

Re: how can we check to not enter the any string or char?

## Re: how can we check to not enter the any string or char?

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2007-11/msg00977.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-11/msg00977.html)

---

- *From:* "Charlie Gordon" <[news@xxxxxxxxxxxx](mailto:news@xxxxxxxxxxxx)>
  - *Date:* Tue, 6 Nov 2007 19:42:11 +0100
- 

"CBFalconer" <[cbfalconer@xxxxxxxxxx](mailto:cbfalconer@xxxxxxxxxx)> a écrit dans le message de news: 472A65A3.5791AAEC@xxxxxxxxxxxx

Flash Gordon wrote:

CBFalconer wrote, On 01/11/07 16:12:

... snip ...

```
if (((UINT_MAX - ch) / 10) > value) {
```

This is meant to be a bounded input, pass in and test against the bounds allowed rather than some other value.

No, this is intended to ensure the value fits into an unsigned int.

`Intent' is correct word: for a working implementation, the OP will have to add more work.

The blatant mistake in this comparison is not the only one.

```
/* overflow detected, decide what to do */  
ch = ch + '0'; /* restore char value */  
break;  
}  
value = 10 * value + ch;  
}  
ungetc(ch, f); /* keep exit char for the user */  
return err;  
} /* untested - you check it out */
```

Re: how can we check to not enter the any string or char?

Re: how can we check to not enter the any string or char?

Still untested by me.

... snip ...

Of course, you have just done most of this chaps homework.

No I haven't. The function is incomplete. Nobody seems to notice that it doesn't bother to return value, as it should, and that err should be a further parameter (\*err). What I have done is encourage him to develop a standard reusable input mechanism, whose value can be checked by the caller, and which doesn't require (possibly) long strings etc.

But your code is obscure and incorrect, makes use of cumbersome constructs such as comparing EOF to an unsigned int... There must be a more effective way to educate the OP.

—  
Chrlie.

.