2006-2007 ACM International Collegiate Programming Contest

Asia Regional Contest
Shanghai Site

Shanghai Site First Round Internet Contest

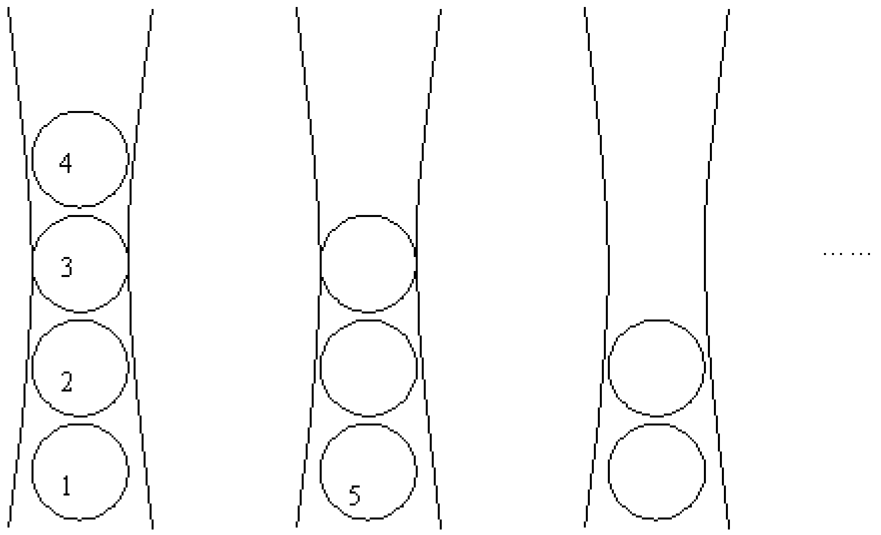
October 21, 2006

This problem set should contain eight (8) problems on fifteen (15) pages. Please

Balls

Time Limit : 1 seconds Memory Limit : 10MB

Tracy has been dreaming of becoming a basketball player for many years. But considering her current height... However, Tracy still loves basketball. One day she saw a basketball game: There were n big baskets with unlimited height but small width which only allows one ball to pass.



A player can shoot a basketball into arbitrary baskets. Basketballs are numbered from 1. The player should obey the following rules:

1. Balls with smaller numbers should be put under balls with larger numbers.
2. The sum of two neighboring balls' number should be a prime.
3. You can't shoot the basketball numbered i unless the basketball numbered i – 1 has already been successfully shot into one of the baskets.

The winner is the person who shoots most number of basketballs.
All the balls numbered from 1 to n should be shot into baskets.
Tracy wanted to win the game. She wanted to know how many balls can be put at most?

Input:

The input file contains several test cases. For each test case:
only one line contains one number n (0<= n < 16).

Output:

For each test case, output one number m, the maximal number of balls, on a separate line.

Sample Input:

```
1
```

Sample Output:

```
4
```

Communication

Time Limit : 5 seconds Memory Limit : 10MB

We have received an order to build a communication system on the Earth. The system consists of several cities connected with some on-ground and under-ground communication paths.

In this problem, we assume that the Earth is a perfect sphere with a radius of exactly 6378 km. On-ground paths are built on the surface of the earth, while under-ground paths are straight lines which pass through the earth.

As you know, constructing under-ground paths is a difficult job. Only K or less than K paths are allowed to build. However, you can build unlimited number of on-ground paths.

Your task is to construct some paths for given cities so that any two cities are connected by the paths directly or indirectly. For economic reasons, the total length of the path should be minimized.

The value of PI is approximately 3.141592653589793.

Input:

The input contains several cases.

The first line contains two numbers N (1 <= N <= 500) and K (0 <= K < N). The following N lines describe the locations of the N cities. Each line contains two real numbers, representing its latitude and longitude.
The latitude will be between -90 and +90. The longitude will be between -180 and +180 where negative numbers denote locations west of the meridian and positive numbers denote locations east of the meridian.

The input ends with a zero on a single line.

Output:

For each case, output a number representing the total length (km) of the paths, rounded to the nearest integer.

