



acm International Collegiate
Programming Contest

2012



event
sponsor

ACM International Collegiate Programming Contest 2012

Latin American Regional Contests

November 9th-10th, 2012

Warmup Session

This problem set contains 2 problems; pages are numbered from 1 to 3.

This problem set is used in simultaneous contests hosted in the following countries:

- Argentina
- Bolivia
- Brasil
- Chile
- Colombia
- Cuba
- Perú
- México
- República Dominicana
- Venezuela

Promotion in Brasil:



Sociedade Brasileira de Computação

Sponsorship in Brasil:



Fundação Carlos Chagas

General Information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename.c*, *codename.cpp* or *codename.java*, where *codename* is the capital letter which identifies the problem.

Input

1. The input must be read from standard input.
2. The input contains several test cases. Each test case is described using a number of lines that depends on the problem.
3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.
4. Every line, including the last one, has the usual end-of-line mark.
5. The end of input is indicated by the end of the input stream. There is no extra data after the test cases in the input.

Output

1. The output must be written to standard output.
2. The result of each test case must appear in the output using a number of lines that depends on the problem.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. Every line, including the last one, must have the usual end-of-line mark.
5. After the output of all test cases, no extra data must be written to the output.
6. To output real numbers, round them to the closest rational with the required number of digits after the decimal point. Test cases are such that there are no ties when rounding as specified.

Problem A

Hot Dogs

In 2012 a new world record was set in the famous Nathan's Hot Dog Eating Competition: the champion, Joey Chestnut, ate 68 hot dogs in ten minutes, an amazing increase from the 62 hot dogs eaten by the same Chestnut in 2011.

Nathan's Famous Corporation, in Brooklyn, NY, is responsible for the contest. They make delicious hot dogs, famous worldwide, but when it comes to math they are not so good. They want to apply for being listed in the Guinness World of Records, but for that they need to fill up a form describing basic facts about the competition. In particular, they are required to inform the average number of hot dogs eaten by participants during the contest.

Can you help them? They promised to pay you with one of their tasty hot dogs. Given the total number of hot dogs consumed and the total number of participants in the contest, you must write a program to determine the average number of hot dogs eaten by participants.

Input

Each test case is described using one line. The line contains two integers H and P representing respectively the total number of hot dogs consumed and the total number of participants in the contest ($1 \leq H, P \leq 1000$).

Output

For each test case output a line with a rational number representing the average number of hot dogs eaten by participants. The result must be output as a rational number with exactly two digits after the decimal point, rounded if necessary.

Sample input	Output for the sample input
10 90	0.11
840 11	76.36
1 50	0.02
33 1000	0.03
34 1000	0.03
36 1000	0.04
37 1000	0.04
1 1000	0.00

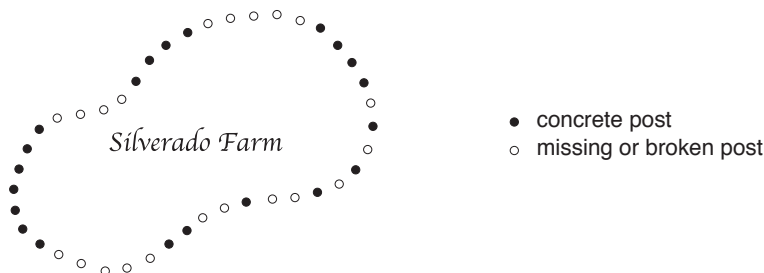
Problem B

Tornado!

Is this crazy weather the result of mankind's continuous interference in the environment? Or is it simply the normal cycle of climate changes through the ages? No one seems to know for sure, but the fact is that natural phenomena such as tornadoes and hurricanes have been hitting our country with more force and frequency than in past decades.

One tornado has just hit Silverado Farm, a cattle and milk producer, and made havoc. The barn roof was torn, several trees were uprooted, the farm truck was overturned... But the worst thing is that the tornado destroyed several sections of the fence that surrounded the property. The fence was very well built, with concrete posts every two meters, and barbed wire enclosing the whole farm perimeter (the perimeter, in meters, is an even number, making the fence perfectly regular).

Now several posts are broken or missing, and there are gaps in the fence. To prevent the cattle from getting out of the property, the fence must be restored as quickly as possible. Reconstructing the fence to its original form, with concrete posts, will take a long time. In the meantime, the farm owners decided to close the gaps with a temporary fence, made with wooden posts. Wooden posts will be placed in exactly the same spots where missing/broken concrete posts were/are. However, in order to make the temporary reconstruction faster and less expensive, the owners decided to use fewer posts: a wooden post will be used to replace a missing/broken concrete post only if the length of the barbed wired needed to close the distance to the next post (wooden or concrete) exceeds four meters.



Given the description of which posts are missing/broken, you must write a program to determine the smallest number of wooden posts needed to close all the gaps in the fence, according to the owners' decision.

Input

Each test case is described using two lines. The first line contains an integer N indicating the number of original concrete posts in the fence ($5 \leq N \leq 5000$). The second line contains N integers X_i indicating the state of each concrete post after the tornado ($0 \leq X_i \leq 1$ for $i = 1, 2, \dots, N$). If $X_i = 1$ post i is in good condition, if $X_i = 0$ post i is broken or missing. Note that post X_N is next to post X_1 .

Output

For each test case output a line with an integer representing the minimum number of wooden posts that are needed to restore the fence, according to the owners' decision.

Sample input	Output for the sample input
10	2
1 0 0 1 0 0 1 0 1 1	2
11	3
1 0 0 1 0 0 0 1 1 0 1	
12	
0 0 0 0 0 1 1 0 0 0 1 1	