

Re: c interview

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-08/msg00001.html

- *From:* gkumar007@xxxxxxxxxx
 - *Date:* 3 Aug 2006 01:03:14 -0700
-

Hi Flash,
First of all, thanks a lot for the detailed comments.

You use printf without a function prototype in scope. This invokes undefined behaviour because printf is a variadic function. ALWAYS include stdio.h before using printf and other headers as appropriate before using other functions.

For most of the programs, I have included it. But as you pointed out I missed out in a few of them. Thanks for pointing it.

In your first CountBits function you assume that int is 32 bits. int could be as small as 16 bits. If you want a number of at least 32 bits use long. In addition it is not guaranteed to work properly with negative numbers, so you should use an unsigned type such as unsigned long.

Would make it a point that the size of int is assumption to be

```
int foobar();  
is *not* a function prototype. To be correct I believe you should say:  
| Are the following two function declarations same?
```

Agreed.

..

```
Your example:  
| #include <stdio.h>  
| int main()  
| {  
| float a = 12.5;  
| printf("%d\n", a);  
| printf("%d\n", *(int *)&a);  
| return 0;  
| }
```

Re: c interview

may also not behave as *you* expect since it invokes undefined behaviour. If I recall correctly one implementation I have would print 12 on the first line and other implementations I have definitely would not.

Yes, But most of the questions (including this) are open ended questions. They are not multiple choice questions with fixed answers..
And it is also stated on the website (in the introduction before the programs):

<snip>

Most of the programs are meant to be compiled, run and to be explained for their behaviour

</snip>

You don't always check the value returned by malloc before using it. You should.

Yes.

Your example of a definition of the offsetof macro invokes undefined behaviour. It is *not* possible to implement it portably and this is probably why it is provided in a standard header.

It may invoke undefined bahaviour. But for the implementation (which is assumed to work), the reader is expected to find out how it is working.

Some of your other example questions are, in my opinion, plain stupid. This does not mean that they are never asked of course

Agreed.

But history says something more about the stupid mistakes[1]

Who have expected that the following stupid bug would have cause the major disruption of AT&T phone services throughout US. (AT&T's network was unusable for almost nine hours starting on afternoon of January 15,1990)

<snip>

```
network code()
{
switch (line) {
case THING1:
doit1();
break;
case THING2:
if (x == STUFF) {
do_first_stuff();
```

Re: c interview

Re: c interview

```
if (y == OTHER_STUFF)
break;
do_later_stuff();
} /* coder meant to break to here... */
initialize_modes_pointer();
break;
default:
processing();
} /* ...but actually broke to here! */
use_modes_pointer();/* leaving the modes_pointer
uninitialized */
}
</snip>
```

and fingerd code leading to a worm creating havoc in November 1998

```
<snip>
main(argc, argv)
char *argv[];
{
char line[512];
...
gets(line);
</snip>
```

And this causing an ANSI C compiler to become very slow:

```
<snip>

int hashval=0;
/* PJW hash function from "Compilers: Principles, Techniques,
and Tools"
* by Aho, Sethi, and Ullman, Second Edition.
while (cp < bound)
{
unsigned long overflow;
hashval = ( hashval <<4)+*cp++;
if ((overflow = hashval & (((unsigned long) 0xF) << 28)) != 0)
hashval ^= overflow | (overflow >> 24);
}
hashval %= ST_HASHSIZE; /* choose start bucket */
/* Look through each table, in turn, for the name. If we fail,
* save the string, enter the string's pointer, and return it.
*/
for (hp = &st_ihash; ; hp = hp->st_hnext) {
int probeval = hashval; /* next probe value */
...
.....
.....
</snip>
```

Re: c interview

You should acknowledge the original author of Duff's device. Tom Duff deserves recognition for his highly warped thinking. I suggest a link over to <http://www.lysator.liu.se/c/duffs-device.html> unless someone can suggest a better link

I would also suggest you include references to other useful resources, in particular the comp.lang.c FAQ at <http://c-faq.com/>

References and credits are badly missing. I would work on it in the near future.

When I hosted up those questions, I never thought the site would become so popular that it now ranks in top 20 for the search "c puzzles" in google. Now I'm making an effort to make the questions more correct and the material to be more useful.

Regards,

Gowri Kumar

[1] From the book "Expert C programming – Deep C secrets" –by Peter Van Der Lindenman

In fact many of the examples are taken directly from the above book (I do have permission for it from Peter).

.