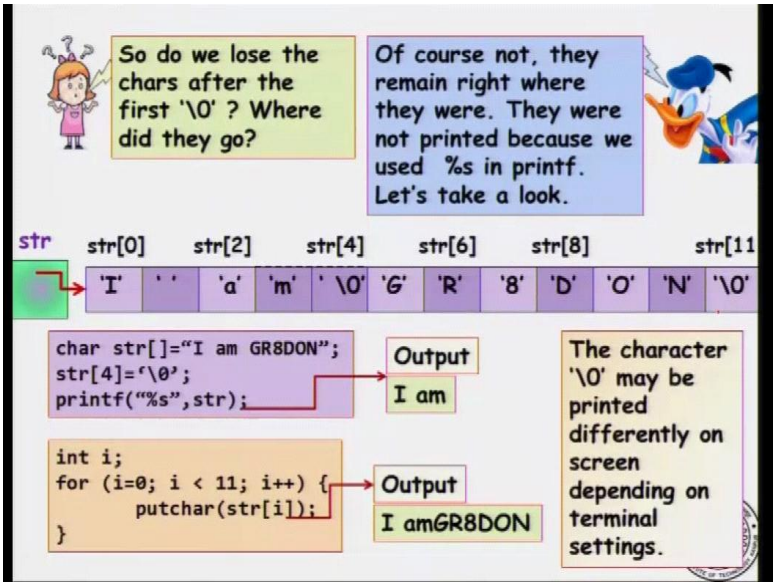
(Refer Slide Time: 05:36)



Now, let us look at the following fragment to understand slightly in a deeper way what percentage s thus when you print it using printf. So, suppose I declare an character array using character str square bracket equal to I am great don with in double quotes. So, this is initialized using the string constant which means that after the last end, there will be a null character in the array. Now, I initialize I i said str 4 equal to null note that there are 11 non-null characters in the string constant. So, this goes from str 0 to str 10 followed by str 11 which is a null character. So, now, I said str 4 equal to null. So, somewhere in the middle of the string I put a null character. What will happen if I print it using printf percentage s? So, let us see what happens here I declare the array and initialize it using a string constant.

So, it has all these letters followed by a null at the end. Then when I said str 4 equal to null what it does is it goes to the fourth location in the array and changes that to null. So, what that does is there was a space there before, but now you insert a null character there. After the null character there are other non-null characters and then there is a second one. What will happen when you print? It will just print I am and stop that it will not print the remaining characters and why does that happen? So, string of C as for as C is concerned is a sequence of characters terminated by a null, this null is not part of the string. So, they may be non-null characters after the first occurrence of null in str, but they are not consider part of the string str their part of the character array. But when you look at str as a string it is just till the first null character. So, when you print it using percentage s only the part until the first null is printed. So, that is considered the string the character array is bigger.

(Refer Slide Time: 08:07)



So, it will just print I am and stop there. So, do I lose the characters after the first null and where do they go? Well, of course they do not go anywhere they remain where they were. So, what is the new state of the array? The new state of the array is I space am and then there is a null and then there are some other characters. So, if I print it using percentage s it will only come up to I am and then stop there. So, is there any way to print the remaining characters? Of course, there is a way right. So, if I print that using percentage s I will get I am, but I could easily write a loop like this. I will say int I and then for I equal to 0 until 11 I plus plus and then putchar str i. So, this will print the

character str 0 str 1 and so on up to str 11, regardless of whether that character is null or not if it is null it will do something, but it will still go on to the next character.

If you run this what you will see is, it will print the first character which is I, then it will print the second character which is space, then it will print the third character which is a. So, these three are printed as they are and then m and the fifth is a null character. What do you mean by printing a null character? It may not print anything. So, it may be just kept, but then it goes on to the next character g r 8 d o n and there it stops, because it does not print the eleventh character. So, the null character in this example is not printed. Now, the way the null character is treated on different terminals may be different. So, on some Linux terminals if you ask to print null character it will just not print anything, but other character terminals may print them in different ways.