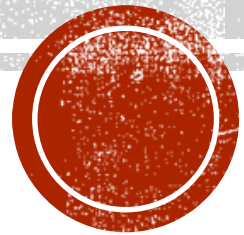
PROGRAMMING CLUB



Week 2 – Brute Force(Complete search)

Oct 25th, 2016

By Yizuo Chen, Yiyou Chen

DEFINITION

- Solving a problem using Brute Force(complete search) is based on the “Keep It Simple, Stupid'' principle. The goal of solving contest problems is to write programs that work in the time allowed, whether or not there is a faster algorithm. (From USACO training page)

PRIME NUMBER PROBLEM

- Find all the prime numbers smaller than n.
- Input: The single integer n
- Output: All the prime numbers smaller than n

PRIME NUMBER PROBLEM

- How to define a prime number?
- A **Prime Number** can be divided evenly only by 1, or itself.
- A loop from 2 to x-1, check if any number can be divided by x.
- A loop from 2 to n, check if x is a prime number.
- Algorithm Complexity: $O(n^2)$

What if n is very large?

PRIME NUMBER PROBLEM

- A small reduce:
- When check if x is a prime, go from 2 to $\sqrt{x}$.
- Algorithm Complexity: $O(n\sqrt{n}) = O\left(n^{\frac{3}{2}}\right)$.

PRIME NUMBER PROBLEM

- Faster way
- Every time find a prime number p, remove the multiples of p.
- For example, find 2 is a prime number, remove all the even numbers at the same time:

4, 6, 8, 10 …

after finding 3 is a prime number, remove 9, 15, 21 …

Algorithm complexity: $O(nloglogn) \approx O(n) = linear\ time$

The graph of log_2^n

$$loglog10000000 = 4.53938$$

BERU-TAXI

CODEFORCES ROUND #367 (DIV. 2)

- Vasiliy lives at point (a, b) of the coordinate plane. He is hurrying up to work so he wants to get out of his house as soon as possible. New app suggested n available Beru-taxi nearby. The i-th taxi is located at point (x_i, y_i) and moves with a speed v_i.
- Consider that each of n drivers will move directly to Vasiliy and with a maximum possible speed. Compute the minimum time when Vasiliy will get in any of Beru-taxi cars.

BERU-TAXI

- **Input**
- The first line of the input contains two integers a and b (- 100 ≤ a, b ≤ 100) — coordinates of Vasiliy's home.
- The second line contains a single integer n (1 ≤ n ≤ 1000) — the number of available Beru-taxi cars nearby.
- The i-th of the following n lines contains three integers x_i, y_i and v_i (- 100 ≤ x_i, y_i ≤ 100, 1 ≤ v_i ≤ 100) — the coordinates of the i-th car and its speed.
- It's allowed that several cars are located at the same point. Also, cars may be located at exactly the same point where Vasiliy lives.

BERU-TAXI

Output

Print a single real value — the minimum time Vasiliy needs to get in any of the Beru-taxi cars. You answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$

BERU-TAXI

Examples

input
0 0 2 2 0 1 0 2 2
output
1.000000000000000000000

input
1 3 3 3 3 2 -2 3 6 -2 7 10
output
0.500000000000000000000

BERU-TAXI

- Physics equation: $time = \frac{distance}{velocity}$
- Math equation: distance between p1 and p2: $d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$
- Brute force: find all the distances between taxis and home, then divide by the velocity of the taxis, find the smallest time.
- Keep more than 6 decimal places.

APPLEMAN AND EASY TASK

CODEFORCES ROUND #263 (DIV. 2)

- Toastman came up with a very easy task. He gives it to Appleman, but Appleman doesn't know how to solve it. Can you help him?
- Given a $n \times n$ checkerboard. Each cell of the board has either character 'x', or character 'o'. Is it true that each cell of the board has even number of adjacent cells with 'o'? Two cells of the board are adjacent if they share a side.
- **Input**
- The first line contains an integer n ($1 \le n \le 100$). Then n lines follow containing the description of the checkerboard. Each of them contains n characters (either 'x' or 'o') without spaces.
- **Output**
- Print "YES" or "NO" (without the quotes) depending on the answer to the problem.

APPLEMAN AND EASY TASK

Input:

3

xxo

xox

oxx

Output:

YES

Input:

4

xxxo

xoxo

oxox

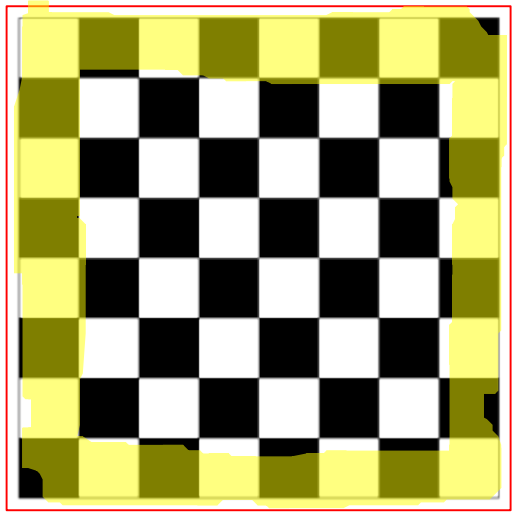
xxxx

Output

NO

APPLEMAN AND EASY TASK

- Search each cell of the board $g[i][j]$ $(0 \leq i, j < n)$
- For each cell, check if $g[i\ -1][j], g[i+1][j], g[i][j-1], g[i][j+1](cells\ around\ the\ cell)$ is 'O'.
- Calculate the number of cells around which is 'O' and check if the number is even or odd.
- How to check even number and odd number?



BUY A SHOVEL

Codeforces Round #377 (Div. 2)

Polycarp urgently needs a shovel! He comes to the shop and chooses an appropriate one. The shovel that Policarp chooses is sold for k burles. Assume that there is an unlimited number of such shovels in the shop.

In his pocket Polycarp has an unlimited number of "10-burle coins" and exactly one coin of r burles ($1 \le r \le 9$).

What is the minimum number of shovels Polycarp has to buy so that he can pay for the purchase without any change? It is obvious that he can pay for 10 shovels without any change (by paying the requied amount of 10-burle coins and not using the coin of r burles). But perhaps he can buy fewer shovels and pay without any change. Note that Polycarp should buy at least one shovel.

BUY A SHOVEL

- Input

The single line of input contains two integers k and r ($1 \le k \le 1000$, $1 \le r \le 9$) — the price of one shovel and the denomination of the coin in Polycarp's pocket that is different from "10-burle coins".

Remember that he has an unlimited number of coins in the denomination of 10, that is, Polycarp has enough money to buy any number of shovels.

- Output

Print the required minimum number of shovels Polycarp has to buy so that he can pay for them without any change.

- Examples

input
117 3
output
9
input
237 7
output
1
input
15 2
output
2

BUY A SHOVEL

- List the number of shovels he buys. All positive integers

- If the total cost of x shovels C satisfy $C \equiv 0 \ (mod\ 10)\ or\ C \equiv r \ (mod\ 10)$, exit the loop.

- loop (#of shovel from 1 to ∞){

  if (total cost of shovel $C \equiv 0 \ (mod\ 10)\ or\ C \equiv r \ (mod\ 10)$){

  break the loop and print # of shovel.

  }

}