# CSE 142 Computer Programming I

Strings

... or, "the Itl prgrmr wh cld."

T-1

T-3

© 2000 UW CSE



Concepts this lecture String constants Null-terminated array representation String library <string.h> String initializers Arrays of strings

T-2

#### **Chapter 9**

Read Sections 9.1, 9.2, and 9.4:

9.1: String Basics

Table 9.1 for summary of common functions

9.2: String Assignment

9.3: String Concatenation

9.4: String Comparison

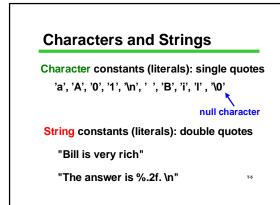
## Character Data in Programs

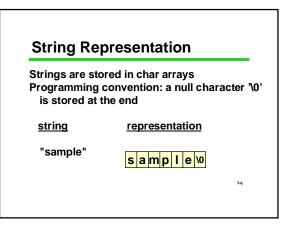
Names, messages, labels, headings, etc.

All of these are common in computer applications

All involve characters: usually multiple characters

So far, our ability to handle these things in C is very limited  $$_{\rm r4}$$ 





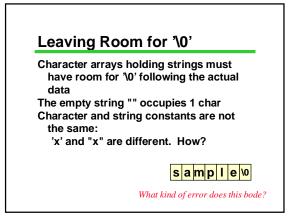


'\0' is not included in strings automatically

'\0' is included in string constants automatically

Programmer must take pains to be sure '\0' is present elsewhere when needed

sample v



#### String Operations

Common needed operations: Copy (assignment) Compare Find length Concatenate (combine strings) I/O Unfortunately...

sample\0

## What You Can't Do

Strings are arrays

They have the limitations of arrays

Can't assign one string to another with =

Can't compare strings with ==, <=

But there are library functions to help do such things

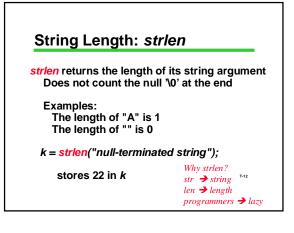
sample\0

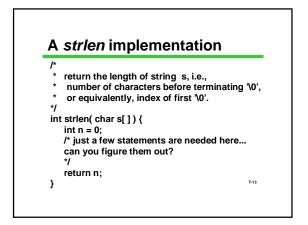
# String Library: <string.h>

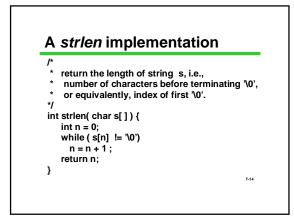
Standard C includes a library of string functions use #include <string.h> Library functions: Require proper null-terminated ('\0') strings as arguments Produce null-terminated strings as results (usually)

sample \0 But... they don't check bounds for you!

Why not?







## String Assignment: *strcpy*

strcpy(dest, source);

Copies characters from *source* to *dest* Copies up to, and including the first '\0' found Be sure that *dest* is large enough to hold the result!

T-15

String Assignment: Examples

#include <string.h>

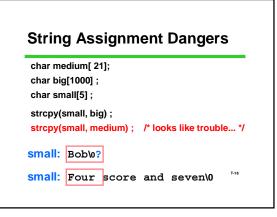
char medium[21] ; char big[1000] ;

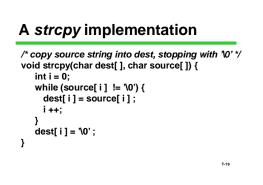
char small[5] ;

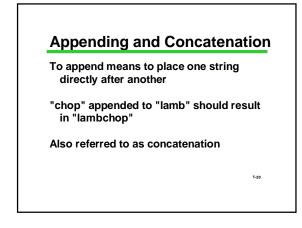
strcpy(medium, "Four score and seven" ) ;

