printf and long double

Asked 8 years, 10 months ago Active 1 month ago Viewed 225k times

52

10

I am using the latest gcc with Netbeans on Windows. Why doesn't `long double` work? Is the `printf` specifier `%lf` wrong?

Code:

```
#include <stdio.h>

int main(void)
{
    float aboat = 32000.0;
    double abet = 5.32e-5;
    long double dip = 5.32e-5;

    printf("%f can be written %e\n", aboat, aboat);
    printf("%f can be written %e\n", abet, abet);
    printf("%lf can be written %le\n", dip, dip);

    return 0;
}
```

Output:

```
32000.000000 can be written 3.200000e+004
0.000053 can be written 5.320000e-005
-19502285125096975000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000.000000
can be written 2.725000e+002
Press [Enter] to close the terminal ...
```

c gcc printf long-double

edited Nov 6 '16 at 23:35 — usr2564301 — 18.3k 7 35 77

asked Nov 3 '10 at 16:20 — gameboy — 622 3 11 16

bytes.com/topic/c/answers/... – gameboy Nov 3 '10 at 20:09

1 obviously there is a problem with MinGW and long double – gameboy Nov 3 '10 at 20:11

possible duplicate of can't print correctly a long double in C – phuclv Sep 21 '15 at 7:48

8 Answers

14

(microsoft C compiler use 64 bits long double for various reasons).

answered Nov 3 '10 at 16:35

AProgrammer
44k 7 75 127

bytes.com/topic/c/answers/… – gameboy Nov 3 '10 at 20:52

42

From the printf manpage:

> l (ell) A following integer conversion corresponds to a long int or unsigned long int argument, or a following n conversion corresponds to a pointer to a long int argument, or a following c conversion corresponds to a wint_t argument, or a following s conversion corresponds to a pointer to wchar_t argument.

and

> L A following a, A, e, E, f, F, g, or G conversion corresponds to a long double argument. (C99 allows %LF, but SUSv2 does not.)

So, you want `%Le` , not `%le`

Edit: Some further investigation seems to indicate that Mingw uses the MSVC/win32 runtime(for stuff like printf) - which maps long double to double. So mixing a compiler (like gcc) that provides a native long double with a runtime that does not seems to .. be a mess.

edited Nov 3 '10 at 17:24

answered Nov 3 '10 at 16:25

nos
182k 45 335 439

I found linux.die.net/man/3/printf, but %Lf or %Le also not working – gameboy Nov 3 '10 at 16:41

Which compiler are you using with Netbeans on Windows ? – nos Nov 3 '10 at 16:56

latest gcc and MinGW – gameboy Nov 3 '10 at 17:13

37

Yes -- for `long double` , you need to use `%Lf` (i.e., upper-case 'L').

answered Nov 3 '10 at 16:23

Jerry Coffin
399k 58 499 941

If you are using MinGW, the problem is that by default, MinGW uses the I/O resp. formatting functions from the Microsoft C runtime, which doesn't support 80 bit floating point numbers (`long`

+50 Depending on the nature of your project, you might also want to globally `#define printf __mingw_printf` or use `-D__USE_MINGW_ANSI_STDIO` (which enables the MinGW versions of all the `printf` -family functions).

edited Apr 23 '14 at 14:15

answered Feb 20 '13 at 19:16

klickverbot
3,153 4 16 21

6

Was having this issue testing long doubles, and alas, I came across a fix! You have to compile your project with -D__USE_MINGW_ANSI_STDIO:

> Jason Huntley@centurian /home/developer/dependencies/Python-2.7.3/test $ gcc main.c
>
> Jason Huntley@centurian /home/developer/dependencies/Python-2.7.3/test $ a.exe
> c=0.000000
>
> Jason Huntley@centurian /home/developer/dependencies/Python-2.7.3/test $ gcc main.c -D__USE_MINGW_ANSI_STDIO
>
> Jason Huntley@centurian /home/developer/dependencies/Python-2.7.3/test $ a.exe
> c=42.000000

Code:

```
Jason Huntley@centurian /home/developer/dependencies/Python-2.7.3/test
$ cat main.c
#include <stdio.h>

int main(int argc, char **argv)
{
   long double c=42;

   c/3;

   printf("c=%Lf\n",c);

   return 0;
}
```

answered Mar 15 '13 at 20:19

Jason Huntley
3,109 2 14 24

4

In C99 the length modifier for `long double` seems to be `L` and not `l`. `man fprintf` (or equivalent for windows) should tell you for your particular platform.

edited Nov 3 '10 at 17:13

answered Nov 3 '10 at 16:23

Jens Gustedt

2

As has been said in other answers, the correct conversion specifier is `"%Lf"` .

You might want to turn on the format warning by using `-Wformat` (or `-Wall` , which includes `-Wformat`) in the gcc invocation

```
$ gcc source.c
$ gcc -Wall source.c
source.c: In function `main`:
source.c:5: warning: format "%lf" expects type `double`, but argument 2 has type `long
double`
source.c:5: warning: format "%le" expects type `double`, but argument 3 has type `long
double`
$
```

answered Nov 3 '10 at 16:34

pmg
87.7k 10 102 175

0

printf and scanf function in C/C++ uses Microsoft C library and this library has no support for 10 byte long double. So when you are using printf and scanf function in your C/C++ code to print a long double as output and to take some input as a long double, it will always give you wrong result.

If you want to use long double then you have to use " __mingw_printf " and " __mingw_scanf " function instead of printf and scanf. It has support for 10 byte long double.

Or you can define two macro like this : " #define printf __mingw_printf " and " #define scanf __mingw_scanf "

Use standard format for long double : %Lf

answered Jul 27 at 7:52

Rupak Paul
1