2098 - I2P(II)2020_Chen_week2_HW Scoreboard

Time		
2020/09/22 23:57:00	Contest Ended	2020/09/29 23:59:00

Clarification						
# Problem	Asker	Description	Reply	Replier	Reply Time	For all team

Overview Problem

12237 - TA's Heartfelt

Status | Limits Submit

Description

"Oh, no! The problem is so hard! No, no, no... Am I too stupid?"

"Hey, my code is right! Why do I always get a WA or TLE?"

"The method is difficult to understand, why are TAs always give us such a hard problem?"

"Did you heard that? I heard that the problem of the other professorss course is far more easier..."
"Really? I should have chosen the other course..."

Yeah, we TA all know that it's hard for you to understand these new concepts. However, we do hope you to learn some knowledge that is important or useful. Such as "Prefix Sum", you can pre-calculate all the sum of (1,i) to get any sum of (i,j), which is far more faster than calculate the sum using for/while... right?

What about gcd or fast power(快速冪)? I think all of you may learn the split phase division(輾轉相除法), so it should be not too hard to you guys to learn that, right? For the part of fpw(fast power), it's not too hard to realize how it works, which is also wonderful that we could calculate n^m or the n-th Fibonacci number so fast, even without include<moon>!

We do hope you guys could enjoy learning these interesting concepts instead of just taking the course, getting an A/A+, and treat this course as a part of your 4.3 GPA.

If you don't want to learn these extra knowledge, or you're too busy to learn, you can just submit a brute-force solution without using any of the concepts above, you can still pass 3-5 testcases(only if the method is right).

So now, we're going to have another question. Don't worry, you can get an AC by brute force (though that would be a little difficult).

You guys have learned IEEE floating point number, right? If you forget it, rewind it, please. Our problem won't become easier for you :(

- You are given a IEEE-754 floating number with single precision(which is type "float" in C)
- You are going to output every bit of the number, from the largest bit to the smallest bit (including every 0)
- **Note that scanf("%f", &x); will automatically parse scientific numbers into floating point number, so you don't have to implement this part.**

Hint.1: You may solve this problem by using pointer tricks extremely easily. However, you can still solve this by using traditional way. :)

Hint.2: You may refer to this page to check the binary of a fp number.

Enjoy!

Input

The input contains multiple testcases, which means we'll input several floating point numbers, ended by EOF.

Note that there may be precision error due to the precision of floating point number. What we want you to output is the **binary bits of the floating point number stored in C variable**.

We recommand to use scanf as your input function.

You can take a look and observe the sample IO.

The range of the floating point number is $-10^{20} \sim 10^{20}$.

Output

Output every bit of the floating point number. Remember to print a '\n' at the end of every output.

Sample Input Download

```
28.759
18
213.888
-31.25
-10
0
1.481233e-13
1.074562e-29
1.514054e+22
-3.760964e-34
5.423886e-13
```

Sample Output Download

```
01000001111001100001001001101111
01000001100100000000000000000000
01000011010101011110001101010100
11000001111111010000000000000000
11000001001000000000000000000000
00000000000000000000000000000000
00101010001001101100010110100010
00001111010110011111001001110011
01100100010011010011000101001101
10000111111110011111010101111101
00101011000110001010101100111000
```

Tags

Discuss