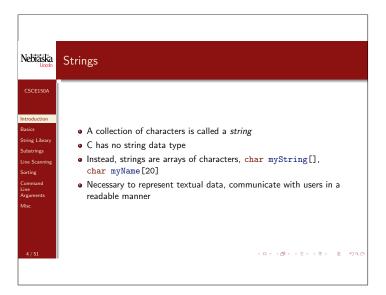


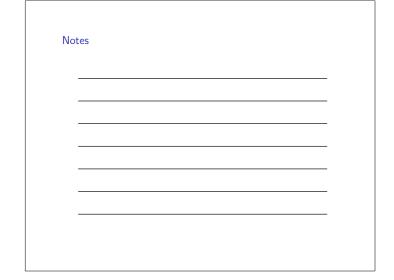
		_

Nebřaska Lincoln	Chapter 9
Introduction Basics String Library Substrings Line Scanning Sorting Command Line Arguments Misc	 9.1 String Basics 9.2 String Library Functions: Assignment and Substrings 9.3 Longer Strings: Concatenation and Whole-Line Input 9.4 String Comparison 9.6 Character Operations 9.7 String-to-Number and Number-to-String Conversion 9.8 Common Programming Errors
2 / 51	(ロ) (중) (문) (문) (문) (문)

Ī	
Nebraska Lincoln	Strings
CSCE150A Introduction Basics String Library Substrings Line Scanning Sorting Command Line Arguments Misc	 Until now we have only dealt with single characters char myChar = 'A', '\n' Processing and manipulating single characters is too limiting Need a way for dealing with groups of characters
3 / 51	(D) (B) (E) (E) & OQ(O

-			





String Basics CSCE150A Introduction Basics String Library Substrings Line Scanning Command Arguments Misc O Calls to scanf or printf used a string constant as the first argument. We have also dealt with static strings: "Hello World!" printf("a = %d\n", a) printf("Average = %.2f", avg) Each string above is a string of 12, 7, and 14 characters, respectively It's possible to use a preprocessor directive: #define INSUFF_DATA "Insufficient Data"

Not	tes		
			_
			_
			_
			_
			_
			_

Nebraska	Static Strings
CSCE150A	
Introduction	
String Library	Static strings cannot be changed during the execution of the program
Substrings	They cannot be manipulated or processed
Line Scanning	May only be changed by recompiling
Sorting Command	Stored in an array of a fixed size
Line Arguments	
Misc	
6 / 51	(ロ) (5) (돈) (돈) 돈 (0)(()

Note	es			
_				
-				
-				
-				
-				
_				

Declaring and Initializing String Variables Strings are character arrays Declaration is the same, just use charchar string_var[100]; char myName [30]; myName will hold strings anywhere from 0 to 29 characters long Individual characters can be accessed/set using indices myName [0] = 'B'; myName [1] = 'r'; myName [2] = 'i'; myName [2] = 'i'; myName [3] = 'a'; myName [4] = 'n'; printf("First initial: %c.\n", myName [0]);

ebiaska _{Lincoln}	Declaring and Initializing String Variables
SCE150A	
roduction sics ing Library bstrings e Scanning ting mmand e guments	 You can declare and initialize in one line Be sure to use the double quotes char myName[30] = "Brian"; You need not specify the size of the array when declaring-initializing in one line: char myName[] = "Brian"; C will create a character array large enough to hold the string
3/51	(D) (B) (E) (E) E - 90(

Nebřaska Lincoln	Null Terminating Character
CSCE150A Introduction Sasics String Library Substrings Library Substrings Command Line Line Line Line Line Line Line Line	 C needs a way to tell where the <i>end</i> of a string is With arrays, it is your responsibility to ensure you do not access memory outside the array To determine where the string ends, C uses the <i>null-terminating character</i>. '\0' Character with ASCII code 0
9 / 51	(B) (B) (E) (E) E MQC