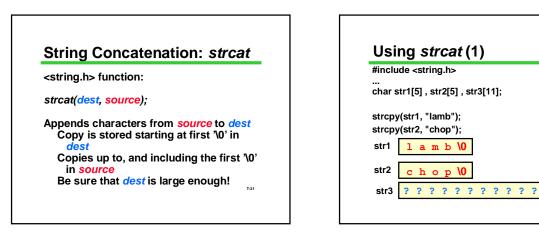
medium: Four score and seven\0

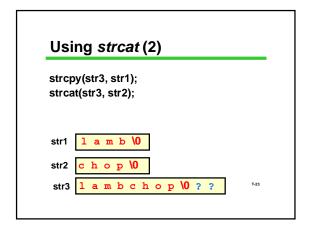
Examples	
char char strc	medium[21]; big[1000]; small[5]; py(big, medium); py(big, "Bob");
	Four score and seven 0?????

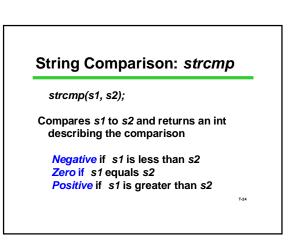




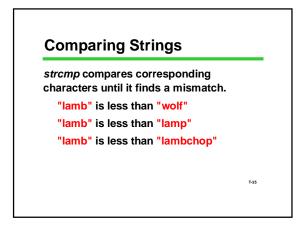


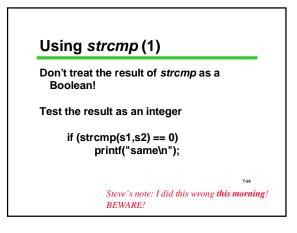






T-22





#### Using strcmp(2)

If you treat the result of *strcmp* as a Boolean, it probably won't do what you want

> if (strcmp(s1,s2)) printf("yikes!");

prints yikes if s1 and s2 are different!

T-27

## String I/O

scanf and printf can read and write C strings Format code is %s

printf assumes '\0' is present

scanf will automatically insert '\0' at the
end
Be sure the array has room for it!

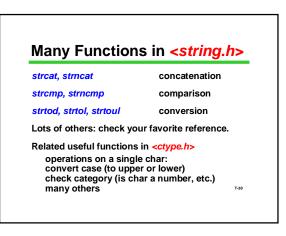
T-28

#### Spot the Security Hole

#define MAX\_INPUT 200 char buffer [MAX\_INPUT];

scanf("%s", buffer);

Never happen? Doesn't matter? <sup>T29</sup> Ever heard of the Internet Worm?



#### **Using Libraries of Functions**

To use strings effectively in C, use functions from string.h

Using libraries is very typical of C programming ANSI C standard libraries such as stdio.h, string.h, ctype.h, math.h

Application-specific libraries: (thousands of them exist)

You can't be an effective programmer without being able to quickly master new libraries of functions

#### Bonus: String Initializers char pet[5] = { 'I', 'a', 'm', 'b', '0' } ; char pet[5]; pet[0] = 'I' ; pet[1] = 'a' ; pet[2] = 'm' ; pet[3] = 'b' ; pet[4] = '0' ; char pet[5] = "lamb" ; char pet[5] = "lamb" ; pet = "lamb" ; /\* No array assignment in C \*/ Remember that initializers are not assignment in C \*/ Remember that initializers are not assignment in C \*/ Remember that initializers are not assignment in C \*/ Remember that initializers are not assignment in C \*/ Remember that initializers are not assignment in C \*/

# Bonus: Arrays of Strings

char month[12][10] = {
 "January",
 "February",
 ...
 "September", /\* longest month: 9 letters \*/
 ...
 "December" } ;
...
printf ("%s is hot \n", month[7] ); /\* August \*/-3

## **Strings Summary**

Definition: Null-terminated array of char

Strings are not fully a type of C They share most limitations of arrays scanf/printf: %s <string.h> library functions Assignment: strcpy Length: str/en strcat and many others

Major Pitfall: overrunning available spaced

## **QOTD: Name the Player**

Strings are often used for names of people and things.

As each player joins a game, how would you choose a name for the player? Each player should have a different, pronounceable, memorable name.