South Pacific
Competitive
Programming
Association

South Pacific Algorithmic Round 2 (SPAR 2)

Round 2 March 20, 2024

Contest Problems

A: Fizz-Buzz
B: A Walk in the Woods
C: Password Hacking
D: Election
E: Collatz Conjecture
F: Double Up
G: Particle Man

Problem set contains 7 problems over 16 pages

This page is intentionally left (almost) blank.

Problem A
FizzBuzz

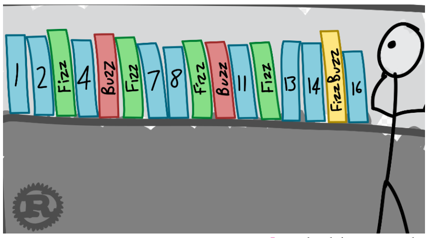

Image by chris morgan cc by

According to Wikipedia, FizzBuzz is a group word game for children to teach them about division. This may or may not be true, but this question is generally used to ~~torture~~ screen young Computer Science graduates during programming interviews.

Basically, this is how it works: you print the integers from 1 to N, replacing any of them divisible by X with `Fizz` or, if they are divisible by Y, with `Buzz`. If the number is divisible by both X and Y, you print `FizzBuzz` instead.

Check the samples for further clarification.

Input

Input file will contain a single test case. Each test case will contain three integers on a single line, X, Y and N ($1 \le X < Y \le N \le 100$).

Output

Print integers from 1 to N in order, each on its own line, replacing the ones divisible by X with `Fizz`, the ones divisible by Y with `Buzz` and the ones divisible by both X and Y with `FizzBuzz`.

Sample Input 1

```
2 3 7
```

Sample Output 1

```
1
Fizz
Buzz
Fizz
5
FizzBuzz
7
```

Sample Input 2

```
2 4 7
```

Sample Output 2

```
1
Fizz
3
FizzBuzz
5
Fizz
7
```

Sample Input 3

```
3 5 7
```

Sample Output 3

```
1
2
Fizz
4
Buzz
Fizz
7
```

Problem B
A (Fast) Walk in the Woods
Time limit: 3 seconds

Brice Bilson loves to take jogs in a nearby forest known as Orthogonal Woods. The forest gets that name as the paths – all two-way – are laid out along an orthogonal grid, with all turns being 90 degrees. Brice is a bit persnickety when it comes to his jogs, and always follows a set of rules when he reaches an intersection of two or more paths. These rules are

1. If there are three remaining branches, Brice takes the middle one.
2. If there are just two remaining branches, Brice takes the one on his left.
3. If there are no branches to take, Brice ends his jog and walks to the nearest exit.

Brice is persnickety in another way too. He has assigned each path an "interest value", which is a positive integer indicating how interesting that path is to jog. The higher the value, the more interesting the path is. If the value of a path is n, then Brice will jog on that path no more than n times in his jog. After the n^{th} pass that path will cease to exist as far as Brice is concerned (so, for example, any three-branch intersection using that path now becomes a two-branch intersection and any two-branch intersection becomes a one-branch intersection). An example is shown in Figure B.1 below:

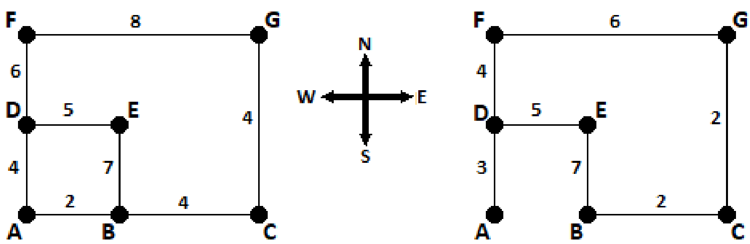

Figure B.1: Sample park corresponding to Sample Input 1

Suppose Brice enters the park at intersection D heading north in the figure on the left, where the numbers next to each path indicate his interest levels. His travels takes him on the route DFGCBADFGCBA at which point we reach the figure on the right, showing the updated interest levels of each path and the "removal" of the path from A to B since it's now been traversed 2 times. From intersection A Brice now traverses the route ADFGCBEDA at which point he hits a dead end and ends his jog.

Input

Input starts with two integers n and m ($2 \leq n \leq 2\,500$) giving the number of intersections and the number of paths between intersections. The next line contains n pairs of integers giving the locations of the intersections. Intersections are numbered from 1 to n in the order they are presented and all location values x, y satisfy $0 \leq x, y \leq 10^6$. After this are m lines each containing three integers i j k ($1 \leq i, j \leq n, 1 \leq k \leq 10^6$) indicating that a path exists between intersections i and j with interest level k. All paths will be either vertical or horizontal and will not touch any other vertices other than the specified intersection points. The final line of input contains an integer s ($1 \leq s \leq n$) and a character $d \in \{$N,S,E,W$\}$ indicating that Brice starts his jog by taking the path in direction d from intersection s. There will always be a path heading in direction d from vertex s.