N		
Not	es	
		_
		_
		_
		_
Not	ces	

No	tes		

Nebřaska Lincoln	Null T Example	ermi	natin	ıg Ch	iarac [.]	ter							
CSCE150A													
Introduction Basics String Library Substrings	char : memor		(0) =	"Init	ial '	value	"; wil	l prod	uce th	e follo	wing in		
Line Scanning		[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]		
Sorting Command		[10]	n [11]	[12]	[13]	[14]	a [15]	[16]	[17]	[18]	[19]		
Line Arguments		1	u	e	\0	?	?	?	?	?	?		
Misc	,												
10/51									4 🗆 🕨 👣	Ø → ∢ ≅	() < 분 >	100	গ ৭৫

			-
			_
Г			
	Not	tes	

Notes

Nebraska Lincoln	Arrays of Strings
CSCE150A	
Introduction Basics String Library Substrings Line Scanning Sorting Command Line Arguments Misc	 Without the null terminating character, C would not know where the string ends Many functions parse a string until it sees '\0' Without it, the program would run into memory space that doesn't belong to it char str[20] can only hold 19 characters: at least one character is reserved for '\0' In declarations, char myName[] = "Brian", C automatically inserts
11/51	in declarations, char myname[] = "Brian", C automatically inserts the null-terminating character

Not	es		
			_
			-
			-
			-
			-
			-

Nebraska Lincoln	Printing Strings
CSCEISOA assics tring Library ubstrings ine Scanning orting ommand ine reguments	 You can use printf to print strings Use %s as a placeholder: printf("My Name is %s.\n", myName); printf prints the string until the first null-terminating character Can specify minimum field width, as with e.g. int: printf("My Name is %20s.\n", myName); A negative field width will left justify instead of right justify
12/51	<□> (♂) (≥) (≥) (≥) (≥) (⊘)

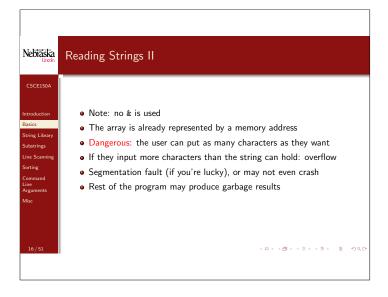
Nebraska Arrays of Strings • One string is an array of characters; an array of strings is a two-dimensional array of characters 1 #define NUM_PEOPLE 30 2 #define NAME_LEN 25 3 . . . 4 char names[NUM_PEOPLE][NAME_LEN]; \bullet names can hold 30 names, each of up to 24 characters long

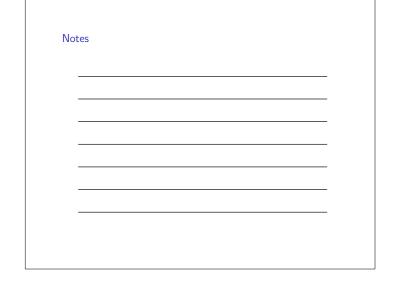
Nebraska Arrays of Strings We can initialize an array of strings at declaration in the following manner: • As with other arrays, the [12] is optional • Why [10]? • September is the longest string with 9 characters • Needs an additional character for the null-terminating character

Nebraska Lincoln	Reading Strings I
CSCE150A	
Introduction Basics String Library Substrings Line Scanning Sorting Command Line Arguments Misc	 You can use scanf and %s to read strings printf("Enter Topic: "); scanf("%s", string_var); scanf skips leading whitespace characters such as blanks, newlines, and tabs Starting with the first non-whitespace character, scanf copies the characters it encounters into successive memory cells of its character array argument When a whitespace character is reached, scanning stops, and scanf places the null character at the end of the string in its array argument
15 / 51	(D)(B)(E)(E) (E) (E)

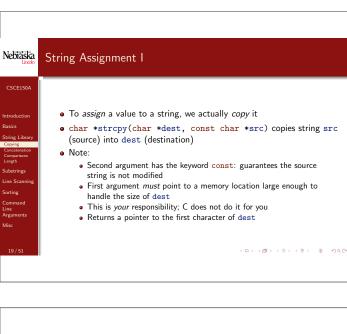
Not	es	
Not	res	
Not	es	

es				
				_
				_
				_
				-
	es	es	es	es





Nebraska Lincoln	String Library
CSCE150A Introduction Basics String Library Coping Conecteration Comparisons Length Substrings Line Scanning Command Line Arguments Misc 18 / 51	 C provides a standard string library Use #include<string.h></string.h> Table 9.1 summarizes which functions are provided Copy, concatenation, comparison, length, tokenizer, etc.



