16.216: ECE Application Programming

Fall 2013

Lecture 21: Key Questions
October 25, 2013

1. (Review) Describe how character arrays can be used to represent strings in C.

2. Describe the C functions used for:
a. Copying one string to another

b. Comparing two strings

c. Checking the number of characters in a string

d. Concatenating two strings together

3. **Example:** What does the following program print?

```
int main() {
      char s1[15];
      int n1;
      char s2[10] = ".216";
      int n;

      strncpy(s1, "16", 15);
      n1 = strlen(s1);
      printf("s1 = %s\n", s1);
      printf("Length of s1 = %d\n\n", n1);

      printf("%c\n\n", s1[1]);

      strncat(s1,s2,10);
      n1 = strlen(s1);
      printf("s1 = %s\n", s1);
      printf("Length of s1 = %d\n\n", n1);

      // Assume user inputs: ABC ABD
      printf("Enter two strings:");
      scanf("%s%s", s1, s2);
      n = strncmp(s1, s2, 15);
      if (n > 0)
            printf("%s > %s\n", s1, s2);
      else if (n < 0)
            printf("%s < %s\n", s1, s2);
      else
            printf("%s == %s\n", s1, s2);
      return 0;
}
```

4. **Example**: Write a function for each of the following:

a. `int readStrings(char *s);`

Repeatedly read strings from standard input until the input string matches `s`. Return the number of strings read.

b. `void copyNull(char *s1, char *s2, int n);`

Copy the first *n* characters of `s2` into `s1`, and make sure that the new version of `s1` terminates with a null character.

c. `int fillString(char *s);`

Repeatedly read strings from standard input and concatenate them to `s` until there is no room in the string. Return the final length of the string.

For example, if `s` is a 6-character array already holding “abcd”:

- User enters “e”—string is full; return 5
- User enters “ef”—there’s not enough room; return 4