Sample Input:

```
2 0
-90.00 0.00
90.00 0.00
2 1
```

```
-90.00  0.00
90.00  0.00
0
```

Sample Output:

```
20037
12756
```

Even or Odd

Time Limit : 5 seconds Memory Limit : 10MB

We define f(p1,p2,…,pn) as the number of inversion pairs(an inversion pair is such a pair (pi, pj) which i < j and pi > pj). Here p1p2p3…pn is a permutation of 123..n, i.e. {p1,p2,p3,…,pn} = {1,2,…, n}.

We call permutation p1p2p3…pn an even permutation if f(p1,p2,…,pn) is even. As you can guess, the other permutations are called odd permutations.

Maybe you can easily calculate the function f, but we are only interested in the value (−1) f(p1,p2,…,pn).For this purpose, we want to know whether a permutation is even.

For example f(1,2,3,4) = 0, so 1234 is a even permutation; f(2,4,1,3) = 3, thus 2413 is an odd permutation.

Input:

The input contains several cases.

Each case has two lines, an integer N (1<=N<=2000000) on the first line and the permutation on the second line(numbers are separated with a blank). Input ends with an zero in a single line.

Output:

For each case, output one line. Output "EVEN" if the permutation is an even permutation, otherwise output "ODD".

Sample Input:

```
2
1 2
4
2 4 1 3
0
```

Sample Output:

```
EVEN
ODD
```

Hint:

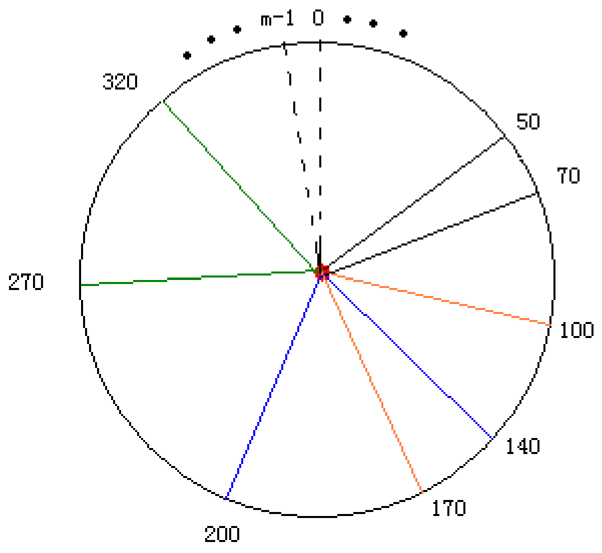
Huge input, cin(c++) and Scanner(Java) will TLE!

Firing range

Time Limit : 2 seconds Memory Limit : 10MB

After reading a novel about the special troops, Tracy dreams of being a special troop too! Now she and her companions were commanded to attack a missile station. But, of course, the enemy arranged army to defend it.

The station's figure is like the picture below:



The station is a circle which is divided into m parts. We call each part a degree. 0 degree is on the top while m-1 degree is the maximal degree. Degrees are numbered clockwise. The red point in the center is the target. Each enemy's firing range covers an arc of the circle. In the picture above, different color stands for different enemy.

Now Tracy wants to know the arcs which are covered by k(k is her lucky number) enemies' firing range so that they can enter the circle to finish the mission. But she doesn't know how to work on a circle. OK, please help her.

Input:

The input file contains several test cases. For each test case:
the first line contains three numbers n(0<=n<=100000) , m(1<=m<=100000) and k (0<=k<= n),n stand for the number of enemies. Each of the following n lines contains 2 numbers li, ri (-10^9<=li<ri<=li+m<=10^9), standing for one firing range1(in degree).Note li, ri may >=m, (li, ri)<=> (li-m, ri-m), and data like (300, 560) won't appear(because (300,560) = (300,200), li > ri is invalid).

Output:

The first line contains one number p, standing for the number of arcs which weren't covered by fire. The following p lines: each contains 2 numbers li, ri (-10^9<=li<ri<=10^9, li<m), separated by one space, standing for a arc. Arcs should be sorted in ascending order by using li as the first keyword, ri as the second keyword.

Sample Input:

```
4 360 0
50 70
100 170
140 200
270 320
```

Sample Output:

```
3
70 100
200 270
320 410
```

Lazy girl

Time Limit : 1 seconds Memory Limit : 10MB

Tracy and her classmates are learning combinatory. Today’ s homework is to calculate some equation like C_n^k. Tracy is a lazy girl, so she decides to calculate them with computer. Suppose you are the computer, then what’ s the answer?

