# Manipulating C Strings

<table>
<thead>
<tr>
<th>String Operation</th>
<th>String in Stack</th>
<th>String in Rodata Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocating memory for a string</td>
<td><code>{ char ac[5]; ... }</code></td>
<td><code>{ ... &quot;hi&quot;... }</code></td>
</tr>
<tr>
<td>Initializing a string</td>
<td>`{ char acA[3] = {'h', 'i', '&quot;\0&quot;}; char acB[] = {'h', 'i', '&quot;\0&quot;}; char acC[10] = {'h', 'i', '&quot;\0&quot;}; char acD[3] = &quot;hi&quot;; char acE[] = &quot;hi&quot;; char acF[10] = &quot;hi&quot;; ... }</td>
<td><code>{ ... &quot;hi&quot;... }</code></td>
</tr>
<tr>
<td>Computing the length of a string</td>
<td>`{ char ac[20] = &quot;hello, world&quot;; /* Evaluates to 12 <em>/ ... strlen(ac) ... /</em> Evaluates to 20 */ ... sizeof(ac) ... }</td>
<td>`{ char <em>pc = &quot;hello, world&quot;; /</em> Evaluates to 12 <em>/ ... strlen(pc) ... /</em> Evaluates to 8 */ ... sizeof(pc) ... }</td>
</tr>
<tr>
<td>Changing the characters of a string</td>
<td>`{ char ac[10] = &quot;hi&quot;; /* Compile-time error. <em>/ ac = &quot;bye&quot;; /</em> The long way. <em>/ ac[0] = 'b'; ac[1] = 'y'; ac[2] = 'e'; ac[3] = '&quot;\0&quot;; /</em> The shortcut. <em>/ strcpy(ac, &quot;bye&quot;); /</em> Dangerous. */ }</td>
<td><code>{Runtime error to attempt to change the characters of a string that resides in the rodata section}</code></td>
</tr>
</tbody>
</table>
### Comparing one string with another

```c
{  char acA[] = "hi";
    char acB[] = "bye";

    /* Legal, but compares addresses!!! */
    if (acA < acB) ...
    /* Compares strings */
    if (strcmp(acA, acB) < 0) ...
}
```

(Same as string in stack)

### Reading a string

```c
{  char ac[10];

    /* Reads a word as a string. */
    iConvCount = scanf("%s", ac);
    /* Dangerous. */

    /* Reads a line as a string, removing the \n character. */
    iRet = gets(ac);
    /* Dangerous. */

    /* Reads a line as a string, retaining the \n character. */
    iRet = fgets(ac, 10, stdin);
}
```

(Runtime error to attempt to change the characters of a string that resides in the rodata section)

### Writing a string

```c
{  char ac[] = "hi";

    /* Writes a string. */
    iCharCount = printf("%s", acStr);

    /* Writes a string, appending a \n character. */
    iSuccessful = puts(ac);

    /* Writes a string. */
    iSuccessful = fputs(ac, stdout);
}
```

(Same as string in stack)

### Converting a string to another type

```c
{  char ac[] = "123";
    int i;
    long l;
    double d;

    iConvCount = sscanf(ac, "%d", &i);
    i = atoi(ac);
    l = atol(ac);
    d = atof(ac);
}
```

(Same as string in stack)

### Converting another type to a string

```c
{  char ac[10];
    int i = 123;

    iCharCount = sprintf(ac, "%d", i);
    /* Dangerous. */
}
```

(Runtime error to attempt to change the characters of a string that resides in the rodata section)