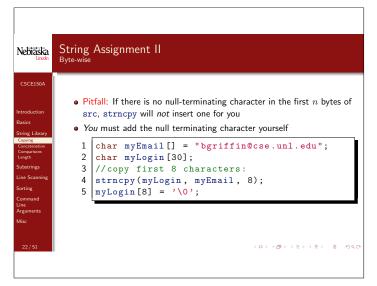
Nebraska Lincoln	String Assignment II
CSCE150A	
Introduction Basics	<pre>1 char myEmail[30]; 2 strcpy(myEmail, "bgriffin@cse.unl.edu");</pre>
String Library Copying Concatenation Comparisons Length Substrings Line Scanning	<pre>Be very careful: 1</pre>
Sorting Command Line Arguments	• In this case, se.unl.edu would overwrite adjacent memory cells
Misc 20 / 51	(B)(B)(E)(E) E DQ

Nebraska Lincoln	String Assignment I Byte-wise
CSCE150A	
Introduction Basics String Library Copying Concatenation Comparisons Length Substrings	 C provides another copying function called strncpy: char *strncpy(char *dest, const char *src, size_t n); size_t is an unsigned integer (no negative value) Copies (up to) n character values of src to dest Actually copies n bytes, but 1 char is one byte
Line Scanning Sorting Command	<pre>char myEmail[] = "bgriffin@cse.unl.edu"; char myLogin[30]; //copy first 8 characters:</pre>
Line Arguments Misc	4 strncpy(myLogin, myEmail, 8);
21 / 51	(ロ)(愛)(言)(言) (言) (言) (つ) (の)

		-
		-
		-
		-
		-

Not		
NO	es	
		-
		- -
		-
		-

Not	es	
		_
		_
		_
		_
		_
		-

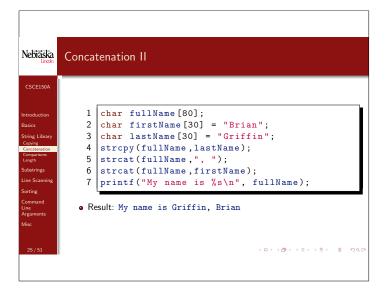


String Assignment III Byte-wise Introduction Basics Basics In is larger than src, the null-terminating character is copied multiple times: string tibray Congruinon Lingsh Substrings Line Scaning Sorting Command Line Arguments Misc 23/51

No				
INO	tes			

Nebraska _{Lincoln}	Concatenation I
CSCE150A Introduction Lasics Itring Library Copying Comparison Length Lubstrings Line Scanning Lioriting Comparison Longth Library	 Concatenation is the operation of appending two strings C provides concatenation functions: <pre>char *strcat(char *dest, const char *src); char *strncat(char *dest, const char *src, size_t n);</pre> Both append src onto the end of dest

Notes			

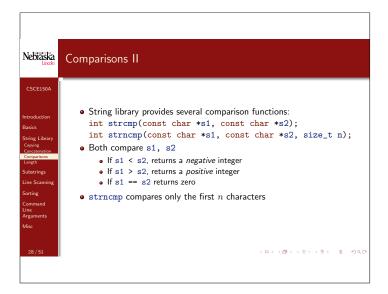


Concatenation III CSCEISOA Introduction Basics String Library Comparation Comparation Length Length Length Sorting Command Line Comma

