Character Strings

COEN 10

C -- Lecture 5

Character Strings

★ Series of one or more characters

Example: "This is a string!"

◎ The double quotation marks enclose the string

Character Strings

★ Type char and the null character

◎ C has no type for strings

◎ Strings are stored in an array of type char

◎ The end of the string is marked with a null character, represented by '\0'

Character Strings

★ Type char and the null character

◎ Since every string has a null character at the end, the array needs an extra position

Character Strings

★Using a character string

- ◎Declare an array with one extra position
- ◎Place the characters in the array one by one
- ◎Add the ‘\0’ at the end

Using Strings

★Declare

```
char name[20] = “Mary”;
char otherName[20];
char oneMoreName[] = “Joe”;
```

Using Strings

★printf

- ◎The placeholder for strings is %s

Example

```
printf (“My name is %s\n”, name);
```

Using Strings

★scanf

- ◎The placeholder for strings is %s
- ◎Reads a single word per placeholder
- ◎Places the null at the end

Example

```
scanf (“%s”, otherName);
```

Using Strings

- ★Strings versus Characters
  - ◎'x' is a character
    - ✧1 byte
  - ◎"x" is a string
    - ✧2 bytes

Using Strings

- ★Searching for a character in a string

```
char string[SIZE];
…
scanf ("%c", &c);
for (i = 0; string[i] != '\0'; i++)
  if (c == string[i])
    printf ("found!\n");
…
```

Using Strings

- ★Searching for a character in a string

```
char string[SIZE];
…
scanf ("%c", &c);
i = 0;
while (string[i] != '\0')
{
  if (c == string[i])
    printf ("found!\n");
  i++;
}
```

Using Strings

- ★String functions
  - ◎C provides several functions for string manipulation
  - ◎Require string.h

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

Using Strings

- ★The strlen() function
  - ◎Returns the number of characters in the string
    - ✧not including the null character
  - ◎Different than sizeof()
    - ✧sizeof returns the number of bytes in the array

Using Strings

- ★The strcmp() function
  - ◎Compares two strings
    - ✧The return is
      - – zero: if the are equal
      - – negative: if the first one is smaller alphabetically then the second
      - – positive: if the first one is greater alphabetically then the second

Arrays of strings

- ★Represented by a 2D array of characters

  Example

  char names[10][20];

  - ◎Creates an array with space for 10 strings, each of size at most 19.

Arrays of strings

- ★Searching for a string in an array of strings

```
…
scanf (“%s”, searchName);
for (i = 0; i < 10; i++)
   if (strcmp (names[i], searchName) == 0)
      printf (“found!\n”);
…
```

Constants and the C Preprocessor

- ★ Avoid literals in the programs
  - ◎ Use constants instead
    - ✧ Preprocessor substitution

      `#define <NAME> <value>`
    - ✧ The const Modifier

      `const <type> <NAME> = <value>;`
  - ◎ Use capital letters for constants