

ZyBook

- Sections

15.14, 15.15, 15.20, 15.24,

18.10, 18.11, 18.12. 18.13

21.5, 21.6

22 (Input/Output)

C Strings

- Remember C strings are character arrays
- Terminated by NULL '\0' 0 (zero)
- char szName[6];
- sz means NULL terminated string

C	S	3	0	0	'\0'
---	---	---	---	---	------

Input/Output

```
#include <string.h>
```

```
#define SIZE 100
```

```
char szName[SIZE];
```

```
scanf("%s", szName);
```

```
printf("You typed: %s\n", szName);
```

```
printf("You typed %d characters\n",  
    strlen(szName));
```

Zero out a String

```
#include <string.h>
```

```
memset(szName, '\0', 6)
```

'\0'	'\0'	'\0'	'\0'	'\0'	'\0'
------	------	------	------	------	------

C String functions

```
#include <string.h>
```

```
int strcmp(const char *s1, const char *s2);
```

return 0 if the strings are equal, >0 or <0
otherwise

```
char* strncpy(char *dest, const char *src, size_t n);
```

Copy at most n characters from src to dest

Parsing in C

- Functions available

- `char *strtok_r(char *str, const char *delim, char **saveptr);`
 - tokenize a string on any of a number of delimiters
 - **alters the string!**
- `char *strstr(const char *haystack, const char *needle);`
 - find a substring, return pointer to substring or NULL
- `char *strchr(const char *s, int c);`
 - find the first occurrence of a character, return pointer to c or NULL
- `char *strrchr(const char *s, int c);`
 - find the last occurrence of a character, return pointer to c or NULL
- `int isdigit(int c)`
 - is the char a digit? return 0 if false
 - many *istype()* functions exist

strtok_r

- strtok_r is great for *tokenizing* a string start to finish with a variable set of delimiters
 - **ALTERS THE STRING**
- `pToken = strtok_r(pBuffer, " :", &pSavePtr);`
 - find the first token delimited by space or :
- `pToken = strtok_r(NULL, " :", &pSavePtr);`
 - find the second token delimited by space or :

Notes

- `#include <string.h>`
- `#include <ctype.h>`