No	tes			

Nebraska Lincoln	Comparisons I
CSCE150A Introduction Basics String Library Copping Comparison Length Substrings Line Scanning Sorting Command Line Arguments Misc 27 / 51	 We can do character comparisons, 'A' < 'a' We can also do string comparisons (lexicographic order), but not with the usual operators <, > <=, etc. Strings (arrays of characters) are memory addresses string_1 < string_2 would compare the memory locations

		_
		_
		_
		_



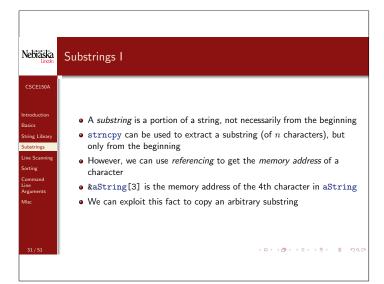
_			
_			
_			
_			
_			

Nebraška Lincoln	Comparisons III
CSCE150A	1
Introduction Basics String Library Copinig Concatenation Comparisons Length Substrings Line Scanning Sorting Command Line Arguments	<pre>1 char nameA[] = "Alpha"; 2 char nameB[] = "Beta"; 3 char nameC[] = "Alphie"; 4 char nameD[] = "BetaFish"; 5 if(strcmp(nameA,nameB) < 0) 6 printf("%s comes before %s\n", nameA, nameB); 7 if(strncmp(nameA,nameC,4) == 0) 8 printf("Almost the same!\n"); 9 if(strcmp(nameB,nameD) < 0) 10 printf("%s comes before %s\n", nameB, nameD);</pre>
Misc 29 / 51	<□> (♂) (≥) (≥) (≥) ≥ ⊅Q(

No	tes	
		-
		-
		-
		-
		-

String Length Throduction Basics String Library Copying Concateration Comparison Command Line C

-			
-			
-			
-			

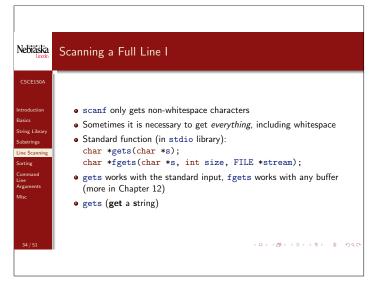


Nebraska Lincoln	Su	bstrings II
CSCE150A Introduction Basics String Library Substrings Line Scanning Sorting Command	1 2 3 4 5 6	<pre>char aString[100] = "Please Email me at the address bgriffin@cse.unl.edu, thn:" char myEmail[20]; //copy a substring strncpy(myEmail, &aString[31], 20); printf("email is %s\n",myEmail);</pre>
Line Arguments Misc 32 / 51	Re	sult: email is bgriffin@cse.unl.edu .□

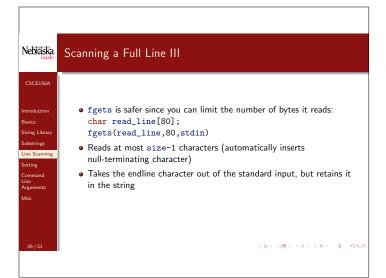
Nebraska Lincoln	Pitfalls & Strategies
CSCE150A Introduction Basics String Library Substrings Line Scanning Sorting Command Line Arguments Misc	Two most important questions when dealing with strings: Is there enough room to perform the given operation? Does the created string end in '\0'? Read the documentation (man pages) Each string function has its own expectations and guarantees
33/51	(D) (B) (E) (E) E VQ(

