C-Strings

Definition
- An array of characters
- Where the used portion is terminated by a null character
- `<string.h>`
- Library that acts on C-strings
- Most will crash if given something that does not fit the definition above

Creating and Initializing a string
- `char name1[4] = {'J','a','n','\0'};`
- `char name2[6] = "Plane";`

Characters, strings and numeric values are all different
- `strlen` uses the definition of C-string to find number of used characters
- `sizeof` operator tells the size of the variable or type
- `strlen` uses the definition of C-string to find number of used characters

Input and Output

Output
- `%s` in printf format string
- `puts()` function takes a string as the only argument

Input
- Dangerous to use `%s` in scanf or to use `gets()` function
- `char *fgets(char *buffer, int bufferSize, FILE *stream);`
  - read a line into buffer (at most bufferSize-1 characters)
  - null byte added at end of buffer
  - reads from stream – for standard input just type `stdin` as the name of the stream
  - returns `NULL` on error or end of file
  - on success returns pointer to the space where you read into (here called the buffer)
Strings

- Zero or more characters followed by null char \('0'\)
  - Also called NUL
  - Not counted as part of string
- String.h defines prototypes for string routines

Copying Strings

- `size_t  strlen(char const *str);`
  - Returns count of characters in str
- `char *strncpy(char *dst, char const *src, size_t len);`
  - Copy src to dst
  - Copy until '0' in src or at most len characters
  - Pad extra characters with '0'
  - Safety tip: `dst[len-1] = '0';` to force termination of new string
- `char *strncat(char *dst, char const *src, size_t len);`
  - Append src onto the end of dst
  - Always append NUL to end of dst string

String Functions

Comparison

- `int strncmp(char const *s1, char const *s2, size_t len);`
  - Returns 0 if string equal up to len
  - Returns a negative value if s1 is less than s2
  - Returns a positive value if s1 is greater than s2

Searching

- `char *strchr(char const *str, int ch);`
- `char *strrchr(char const *str, int ch);`
  - Finds the first occurrence of ch in str
  - strrchr finds the last occurrence
  - Returns NULL if not found
- `char *strstr(char const *s1, char const *s2);`
  - Find the first occurrence of s2 in s1

String Functions Examples

```
char string[] = "this is a test string";
char *ans;
int length;

length = strlen(string); /* returns 21 */
ans = strchr(string, 'h'); /* returns string + 1 */
ans = strrchr(string, 't'); /* returns NULL */
ans = strstr(string, "test"); /* returns string + 10 */
```
Character Functions
Prototypes in ctype.h

- Classifying characters
  - Parameter is int, but it's a character
  - int isspace(int ch);
    - Returns true if ch is ' ', '
', '	', form feed, or carriage return
  - int isdigit(int ch);
    - Returns true if ch is 0 through 9
  - int islower(int ch) and isupper(int ch);
    - Returns true if it's a-z for islower and A-Z and isupper
  - int isalpha(int ch);
    - Returns true if it's a-z or A-Z
  - int isalnum(int ch);
    - Returns true if it's a-z or A-Z or 0-9

- Transformation
  - int toupper(int ch), int tolower(int ch)
    - Converts to upper/lower case

typedef

- Allows you to define a new type
- Format:
  typedef whatItIs whatYouWantToCallIt;

- For example:
  typedef int Bool;

- Array example:
  typedef char MyString[MAX];