

**CONFIDENTIAL**

# **C Programming Introduction**

## **week 13: Strings**

**Lecturer : Do Quoc Huy**  
**Dept of Computer Science**  
**Hanoi University of  
Technology**

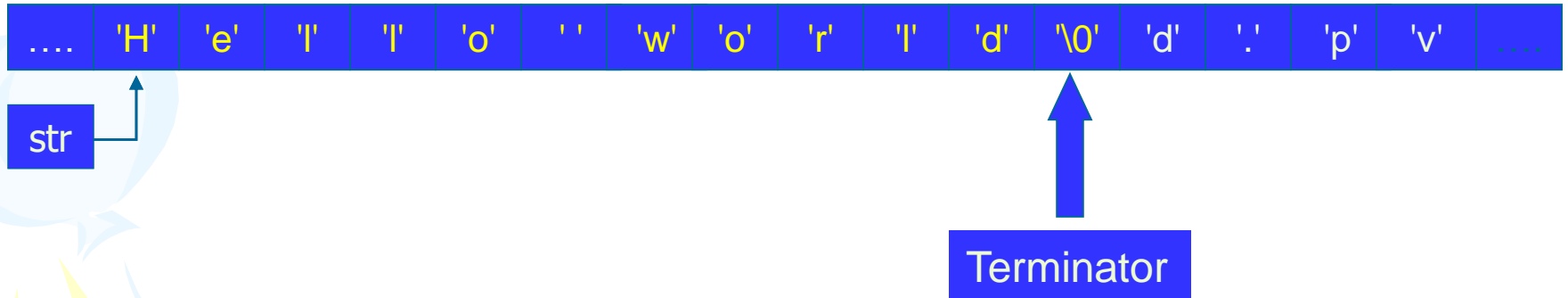


# Strings

- An array of characters
- Used to store text
- Another way to initialize:

```
char str[] = "Text";
```

# The terminator





# The Terminator

- Strings terminate with NULL character, signed by `'\0'` (ascii code 0)
- This is a convention used to know where the string ends
- It means that in order to hold a string of N characters we need an array of length  $N + 1$
- So the previous initialization is equivalent to

```
char str[] = {'b', 'l', 'a', 'b', 'l', 'a', '\0'};
```



# String library

- Like in the case of `stdio.h` and `math.h`, we have a special library for handling strings
- We should **#include** `<string.h>`



# String library

- Functions:

- `strlen(const char s[])`

- returns the length of s

- `strcmp(const char s1[],  
const char s2[])`

- compares s1 with s2

- `strcpy(char s1[],  
const char s2[])`

- copies to contents of s2 to s1

- and more...

# Example

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[ 20 ] = "Happy ";
    char s2[] = "New Year ";
    char s3[ 40 ] = "";
    printf( "s1 = %s\ns2 = %s\n", s1, s2 );
    printf( "strcat( s1, s2 ) = %s\n", strcat( s1, s2 ) );
    printf( "strncat( s3, s1, 6 ) = %s\n", strncat( s3, s1, 6 ) );
    printf( "strcat( s3, s1 ) = %s\n", strcat( s3, s1 ) );
    return 0;
}
```

```
s1 = Happy
s2 = New Year
strcat( s1, s2 ) = Happy New Year
strncat( s3, s1, 6 ) = Happy
strcat( s3, s1 ) = Happy Happy New Year
```

# String Conversion Functions

- Conversion functions
  - In `<stdlib.h>` (general utilities library)
- Convert strings of digits to integer and floating-point values

Prototype	Description
<code>double atof( const char *nPtr )</code>	Converts the string <b>nPtr</b> to <b>double</b> .
<code>int atoi( const char *nPtr )</code>	Converts the string <b>nPtr</b> to <b>int</b> .
<code>long atol( const char *nPtr )</code>	Converts the string <b>nPtr</b> to long <b>int</b> .



# Character Analysis and Conversion

Functions (ctype.h)	Description
<code>isalpha</code>	Check if the argument is a letter
<code>isdigit</code>	Check if the argument is one of the ten digits
<code>isspace</code>	Check if argument is a space, newline or tab.
<code>tolower</code>	Converts the lowercase letters in the argument to upper case letters.



# String conversion function

Prototype	Description
<code>double atof( const char *nPtr )</code>	Converts the string <b>nPtr</b> to <b>double</b> .
<code>int atoi( const char *nPtr )</code>	Converts the string <b>nPtr</b> to <b>int</b> .
<code>long atol( const char *nPtr )</code>	Converts the string <b>nPtr</b> to long <b>int</b> .



# Arrays of Strings

- An array of strings is a two-dimensional array of characters in which each row is one string.

- `char names[People][Length];`

- `char month[5][10] = {"January",  
"February", "March", "April",  
"May"};`

Three balloons (green, blue, and purple) with yellow streamers are positioned on the left side of the slide.

# Exercise 13.1

- Write a program that inputs a line of text, counts the number of blanks by using a function, and displays the number of blanks.



# Exercise 13.2

- write a function that:
  - gets a string and two chars
  - the function scans the string and replaces every occurrence of the first char with the second one.
- write a program to test the above function
  - the program should read a string from the user (no spaces) and two characters, then call the function with the input, and print the result.
- example
  - input: “papa”, ‘p’, ‘m’
  - output: “mama”



## Exercise 13.3

- Write a program that tests a customer number to determine whether it is in the proper format(LLLNNNN with LLL are letters and NNNN are numbers).



## Exercise 13.3

- Write your own replacement for the standard **strcpy()** that comes with C without using `string.h`



## Exercise 13.4

- Write a program asks the user to enter his or her first and last names, separated by a space. Then print out the first name.
- The program should use a function which cuts off the last name off the string in parameter.





## Exercise 13.5

- Write the function `strend(s,t)`, which returns 1 if the string `t` occurs at the end of the string `s`, and zero otherwise.

# Exercise 13.6: Using strstr

- A list of product number and description of shop is:

"TV127 31 inch Television",

"CD057 CD Player",

"TA877 Answering Machine",

"CS409 Car Stereo",

"PC655 Personal Computer"

- Store this list in an array of string and write a program allowing user to lookup a product description by entering all or part of its product number.



## Exercise 13.7

- Write a program that accepts a string from the user and replaces all punctuation signs ( , . ; : ! ? ) with spaces