Problem A: Egyptian Fractions

Input: egypt.in

In ancient Egypt, fractions were represented as sums of distinct unit fractions (reciprocals of positive integers). Today, of course, we represent them as an integer numerator divided by an integer denominator. As the pharaoh's chief computer scientist, you are to convert modern-day fractions, which are guaranteed to be strictly between 0 and 1, to their Egyptian fraction representations.

Luckily, it is known that if you repeatedly subtract the largest possible unit fraction, you will eventually end up with 0, and the fractions you took out will be one correct representation. Although you may get shorter representations using a different method, you are asked to follow this straightforward method.

For example, to represent 3/7:

3/7 – **1/3** = 2/21

2/21 – **1/11** = 1/231

1/231 – **1/231** = 0

Thus, 3/7 = 1/3 + 1/11 + 1/231.

There will be 5 lines of input. Each will consist of a fraction in the form M/N, where $1 \le N \le 1000$, and $1 \le M < N$. For each fraction, you are to output its Egyptian fraction representation, in the format $M/N = 1/a_1 + 1/a_2 + \ldots + 1/a_n$. The denominators must be arranged in increasing order.

You may assume that all denominators fit in unsigned 64-bit integers (**long long** in C++, **long** in Java).

Sample input	**Sample output**
2/3 6/14 113/355 1/2 499/500	2/3 = 1/2 + 1/6 6/14 = 1/3 + 1/11 + 1/231 113/355 = 1/4 + 1/15 + 1/609 + 1/864780 1/2 = 1/2 499/500 = 1/2 + 1/3 + 1/7 + 1/46 + 1/14206 + 1/1715374500

Problem B: BF Search

Input: bfs.in

A young couple is about to go on their first date in downtown Toronto. Unfortunately, the boy forgot where they planned to meet, and if he calls his romantic partner, his reputation will be tarnished. Of course, since contest programmers enjoy little reproductive success, it can be assured that he is not a coder. You are asked to assist him in his dire situation, in the name of love.

The boy has a map of the city (a square grid), and has marked a set of locations where his girlfriend might be. He wishes to find the smallest number of horizontal or vertical steps he can take, such that at least one of these locations is within his line of sight. The boy's line of sight is defined by the horizontal and vertical lines extending from the boy and ending at obstructions. The boy cannot leave the bounds of the map- that would be romantically counterproductive. On the map:

- 'B' indicates the boy's starting position.
- 'G' indicates where the girl might be.
- '.' indicates an empty square.
- '#' indicates an obstruction- the boy cannot walk or see through these.

The input file will contain 5 test cases. Each test case will consist of a line of two space-separated integers, denoting the height and width of the map (neither of which exceeds 100), followed by the map itself. Each line of output should contain the smallest number of steps the boy must take, such that he can see one of the girl's possible locations. If this is impossible, output a single line containing "Call!"

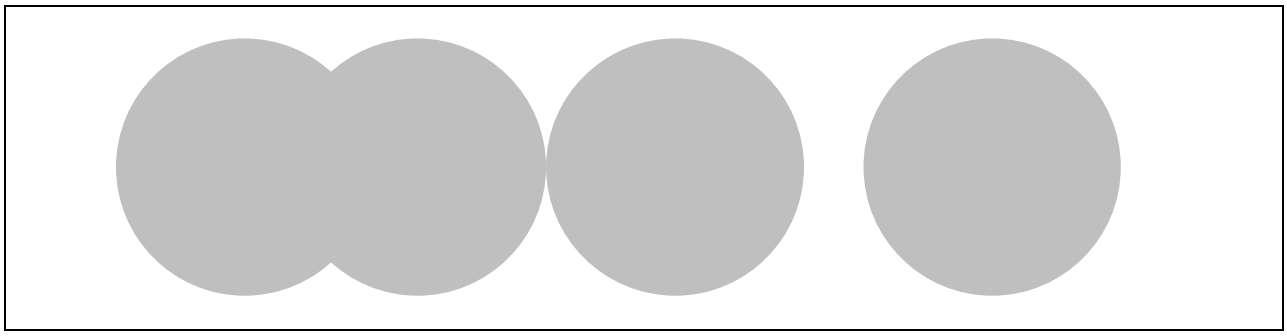
Sample input *(first three shown)*	**Sample output**

`9 8`
`.......G`
`.#..###.`
`.#.G##G.`
`.######.`
`....#G..`
`B######.`
`.#..#...`
`.####...`
`......#G`
`2 3`
`B.G`
`.G#`
`4 4`
`B..#`
`..#.`
`.#..`
`#..G`
	`5` *(five steps up)*
	`0` *(no need to move)*
	`Call!` *(the girl cannot be seen)*

Problem C: Circle Parade

Input: circle.in

You are a famous abstract artist, producing works such as this (entitled *La Parade*):



Indeed, all of your paintings consist of a number of circles with radius 1, centered on a horizontal axis. Unfortunately, your customers have had enough of your minimalistic style, and you are running severely low on money. In fact, you have become so poor that you can no longer afford to paint overlapping circles; you must create your masterpieces with a single layer of paint.

You would like to know the exact amount of paint you need for each of your paintings – no more, no less. In other words, you will be given a set of horizontal positions of circles, and you are to find the area of the union of these (possibly intersecting) circles.

The input file will contain 5 test cases. The first line of each test case will consist of one integer N ($1 \le N \le 100$), the number of circles in the painting. The next line will contain N space-separated real numbers x_i ($-1000 \le x_i \le 1000$), denoting the x-coordinates of the centers of N circles. For each case, you are to output one line, consisting of the area of the union of the circles, to 8 decimal places.

Sample Input	**Sample Output**
1 1 2 3.618033 1.618033 2 42.1337 42.1337 2 0 1 5 0 0.5 -0.5 1 -1	3.14159265 6.28318531 3.14159265 5.05481561 7.09952637

Problem D: March Madness

Input: march.in

March Break is almost upon us, and a certain bank robber Fran has requested for you to help her to make her holidays a little bit more… rewarding. She has retained you to write a program that will construct this plan; perhaps she will even spare you a penny from her exploits.

Given a set of N cities (numbered from 1 to N) and M two-way highways connecting the cities, Fran wishes to plan a journey starting from city 1 (her unfortunate hometown), and ending back at city 1, robbing each city of its riches. She may return to a city, but it may only be robbed once. Travelling on each highway consumes a given amount of gasoline. She would be caught at a gas station, so she cannot take a trip that consumes more than her car's capacity of G litres of gasoline. She wishes to know the maximum amount of money she can obtain, given these constraints.

The input file will contain 5 test cases. The first line in each test case will consist of three integers, N ($1 \le N \le 15$), M ($0 \le M \le 105$), and G ($0 \le G \le 100$). Each of the following M lines will contain three integers X_i ($1 \le X_i \le N$), Y_i ($1 \le Y_i \le N$, $Y_i \ne X_i$), and g_i ($1 \le g_i \le 100$), denoting a highway between cities X_i and Y_i that consumes g_i litres of gasoline. You may assume that at most one highway connects any two cities. The next N lines will contain B_i ($0 \le B_i \le 100$), the amount of money in city i.

You are to output the maximum amount of money that Fran can obtain this March Break.

Sample Input *(first one shown)*	**Sample Output**
5 6 11 1 2 3 2 3 6 2 4 1 2 5 4 3 5 4 4 5 3 5 3 4 4 6	12 *(Fran can take the path 1-2-4-2-1.)*