Output

Output the location where Brice ends his jog.

Sample Input 1	Sample Output 1
7 8 0 0 5 0 12 0 0 5 5 5 0 10 12 10 1 2 2 2 3 4 4 5 5 6 7 8 1 4 4 2 5 7 3 7 4 4 6 6 4 N	0 0

Problem C
Password Hacking

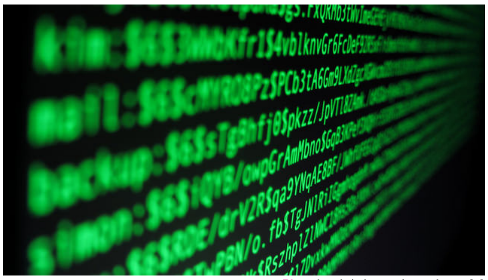
Photo by christiaan colen cc by-sa 2.0

We all know that passwords are not very secure unless users are disciplined enough to use passwords that are difficult to guess. But most users are not so careful, and happily use passwords such as "123456". In fact, there are lists of commonly used passwords that hackers can use for breaking into systems, and these passwords often work.

You have done a lot of hacking using such lists, and you have a good idea of how likely each password in the list is the correct one (you are very surprised by the number of people using "123456" as their passwords). You have a new account to hack, and you have decided to try each of the passwords in the list one at a time, until the correct one is found. You are absolutely sure that the account you want to hack uses a password in the given list.

What is the expected number of attempts to find the correct passwords, assuming that you try these passwords in the optimal order?

Input

The first line of input contains a positive integer N, the number of passwords in the list. Each of the next N lines gives the password, followed by a space, followed by the probability that the password is the correct one. Each password is a non-empty string consisting only of alphanumeric characters and is up to 12 characters long. Each probability is a real number with 4 decimal places. You may assume that there are at most 500 passwords in the list, and that the sum of all probabilities equals 1. No two passwords in the list are the same.

Output

Output on a single line the expected number of attempts to find the correct passwords using the optimal order. Answers within 10^{-4} of the correct answer will be accepted.

Sample Input 1	Sample Output 1
2 123456 0.6666 qwerty 0.3334	1.3334

Sample Input 2	Sample Output 2
3 qwerty 0.5432 123456 0.3334 password 0.1234	1.5802

This page is intentionally left (almost) blank.

Problem D
Election

Image by democratic alliance cc by-sa 2.0

After all the fundraising, campaigning and debating, the election day has finally arrived. Only two candidates remain on the ballot and you work as an aide to one of them.

Reports from the voting stations are starting to trickle in and you hope that you can soon declare a victory.

There are N voters and everyone votes for one of the two candidates (there are no spoiled ballots). In order to win, a candidate needs more than half of the votes. A certain number $M \leq N$ of ballots have been counted, and there are V_i votes for candidate i ($V_1 + V_2 = M$), where V_1 is the number of votes your candidate received.

Due to the historical data and results of highly scientific polls, you know that each of the remaining votes has a 50% chance to go to your candidate. That makes you think that you could announce the win before all the votes are counted. So, if the probability of winning strictly exceeds a certain threshold W, the victory is yours! We just hope you are sure of this, we don't want any scandals...

Input

The first line of input contains a single positive integer $T \leq 100$ indicating the number of test cases. Next T lines each contain four integers: N, V_1, V_2 and W as described above.

The input limits are as follows:
$1 \leq N \leq 50$
$50 \leq W < 100$
$V_1, V_2 \geq 0$
$V_1 + V_2 \leq N$

Output

For each test case print a single line containing the appropriate action:

- If the probability that your candidate will win is strictly greater than $W\%$, print
  `GET A CRATE OF CHAMPAGNE FROM THE BASEMENT!`
- If your candidate has no chance of winning, print
  `RECOUNT!`
- Otherwise, print
  `PATIENCE, EVERYONE!`

Sample Input 1	Sample Output 1
4 5 0 3 75 5 0 2 75 6 1 0 50 7 4 0 75	RECOUNT! PATIENCE, EVERYONE! PATIENCE, EVERYONE! GET A CRATE OF CHAMPAGNE FROM THE BASEMENT!

This page is intentionally left (almost) blank.

Problem E
Prof. Fumblemore and the Collatz Conjecture
Time limit: 2 seconds

The ***Collatz function***, C(n), on positive integers is:

$$n/2 \text{ if } n \text{ is even and } 3n{+}1 \text{ if } n \text{ is odd}$$

The ***Collatz sequence***, CS(n), of a positive integer, n, is the sequence

$$\text{CS}(n) = \text{n}, \text{C}(n), \text{C}(\text{C}(n)), \text{C}(\text{C}(\text{C}(n))), \ldots$$

For example, CS(12) = 12, 6, 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, . . .

