Programming for Engineers I
Lab 09
Strings

Hina Ashraf Attique Dawood

November 16, 2012

1 What are Strings?

Simple answer: character arrays are called strings. A bunch of characters most often only make sense if they form words. And words only make sense if they are part of a sentence. Essentially, character arrays are used for storage of words, sentences or the like.

2 Initialization at Declaration

Character arrays have an additional way of initialisation, in addition to other array initialisation methods. Without specifying size, a char array is assigned a string at declaration. In this case, compiler automatically allocates space for array. To print the contents of an array using `printf`, array name is passed as argument and `%s` is used for 'string'.

```
char myName[] = "abc";
printf("%s\n", myName);
Output:
abc
```

3 The `NULL` Character

You might ask the question, when using `printf` how does the compiler know when the string ends? The answer is that every string ends (or must end) with a `NULL` character (or `'\0'` in ASCII). In above example, `myName` was allocated 4 bytes by compiler to end the string with a `'\0'` character. Built-in string functions like this one automatically append `'\0'` at the end of a string. In case you are working or manipulating character arrays directly, then you must take care of `'\0'` yourself.

4 Length of String

Length of a string does not include `'\0'`. It only counts the total number of characters before `'\0'`. The **size of array** containing string and **length of string** are two different things. In the code segment below, size of array is 10 and length of stored string is 3.

```
char myName[10];
myName[0] = 'a';
myName[1] = 'b';
myName[2] = 'c';
myName[3] = '\0'; // Terminating NULL character.
printf("%s\n", myName);
```

```
Output:
abc
```

5 String Input

Strings can be input using either `scanf()` or `gets()` functions. `scanf` cannot input spaces so `gets` should be used if input string contains spaces.

Care should be taken when entering strings. The total number of characters must not exceed the size of array. In addition maximum number of characters must be 1 less than array size for a terminating `‘\0’`.

```
char CourseName1[10];
char CourseName2[10];
char CourseName3[10];

gets(CourseName1);
gets(CourseName2);
gets(CourseName3);

Output:
Physics                         // Correct input, string length=7.
Programming for Engineers I     // WRONG! Input exceeds size of array.
Calculus I                      // WRONG! No space for ‘\0’, string length=10.
```

CourseName1:	P	h	y	s	i	c	s	NULL	X	X	✓
CourseName2:	P	r	o	g	r	a	m	m	i	n	✗
CourseName3:	C	a	l	c	u	l	u	s		I	✗

The solution in this case would be to increase the size of arrays for course 2 and course 3 so they can hold the input strings.

6 Exercise

Question No. 1: Take a string of arbitrary size. Determine the length of this string.
Question No. 2: Take two arrays. Input a random string in one array. Copy this string into second array.
Question No. 3: Take an input string. Reverse this string storing the resulting string in the same array. Only reverse characters.
Question No. 4: Take two strings. Check if the strings match.
Question No. 5: Concatenate two strings to form a third string. For third string, array should be large enough to hold the resultant string.