Problem A

Harmonic progression sum

The simplest harmonic progression is

$$1/2, 1/3, 1/4, 1/5, \cdots$$

Let $S_n = \sum_{i=1}^{n}(1/i)$, compute this sum to arbitrary precision after the decimal point.

Input

The input contains only one test case. The first line contains two values: the first is the number of digits *D* and the second is the value of *N*. Consider ($1 \leq D \leq 10^5$) and ($1 \leq N \leq 10^8$).

The input must be read from the standard input.

Output

The output contains only one line printing the value of the sum with exact *D* precision.

The output must be written to the standard output.

Example

Input	Output for the input
12 7	2.592857142857

```
#include <iostream>
#include <sstream>

using namespace std;

void sum(char* output, const long unsigned int d,
const long unsigned int n) {
  long unsigned int digits[d + 11];
  for (long unsigned int digit = 0; digit < d + 11;
++digit) {
     digits[digit] = 0;
  }
  for (long unsigned int i = 1; i <= n; ++i) {
     long unsigned int remainder = 1;
     for (long unsigned int digit = 0; digit < d +
11 && remainder; ++digit) {
          long unsigned int div = remainder / i;
          long unsigned int mod = remainder % i;
          digits[digit] += div;
          remainder = mod * 10;
     }
  }
  for (long unsigned int i = d + 11 - 1; i > 0; --
i) {
     digits[i - 1] += digits[i] / 10;
     digits[i] %= 10;
  }
  if (digits[d + 1] >= 5) {
     ++digits[d];
  }
  for (long unsigned int i = d; i > 0; --i) {
     digits[i - 1] += digits[i] / 10;
     digits[i] %= 10;
  }
  stringstream stringstreamA;
  stringstreamA << digits[0] << ".";
  for (long unsigned int i = 1; i <= d; ++i) {
     stringstreamA << digits[i];
  }
  stringstreamA << '\0';
  string stringA = stringstreamA.str();
  stringA.copy(output, stringA.size());
}

int main() {

  long unsigned int d, n;

  cin >> d >> n;

  char output[d + 10]; // extra precision due to
possible error

  sum(output, d, n);

  cout << output << endl;

  return 0;
}
```