Notes			

		_
		_
		_
		_
		_
		_
		_



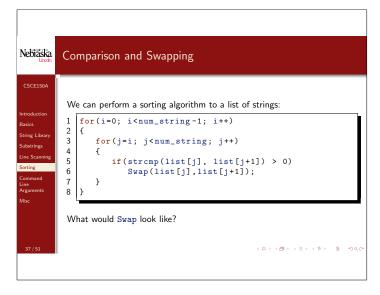
A.Y. Junetryle		
Nebřäska Lincoln	Sc	anning a Full Line II
CSCE150A		
Introduction Basics String Library	1 2 3	<pre>char read_line[80]; gets(read_line); printf("I read your line as \"%s\"\n", read_line);</pre>
Substrings Line Scanning Sorting Command Line Arguments Misc		 Dangerous: If the user enters more than 79 characters, no room for null-terminating character If user enters more than 80 characters: overflow Can actually be a security hazard Compiler message: (text+0x2c5): warning: the 'gets'function is dangerous and should not be used.
35 / 51	ı	(D)(B)(E)(E) E) 3 (O)



		-
		-
		-
		-
		-

Not				
No	es			

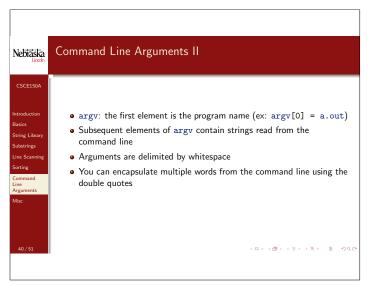
		-
		-



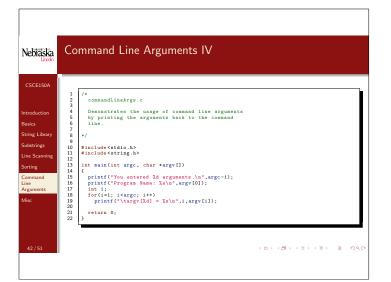
ĺ									
Nebraska Lincoln	Сс	mparison and Swapping							
CSCE150A									
Introduction Basics	S۱	vapping two strings:							
String Library Substrings Line Scanning Sorting	1 2 3	<pre>strcpy(tmp, list[j]); strcpy(list[j], list[j+1]); strcpy(list[j+1], tmp);</pre>							
Command Line Arguments Misc	Ci	areful: how big does tmp need to be?							
38 / 51			< 0	- +	· Ø	> ∢∃	ž > + 3	i > 1	∄ ≁9 ੧ (

Nebraska Lincoln	Command Line Arguments I
CSCE150A	
Introduction Basics String Library	Up to now, your int main(void) functions have not taken any parameters. To read parameters (delimited by white space) in from the command line, you can use
Substrings Line Scanning Sorting	<pre>int main(int argc, char *argv[])</pre>
Command Line Arguments	 argc gives you a count of the number of arguments which are stored in argv
Misc	argv is an array of strings (two-dimensional array of characters)
39 / 51	<ロ> (費) (を) を かえか

Notes			



Nebraska Lincoln	Command Line Arguments III
CSCE150A Introduction Basics String Library Substrings Line Scanning Sorting Command Line Arguments Misc	<pre>cse> a.out hello world abc 123 "hi everyone" would result in: argc = 6 argv[0] = a.out argv[1] = hello argv[2] = world argv[3] = abc argv[4] = 123 argv[5] = hi everyone</pre>
41 / 51	(D)(B)(E)(E) E 4)Q(b



		
-		

Not	res	

Nebiaska

Character Analysis and Conversion

CSCE150A

- The C library ctype.h provides several useful functions on characters
- isalpha(char ch) is true if ch is an alphabetic character (upper or lower case)
- isdigit(char ch) is true if ch is a character representing a digit
- islower(char ch) is true if ch is a lower-case character
- isupper(char ch) (guess)
- toupper and tolower convert alphabetic characters (no effect otherwise)
- ispunct(char ch)
- isspace(char ch) true if ch is any whitespace character
- stdio.h has getchar(void) and getc(FILE *inp), which read in one character at a time (use to build scanline in Fig 9.15)

< 마 > (# > (분 > (분 > 분) 원)



Nebraska	

String-to-Number and Number-to-String Conversions I

CSCE150

Introduction
Basics
String Library
Substrings
Line Scanning
Sorting
Command

Command Line

 stdlib.h provides several functions for converting between strings and numbers

• String to numbers: int atoi(const char *nptr); double atof(const char *nptr);

