STRINGS

CONSIDER THE FOLLOWING SCENARIO

- Suppose, we want to store the name of a student:
- What will be the datatype?
  - Multiple characters
  - All characters must be stored consecutively in the memory
  - Therefore we need an array of characters
  - An array of characters is called as a STRING.
  - Examples of a string:
    - “India”, “Hello World”, “123456”
  - All string constants are enclosed within a “ “.

STRINGS

- Declaration of a string:
  - Syntax

    char array_name[MAX_SIZE];
  - Example:

    char str[20];
  - This creates an array of 20 characters.
  - Since each character is of 1 byte, the total size of the array is 20 bytes.

STRINGS

- Initialization of a string in the line of declaration:
- Syntax:
- There are two possible ways:

char array_name[SIZE]="string_constant";

OR

char array_name[SIZE]={comma separated character values };

- Example:

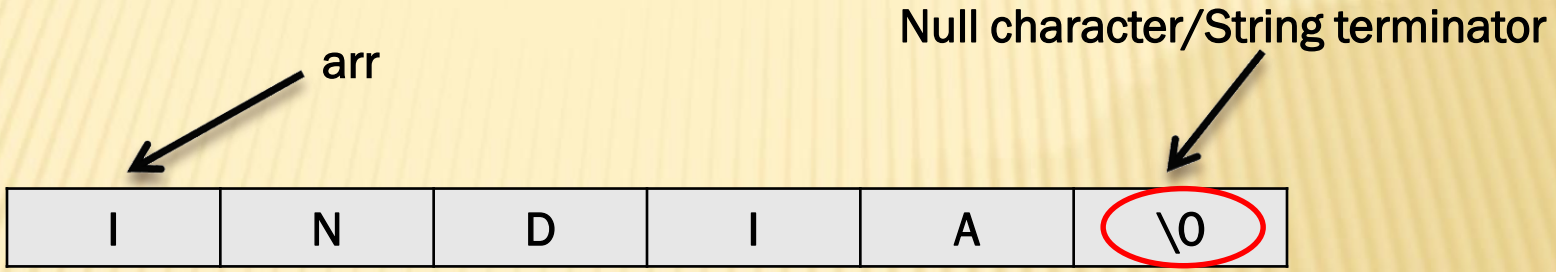
char str[20]="Hello India";

char str[20]={ 'H','e','l','l','o'};

MEMORY REPRESENTATION OF STRINGS

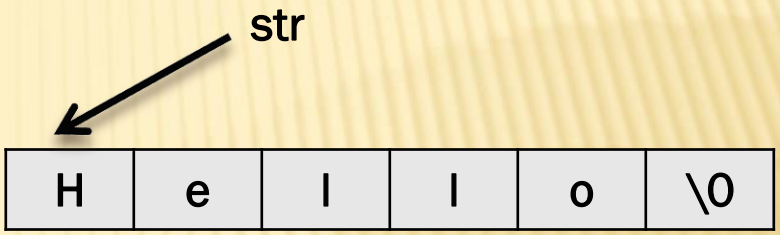
- Consider we have the following string :

  char arr[10]="INDIA";



- Every string constant ends with a null character ('\0').
- Each character in a string is stored in it's ASCII form.
- The ASCII value of a '\0' is zero.
- The null character is an escape sequence as it starts with a '\'.
- The actual size of any string constant is len+1, where len is the number of characters plus one byte for the null character.
- Example: The total size in bytes of the string "INDIA" is 6 bytes.

PRINTING STRINGS



Example:

char str[20]="Hello";

printf("%s", str); ⟶ Hello

printf(%c , str[2]); ⟶ l

- %s format specifier is exclusively to print an entire string.
- %s must be replaced with the name of a character array.
- In the above example, let us assume str is a constant pointer pointing to address 715.(since it is an array name)
- When we use %s, it starts from the first character (at address 715) and prints each character upto the '\0'.
- However if we want to print a single character of a string, we must use a %c.

QUICK EXERCISE

```
char str[20]="Hello";
printf("%s",str);
printf("%c", str[0]);
str[2]='y';
printf("%s", str);
str[3]-=32;
str[4]=65;
printf("%s", str);
```

→ Hello

→ H

→ Heylo

→ HeyLA

QUICK EXERCISE
(ASSUME ADDRESS OF FIRST CHARACTER = 715)

```
char str[20]="Hello";
printf("%u\n",str);
printf("%u\n", str+1);
printf("%s", str+1);
char *p;
p=str;
printf("%u\n",p);
printf("%c\n", *p);
printf("%s", p+2);
```

715

716

ello

715

H

llo

QUICK EXERCISE

```
char str[20]="Hello to all";
str[6]='T';
printf("%c%c%c", str[4], str[5], str[6]);
```
→ o T

```
str[12]='s';
printf("%s\n", str);
```
→ "Hello To alls" followed by some garbage value as the '\0' has been replaced

```
str[7]=0;
printf("%s\n", str);
```
→ Hello T

USER INPUT OF A STRING

- Unlike integer arrays, we need not use a “for” loop to input an entire string.
- It can be done in a single statement using any one of the following two methods:

Method 1	Method 2
char str[20]; scanf(“%s”, str);	char str[20]; gets(str); //No format specifiers
Cannot be used to input strings which have a ‘ ‘ space in them. Only a single word strings can be input.	Can be used to input strings with spaces also. Can be used to input sentences with multiple words.

ACCESSING EACH CHARACTER OF A STRING ONE BY ONE

- If we want to individually access every character of a string element by element, we must use the for loop.
- Example(Try it yourself):

  Input a string and print each character on a separate line.

```
char str[20];
gets(str);
for(i=0;str[i]!='\0';i++)
{
printf("%c\n", str[i]);
}
```

PROGRAMS

- Input a string and print the length of the string (Exclude the null character)
- Input a string and convert the string into uppercase.
- Input a string and count the number of words in the string
- Input a string and convert first letter of each word into uppercase
- Program to copy one string into another character by character.