The ***Collatz Conjecture*** (also known as the *3n+1 problem*) is that CS(n) for every positive integer n eventually ends repeating the sequence 4, 2, 1. To date, the status of this conjecture is still unknown. No proof has been given and no counterexample has been found up to very large values.

Prof. Fumblemore wants to study the problem using *Collatz sequence types*. The *Collatz sequence type* (CST) of an integer n, CST(n) is a sequence of letters E and O (for even and odd) which describe the parity of the values in CS(n) up to but not including the first power of 2. So,

$$\text{CST}(12) = \text{EEOEO}$$

Note that

$$\text{CS}(908) = 908, 454, 227, 682, 341, 1024, 512, 256, 128, 64, 32, 16, 8, 4, 3, 2, \ldots$$

so 12 and 908 have the same CST.

Prof. Fumblemore needs a program which allows him to enter a sequence of E's and O's and returns the **smallest** integer n for which the given sequence is CST(n).

Notes:

- E's are even numbers which are not powers of 2,
- O's are odd numbers greater than 1.
- The last letter in a sequence must be an O (if C(n) is a power of 2, so is n)
- There can not be two O's in succession (C(odd) = even)
- Since, Prof. Fumblemore does not type well, you must check that the input sequence is valid before attempting to find n. That is, the sequence contains only E's and O's, ends in O and no two O's are adjacent.

Input

Input consists of one line containing a string of up to 50 letters composed of E's and O's.

Output

There is one line of output that consists of the string `INVALID` if the input line is invalid, or a single decimal integer, n, such that n is the *smallest* integer for which CST(n) is the input sequence. Input will be chosen such that $n \leq 2^{47}$.

Sample Input 1	Sample Output 1
`EEOEO`	`12`

Sample Input 2	Sample Output 2
`EEOOEO`	`INVALID`

This page is intentionally left (almost) blank.

Problem F
Double Up

A Double Up game consists of a sequence of n numbers $a_1, \ldots, a_n$, where each a_i is a power of two. In one move one can either remove one of the numbers, or merge two identical adjacent numbers into a single number of twice the value. For example, for sequence $4, 2, 2, 1, 8$, we can merge the 2s and obtain $4, 4, 1, 8$, then merge the 4s and obtain $8, 1, 8$, then remove the 1, and, finally, merge the 8s, obtaining a single final number, 16. We play the game until a single number remains. What is the largest number we can obtain?

Input

The input consists of two lines. The first line contains n ($1 \leq n \leq 1000$). The second line contains numbers $a_1, \ldots, a_n$, where $1 \leq a_i \leq 2^{100}$ for each i.

Output

The ouput consists of a single line containing the largest number that can be obtained from the input sequence $a_1, \ldots, a_n$.

Sample Input 1	Sample Output 1
5 4 2 2 1 8	16

This page is intentionally left (almost) blank.

Problem G
K. Particle Man

Particle Man, Particle Man
Doing the things a particle can
What's he like? It's not important
Particle Man

Is he a dot, or is he a speck?
When he's underwater does he get wet?
Or does the water get him instead?
Nobody knows, Particle Man

Triangle Man, Triangle Man
Triangle Man hates Particle Man
They have a fight, Triangle wins
Triangle Man, ...

They Might Be Giants.

The above *They Might Be Giants* song describes a part of the ongoing saga between Particle Man and Triangle Man.

After Particle Man's last encounter with Triangle Man, he is very keen to avoid another thrashing. Being a particle he decides to hide in the real number line between 0 and 1 (inclusive). However, Triangle Man finds out and comes looking for Particle Man, but because Particle Man is so small (is he a dot, or is he a speck), Triangle Man can't see him.

"No matter", says Triangle Man, "using my Triangle powers I can divide any line into three equal parts, and smash the middle part to smithereens."

That is he can obliterate all points between $1/3$ and $2/3$ (exclusive) including any particles that might be hiding there. What's more worrying, is that he can do this all day long, so after smashing the line $[0, 1]$ into the smaller segments $[0, 1/3]$ and $[2/3, 1]$, he might then choose to smash the smaller segments, so $(1/9, 2/9)$ might be obliterated, leaving the segments $[0, 1/9]$, $[2/9, 1/3]$, and $[2/3, 1]$.

Particle Man is actually hiding in a rational number a/b where $a \leq b$ and both are non-negative 32 bit integers. Given a and b, determine if Particle Man is safe. If he is, print "Particle wins." if he isn't, print "Triangle wins."

Input

Each case will be a pair of integers, $0 \leq i \leq j < 10^9$. Particle Man will be hiding at the point i/j on the real number line.

Output

For each case, output a single line: "Particle wins." if Particle Man is safe; and "Triangle wins." otherwise.

Sample Input 1	Sample Output 1
`1 2`	`Triangle wins.`

Sample Input 2	Sample Output 2
`2 27`	`Particle wins.`