

Re: division by 7 without using division operator

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-02/msg00918.html

- *From:* "Harald van D3k" <truedfx@xxxxxxxx>
 - *Date:* 6 Feb 2007 00:58:46 -0800
-

Richard Heathfield wrote:

CBFalconer said:

"Dik T. Winter" wrote:

"Francois Grien" <fgrien@xxxxxxxx> writes:

...

To succeed in these interviews, I fear that you must not be so smart as to outsmart the person asking the questions. So I suspect the right answer might have been

```
number *= 0.142857142857143; // divide by  
7
```

which indeed, in quite a few contexts, is an appropriate answer.

Indeed, it gives the correct answer for all integers from 0 to 2147483647.

And probably down to -2147483648.

An exercise for those with way too much time on their hands: what is the smallest-magnitude integer (i.e. regardless of sign) for which it fails? If this number is negative, what is the smallest *positive* integer for which it fails?

It depends on the implementation's precision and the rounding mode. One possible answer: C90 (but not C99) allows integer division to round towards negative infinity, so $-1 / 7$ could be -1 , but $-1 * 0.142857142857143$ is always 0.

Re: division by 7 without using division operator

Mathematically, 0.142857142857143 is equal to $1.000000000000001 / 7$, so another possible answer is easy to guess and confirm:
 1000000000000000 .

$1000000000000000 * 0.142857142857143$ is 142857142857143
 $1000000000000000 / 7$ is $142857142857142.8\dots$, which becomes 142857142857142 because floating point to integer conversions round towards zero. For any smaller positive number, the exact difference between the two numbers cannot be greater than $1/7$, so when rounding towards zero, it cannot make a difference. Also, when rounding towards zero, the sign doesn't matter.

There are other possible answers depending on the representation of double.

.