

## Re: Another Understanding Pointers Question

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2004-05/1036.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2004-05/1036.html)

---

**From:** Materialised (*Materialised\_at\_privacy.net*)

**Date:** 05/11/04

Date: Tue, 11 May 2004 01:56:17 +0100

Materialised wrote:

```
> Hi everyone,
> I seen the post by Rob Morris, and thought that I would double check
> that I was using pointers in the correct way. So I written the following
> string functions to test. I know soem can be iumplimented using the
> standard library, but I just wanted to test writing my own functions.
> They work ok, but I would like some feed back on any issues you can see
> with them etc
>
> #include <stdio.h>
> #include <stdlib.h>
> #include <string.h>
>
> char *left(char *string, int count)
> {
> char *p;
> int i;
> p = malloc((count * sizeof(char)+1));
> if(!p){
> printf("Cannot allocate memory\n");
> exit(1);
> }
>
>
> for(i = 0; i <= count -1; i++) {
> p[i] = string[i];
> }
> p[i++] = '\0';
>
> return(p);
>
> }
>
> char *right(char *string, int count)
> {
> char *p;
> int len, i, j = 0;
> p = malloc((count * sizeof(char)+1));
```

comp.lang.c: Re: Another Understanding Pointers Question

```
> if(!p){
> printf("Cannot allocate memory\n");
> exit(1);
> }
> len = strlen(string);
> for(i = (len - count); i <= len; i++){
> p[j] = string[i];
> j++;
> }
>
> p[j++] = '\0';
> return(p);
>
> }
>
> char *chreplace(char *string, int count, char rep)
> {
> char *p;
> p = malloc((sizeof(string)+1));
> if(!p){
> printf("Cannot allocate memory\n");
> exit(1);
> }
> count--;
> strcpy(p, string);
> p[count] = rep;
>
> return(p);
> }
>
> char *section(char *string, int from, int to)
> {
> char *p;
> int i, j = 0;
>
> p = malloc(((to - from) * sizeof(char)+1));
> if(!p){
> printf("Cannot allocate memory\n");
> exit(1);
> }
> for( i = from; i <= to; i++) {
> p[j] = string[i];
> j++;
> }
> p[j++] = '\0';
>
> return(p);
> }
> int main(void)
> {
> char blah[] = "abcdefghijklm";
```

## comp.lang.c: Re: Another Understanding Pointers Question

```
> char *test;
> char *test2;
> char *test3;
> char *test4;
>
> test = left(blah, 10);
> test2 = right(blah, 10);
> test3 = chreplace(blah, 2, 'Q');
> test4 = section(blah, 4, 10);
> puts(test);
> puts(test2);
> puts(test3);
> puts(test4);
> return 0;
> }
>
> Comments and improvements are welcome, flames to if appropriate.
```

Just a question,

I have taken all the people who replied comments to heart, and I am glad for your feedback.

What I dont understand is what is wrong with sizeof(char)? True char may be defined as 1 on most architectures. But does the way I have referenced it require extra CPU useage or something?

Sorry if thats a dumb question..

--

-----

Materialised

```
perl -e 'printf "%silto%c%sck%ccodegurus%corg%c", "ma", 58, "mi", 64, 46, 10;'
```