

# Re: exercise 1–20 K&R

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2006-06/msg01282.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-06/msg01282.html)

---

- *From:* Michael Mair <[Michael.Mair@xxxxxxxxxxxxxxxxxxx](mailto:Michael.Mair@xxxxxxxxxxxxxxxxxxx)>
  - *Date:* Sun, 11 Jun 2006 00:40:19 +0200
- 

Aenima1891 schrieb:

write a program that replaces strings of blanks by the minimum number of blanks to space to the next tab stop.

```
#include <stdio.h>
#define TABSIZE 8
```

```
int main()
```

```
int main (void)
```

```
{
  int counter, c;
```

Note: It often helps understanding to separate variable declarations by "purpose" and not only by type. counter is used just as that, whereas c is intended to hold a character[\*].

This also makes for easier maintenance if you find out that the `_guaranteed_` range of int is not sufficient for your purposes and you have to switch to long, unsigned long or size\_t as counter type.

```
int counter, anotherCounter;
int c, anotherTempForCharacters;
int resultOfSomeNoncounterCalculation;
```

Some company coding standards even forbid declaring more than one variable per line if the declared variables are not loop variables.

```
while ((c = getchar()) != EOF) {
  if (c == '\t') {
    for (counter = 1; counter <= TABSIZE; ++counter)
      putchar(' ');
```

You misunderstood the purpose of the task.

Every `_line_` is to be formatted independently of the other. For every line, you start anew with tab stops. This means that you need a character count for the current line, not an overall character count.

Tab stops are at the same character number (modulo `TABSIZE`) throughout the line, i.e. you insert space characters until your counter modulo `TABSIZE` equals that number.

Example: For `#define TABSIZE 4`,  
"`\ta\tab\tabc\tabcd\tabcde\t\n`" has to become either  
"`a ab abc abcd abcde \n`"  
`^ ^ ^ ^ ^ ^ ^ ^`

where the second line indicates the tab stop positions for clarity.

Or, if you admit the possibility of inserting 0 spaces to fit the description, your result would be

"`a ab abc abcdabcde \n`"  
`^ ^ ^ ^ ^ ^ ^`

This is merely the difference between a `while()` and a `do—while()` loop (figure out which one for which case yourself) for you.

```
    }  
    else  
    putchar(c);  
    }  
    return 0;  
    }
```

what do you think about this??

[\*] To be more precise: A character cast to unsigned char or EOF.

This is what you typically pass to `putchar()`, `[f]putc()` and the `is...()` functions and get from `getchar()`, etc.

Cheers  
Michael

—

E-Mail: Mine is an `/at/ gmx /dot/ de` address.

.