Problem A
Miles 2 Km

Source file name: `miles2km.c`, `miles2km.cpp` *or* `miles2km.java`

It is a well known fact that you can use Fibonacci numbers to convert distances from miles 2 kilometers. Well, it's actually not a well known fact. It is, however, a well known fact that one mile is 1.6 km. Now let me explain how you can use Fibonacci numbers in order to convert (approximately) miles to kilometers. Lets first write down the first numbers of the Fibonacci sequence:

$$0, \ 1, \ 1, \ 2, \ 3, \ 5, \ 8, \ 13, \ 21, \ 34, \ 55, \ 89, \ 144, \ \ldots$$

So let's suppose you want to convert 5 miles into kms. Well, in the Fibonacci sequence the number following 5 is 8 so 5 miles is 8 kms. You don't believe me? ok, just do the maths: multiply 5 by 1.6... $5 \times 1.6 = 8$! Amazing right?. The bad thing is that it doesn't work all the time, for example, how do you convert 89 miles into kms?. According to the previous procedure it would be 144 km, but it is really $89 \times 1.6 = 142.4$, so there is an error of 1.6. Still amazing no?. Anyway, what would you do if you want to convert 6 miles into kilometers?. You could, for example, say $6 = 5 + 1$, so we convert 5 and 1 to kms and then we add up. 5 miles is 8 kms and 1 miles is 2 kms, adding up we obtain $8 + 2 = 10$ so 6 miles is approximately 10 kms. Of course, we could also say $6 = 2 + 2 + 2$, so we convert 2 miles to kms and obtain 3. So 6 miles is approximately $3 + 3 + 3 = 9$ kms. Which answer is closer to the real value?.

It is your task to write a program which finds the best conversion from miles to kilometers using the previous Fibonacci method. The best conversion is the one with the lowest error.

Input

The input will consist of several test cases. Each case will be a single line containing one integer number $0 < N \le 1000$. The end of input is indicated by a test case with $N = 0$.

The input must be read from standard input.

Output

For each test case in the input, your program must print the lower error e for converting the corresponding input to kilometers (rounding up to 2 decimal digits).

The output must be written to standard output.

Sample input	Output for the sample input
6 5 12 0	0.40 0.00 0.20