- \bullet Returns the value of the number represented in the string ${\tt nptr}$
- a (alpha-numeric) to integer, floating point
- Does not handle errors well: returns zero if it fails (see strtol for advanced behavior)

□ > +@ > + ≥ > + ≥ > - ≥ - • 0 <

Nebraska

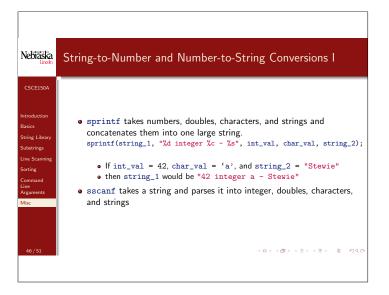
String-to-Number and Number-to-String Conversions II

Introduction
Basics
String Library
Substrings
Line Scanning
Sorting
Command
Line
Arguments

1	#include < stdlib.h>
2	#include <stdio.h></stdio.h>
3	
4	<pre>int main(int argc, char *argv[])</pre>
5	{
6	if(argc != 3)
7	{
8	<pre>printf("Usage: %s integer double\n", argv[0]);</pre>
9	exit(-1);
LO	}
11	<pre>int a = atoi(argv[1]);</pre>
12	<pre>double b = atof(argv[2]);</pre>
13	<pre>printf("You gave a = %d, b = %f ",a,b);</pre>
4	<pre>printf("as command line args\n");</pre>
15	return 0;
16	}

Notes	

		_
		_
		_
		_
		_
		_



```
Notes
```

```
Nebraska
Introduction
Basics
String Library
Substrings
Line Scanning
Sorting
Command
Line
Command
Line
Line
Arguments
Misc

String Library
Substrings
Line
Line
Arguments
Misc

String Library
Substrings
Line
To Sorting
Command
Line
Arguments
Misc

String Library
Substrings
Fig. 1

Int num;
double pi;
double pi;
scanf ("42 3.141592 Stewie Griffin", "%d %lf %s %s", &num,
&pi,
a,
b);
printf ("num = %d\n", num);
printf ("num = %d\n", num);
printf ("a = %s\n", a);
Introduction
Substrings
Applied
To Substrings
Introduction
Int
```

Notes		
		_
		_
		_
		_
		_
		_
		_

```
String-to-Number and Number-to-String Conversions III

CSCE190A

Introduction
Basics
String Library
Subtrings
Subtrings
Sorting
Command
Line Scanning
Sorting
Command
Line
Arguments
Misc

48/51
```

Notes		
		_
		_
		_
		_
 		_
		_

Nebraska Common Programming Errors I • We usually use functions to compute some value and use the return to send that value back to the main function. However, functions are not allowed to return strings, so we must use what we learned about input/output parameters • Know when to use & and when not to Use them for simple data types: int, char, and double Do not use them for whole arrays (strings)

Nebiaska Lincoln	Common Programming Errors II
CSCE150A	
Introduction	Be careful not to overflow strings
Basics	Always follow expected formats
String Library	Read the documentation!
Substrings Line Scanning Sorting	 Most important: make sure all strings are null-terminated (a '\0' at the end)
Command Line Arguments Misc	 Just because your program seems to work, doesn't mean it always does (ex: add & to a, b in the sscanf snippet above)
50/51	(ロ) (西・(さ) (き) を か)

Nebraska Lincoln	Exercises I
CSCE150A Introduction Basics String Library Substrings Line Scanning Sorting Command Line Arguments Misc	 Write a program that takes command line arguments and prints them out one by one. Then sort them in lexicographic order and print them out again. A palindrome is a string that is the same backwards and forwards (example: tenet, level). Write a program that reads a string from the command line and determines if it is a palindrome or not. In the case that it is not, make the string a palindrome by concatenating a reversed copy to the end.
51/51	(D) (Ø) (E) (E) E NQ(P

Not	ec	
NOL	es	
Niet		
Not	es	
-		
-		
Not	es	
-		

Not	es	