

# 4th Marathon of Parallel Programming

## SBAC'09

*October 29<sup>th</sup>, 2009.*

### **Warmup Rules**

For all problems, read carefully the input and output session. For all problems, a sequential implementation is given, and it is against the output of those implementations that the output of your programs will be compared to decide if your implementation is correct. You can modify the program in any way you see fit, except when the problem description states otherwise. You must submit your source code and an execution script. The program to execute should have the name of the problem. You can submit as many solutions to a problem as you want. Only the last submission will be considered. The script must be a *Makefile* with two rules: *all* and *run*.

All teams have access to a test machine during the marathon. Your execution may have concurrent process from other teams. Only the judges have access to the target machine. The execution time of your program will be measured running it with time program and taking the user CPU time given. Each program will be executed at least twice with the same input and only the smaller time will be taken into account. The sequential program given will be measured the same way. You will earn points in each problem, corresponding to the division of the sequential time by the time of your program. The team with the most points at the end of the marathon will be declared the winner.

*This problem set contains 1 problem; pages are numbered from 1 to 1.*

# Problem A

## Harmonic progression sum

The simplest progression sum is

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

Let  $S_n = \sum_{i=1}^n (1/i)$ , compute these 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  $(1 \leq D \leq 10^{127})$  and  $(1 \leq N \leq 10^{127})$ .

*The input must be read from 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 standard output.*

### Example

Input	Output for the input
12 7	2,592857142857