

# Tutorial 7: Strings

APS 105: Computer Fundamentals  
Winter2020

This week we cover strings, as they are very important in all modern programming languages. In the following examples, we first focus on how strings are processed with the help of character by character operations. Then, we will try some useful functions to tackle these things.

## 1 Character by character operations(a)

In this part, we will learn how to initialize a string in different ways and change the specific parameter within the given string.

```
1 void Q1()  
2 {  
3     printf("Q1:\n");  
4     //Approach 1  
5     const int LENGTH = 5;  
6     //Declaring and initializing a string as a character array  
7     char str[LENGTH + 1];  
8     str[0] = 'H';  
9     str[1] = 'e';  
10    str[2] = 'l';  
11    str[3] = 'l';  
12    str[4] = 'o';  
13    str[5] = '\0';  
14  
15    printf("Approach 1: str = %s\n", str);  
16  
17    //Approach 2  
18    char s[] = "Hello";  
19    // The length of the string could be ignored.  
20    // However, if you declare a string without enough space, for example  
21    "char s[4] = "Hello";"  
22    // A compile-time warning and run-time error would occur  
23  
24    printf("Approach 2: s = %s\n", s);  
25  
26    //Approach 3  
27    char *p = "Hello";  
28    // Declaring and initializing a string as a character pointer  
29    printf("Approach 3: p = %s\n", p);
```

```

30
31 //Now try to change the characters in the string s
32 printf("\n");
33 s[1] = 'E';
34 printf("s = %s\n", s);
35 // But you cannot change s to point to another string.
36 // For example, "s = "HELLO;" is wrong!!!
37 // A compile-time error, which is "array type is not assignable", will
  occur
38
39 // You can switch the pointer variable p to another string
40 p = "HELLO";
41 printf("p = %s\n", p);
42 // You cannot change characters in the string pointed to by p
43 // For example, "p[1] = 'E';" is wrong!
44 // It may cause a segmentation fault or bus error (on Mac OS X)
45
46
47 // Compare the pointer values to see where these strings are
48 printf("\nThe pointer variable of s: %p\n", s);
49 printf("The pointer variable of p: %p\n\n", p);
50
51 // Now let p point to s, so that we can safely change the characters
52 p = s;
53 printf("p = %s\n", p);
54
55 p[2] = 'L';
56 printf("p = %s\n", p);
57
58 }

```

Listing 1: Solution 1

The output is shown in fig. 1.

```

Q1:
Approach 1: str = Hello
Approach 2: s = Hello
Approach 3: p = Hello

s = Hello
p = HELLO

The pointer variable of s: 0x7ffeee626810
The pointer variable of p: 0x1015d9d49

p = Hello
p = HELLO
Hit any key to continue...

```

Figure 1: Output 1

## 2 Character by character operations(b)

In this part, we will try to compute the number of words in the given sentence by applying character to character operations we learnt above.

```
1 void Q2()
2 {
3     printf("Q2:\n");
4     char s[] = "This is a given sentence.";
5     int numberOfSpaces = 0;
6
7     for(int i = 0; s[i] != '\0'; i++)
8     {
9         if (s[i] == ' ')
10             numberOfSpaces++;
11     }
12
13     printf("The sentence is:\n");
14     puts(s); // We can use puts() to print a string
15     printf("The number of words in the given sentence is %d\n",
16           numberOfSpaces + 1);
17 }
```

Listing 2: Solution 2

The output is shown in fig.2.

```
Q2:
The sentence is:
This is a given sentence.
The number of words in the given sentence is 5
Hit any key to continue...
```

Figure 2: Output 2

## 3 Basic string functions(a)

Currently, many useful functions were predefined for you, meaning that you could simply use those functions in your own program, without worrying about how the strings are processed within the functions. So, please just try to use them. In this problem, we would look for the index of the specific character in the given string.

(Hints: The *strchr()* function finds the first occurrence of a character in a string. The character *c* can be the null character (`\0`); the ending null character of string is included in the search.)

```
1 void Q3()
2 {
3     // strchr: Looking for a specific character in a given string
4     printf("\nQ3:\n");
5     char s[] = "This is a given string";
6     char a;
```

```

7   printf("What are you looking for?\n");
8   scanf("%c",&a);
9   printf("Looking for the character '%c' in \"%s\"...\n", a, s);
10
11  char *p = strchr(s, 's');
12
13  while (p != NULL)
14  {
15      printf("Found at %ld\n",p - s + 1);
16      p = strchr(p + 1, 's');
17  }
18 }

```

Listing 3: Solution 3

The output is shown in fig. 3.

```

Q3:
What are you looking for?
e
Looking for the character 'e' in "This is a given string"...
Found at 4
Found at 7
Found at 17
Hit any key to continue...
_

```

Figure 3: Output 3

## 4 Basic string function(b)

Let's explore more!

```

1 void Q4()
2 {
3     printf("Q4:\n");
4     char s1[99];
5     char s2[99];
6
7     printf("\nPlease enter a string:\n");
8     gets(s1); //Instead of "scanf"
9     printf("The string you entered is:\n");
10    puts(s1); //Instead of "printf"
11
12    printf("Length of the string is: %lu\n", strlen(s1));
13    //strlen(string) could compute the length of the string
14
15    printf("\nPlease enter another string:\n");
16    gets(s2);
17
18    if (strcmp(s1, s2) ==0) // strcmp(string1, string2) could compare these
19        {

```

```

20     printf("str1 and str2 are equal.\n");
21 }else
22 {
23     printf("str1 and str2 are different.\n");
24 }
25
26 printf("\nLet's concatenate these two strings:\n");
27 strcat(s1,s2); // Concatenate string1 and string2.
28 printf("Output string after concatenation: %s\n", s1);
29
30 printf("\nLet's copy the string str1 into string str2:\n");
31 strcpy(s2,s1); //copy string1 into string2,including the end character
32               '\0'
33 printf("Str2 is: %s\n", s2);
34 }

```

Listing 4: Solution 4

The output is shown in fig. 4

Q4:

```

Please enter a string:
warning: this program uses gets(), which is unsafe.
Hello World!
The string you entered is:
Hello World!
Length of the string is: 12

Please enter another string:
Happy Coding!
str1 and str2 are different.

Let's concatenate these two strings:
Output string after concatenation: Hello World! Happy Coding!

Let's copy the string str1 into string str2:
Str2 is: Hello World! Happy Coding!
Hit any key to continue...

```

Figure 4: Output 4