Princeton University
COS 217: Introduction to Programming Systems
Manipulating C Strings

String Operation	String in Stack	String in Rodata Section

Allocating memory for a string	`{`
	`  char acStr[5];`
	`  ...`
	`}`
		`{`
		`  ...`
		`  ..."hi"...`
		`  ...`
		`}`

Initializing a string	`{`
	`  char acStr1[3] = {'h', 'i', '\0'};`
	`  char acStr2[] = {'h', 'i', '\0'};`
	`  char acStr3[3] = "hi";`
	`  char acStr4[] = "hi";`
	`  char acStr5[2] = "hi"; /* truncation */`
	`  char acStr6[10] = "hi";`
	`  ...`
	`}`
		`{`
		`  ...`
		`  ..."hi"...`
		`  ...`
		`}`

Computing the length of a string	`{`
	`  char acStr[10] = "hello";`
	
	`  ... strlen(acStr) ...`
	`  /* Evaluates to 5 */`
	
	`  ... sizeof(acStr) ...`
	`  /* Evaluates to 10 */`
	`}`
		`{`
		`  char *pcStr = "hello";`
		
		`  ... strlen(pcStr) ...`
		`  /* Evaluates to 5 */`
		
		`  ... sizeof(pcStr) ...`
		`  /* Evaluates to 4 */`
		`}`

Changing the characters of a string	`{`
	`  char acStr[10] = "hi";`
	
	`  acStr = "bye"; /* compiletime error */`
	
	`  acStr[0] = 'b';`
	`  acStr[1] = 'y';`
	`  acStr[2] = 'e';`
	`  acStr[3] = '\0';`
	
	`  strcpy(acStr, "bye");`
	`  /* Danger of memory corruption. */`
	`}`
		`(Runtime error to attempt to change the characters of a string that resides in the rodata section)`

Concatenating characters onto a string	`{`
	`  char acStr[10] = "hi";`
	
	`  acStr += "bye"; /* compiletime error */`
	
	`  acStr[2] = 'b';`
	`  acStr[3] = 'y';`
	`  acStr[4] = 'e';`
	`  acStr[5] = '\0';`
	
	`  strcat(acStr, "bye");`
	`  /* Danger of memory corruption. */`
	`}`
		`(Runtime error to attempt to change the characters of a string that resides in the rodata section)`

Comparing one string with another	`{`
	`  char acStr1[] = "hi";`
	`  char acStr2[] = "bye";`
	
	`  if (acStr1 < acStr2) ...`
	`  /* Legal, but compares addresses!!! */`
	
	`  if (strcmp(acStr1, acStr2) < 0) ...`
	`  /* Compares strings */`
	`}`
		(Same as string in stack)

Reading a string	`{`
	`  char acStr[10];`
	
	`  iConvCount = scanf("%s", acStr);`
	`  /* Reads a word as a string.`
	`  **Grave** danger of memory corruption. */`
	
	`  iRet = gets(acStr);`
	`  /* Reads a line as a string,`
	`  removing the \n character.`
	`  **Grave** danger of memory corruption. */`
	
	`  iRet = fgets(acStr, 10, stdin);`
	`  /* Reads a line as a string,`
	`  retaining the \n character. */`
	`}`
		(Runtime error to attempt to change the characters of a string that resides in the rodata section)

Writing a string	`{`
	`  char acStr[] = "hi";`
	
	`  iCharCount = printf("%s", acStr);`
	`  /* Writes a string. */`
	
	`  iSuccessful = puts(acStr);`
	`  /* Writes a string, appending a \n`
	`  character. */`
	
	`  iSuccessful = fputs(acStr, stdout);`
	`  /* Writes a string. */`
	`}`
		(Same as string in stack)

Converting a string to another type	`{`
	`  char acStr[] = "123";`
	`  int i;`
	`  long l;`
	`  double d;`
	`  iConvCount = sscanf(acStr, "%d", &i);`
	`  i = atoi(acStr);`
	`  l = atol(acStr);`
	`  d = atof(acStr);`
	`}`
		(Same as string in stack)

Converting another type to a string	`{`
	`  char acStr[10];`
	`  int i = 123;`
	
	`  iCharCount = sprintf(acStr, "%d", i);`
	`  /* Danger of memory corruption. */`
	`}`
		(Runtime error to attempt to change the characters of a string that resides in the rodata section)