Input:

The input file contains several test cases. For each test case:
There’ s only one line containing two integers n (n>=1) and k (0<=k<=n). Input is terminated by two zeroes for n and k.

Output:

For each test case, print one line containing the required number. This number will always fit into an 64-bit integer, i.e. it will be less than 2^{63}.

Sample Input:

```
4 2
10 5
49 6
0 0
```

Sample Output:

```
6
252
13983816
```

msn problem

Time Limit : 2 seconds Memory Limit : 10MB

As you know, MSN is a very common chatting software. Tracy also became a MSN user recently.

There's some friends in Tracy's contact list. Tracy wishes to see friends online when she is online so that they can have a talk. Unfortunately, Tracy often finds no friends online. Tracy has to get offline after some time.

Tracy expects to see friends online when she is online. So she decides to get friends' online time. But, as you know, our online time can't be steady. This troubled Tracy. However, Tracy is a curious girl, so she decides to solve this problem in another way.

Tracy writes a useful program for several months' . This program can report one's online time. Tracy puts it in her friends' computer. When a friend gets online, the program records the time until he or she gets offline. In this way, the program records a time interval. If Tracy is online, she can get this friend's online time interval.

But, Tracy finds a serious mistake in her program: when some programs work together, they may confuse each other. That is, program i's report may confuse with program j. So, program i's report may report friend j instead of friend i.

Tracy is so lazy that she doesn't want to correct her program. But she still wants to use it. So she thinks out a way to remedy. Tracy thinks that a person spend more time online if he is more active.(Let's believe this theory tentatively.)Tracy gives each of her friends a "active value"Ci. A big value stands for the person is more active. Besides, the online time which programs reported is recorded as Si.(Attention, only when both Tracy and a friends are online can Tracy got the report in current time.)

Tracy wants to know which friend does program i report. If program i reports friend j, let "difference value" Di = abs (Ci — Si). Tracy wants to find a best arrangement through making the sum of Di minimal. So she finds you to help her find the best arrangement. If the minimal sum is larger than "limit" , it means that the theory which Tracy believes isn't reliable, so that this program becomes not reliable too.

To make this problem more easily, Tracy only interests in the friends who are n' most active or spend n' most time online. That is, you only need to consider n' programs which report the most Si.

Input:

The input file contains several test cases. For each test cases:

the first line includes 3 numbers: n(1<=n<=2000), n'(1<=n'<=100), limit(0<=limit<=10^9). n means we have n friends and n programs.

the next line: first number m(m<=300), means Tracy's online time interval. Following 2*m number, each pair (l, r) stands for a time interval [l, r]. We promise that intervals may not intersect.

the following n lines: for ith line, first number m(m<=300), means program i reports m intervals. following 2*m number, each pair stands for a time interval. We promise that intervals may not intersect.

the following n' lines, for ith line, contains a number Ci(0<=Ci<=10^9).

All the time starts from time 1 and won't exceed 10^5.

Output:

the first line contain a number stands for the minimal sum of Di. If this value exceeds limit, you should output another line with words "Poor Tracy".

Sample Input 1:

```
3 3 5
2 4 10 19 20
2 3 8 10 11
3 1 3 8 9 15 20
3 1 9 10 15 19 19
7
8
1
```

Sample Output 1:

```
4
```

Details: Si = 6,4,8, Ci = 7,8,1. For arrangement program1->friend1, program2->friend3, program3->friend2, the minimal sum of Di = 4.

Sample Input 2:

```
3 2 1
2 4 10 19 20
1 5 11
3 1 3 8 9 15 20
1 1 10
8
5
```

Sample Output 2:

```
2
Poor Tracy
```

Details: considering Si = 6,7, Ci = 8,5. For arrangement program1->friend3, program2->friend2, the minimal sum of Di = 2.But this value exceed limit.

Hint:

Huge input, scanf(c++) and BufferedReader(Java) is recommended!

Travel

Time Limit : 1 seconds Memory Limit : 10MB

Tracy is employed to be a tour guide in a beautiful small town. The small town has n scenic spots; some of them are connected by paths. You can reach any spots from any spots. Tracy' s tourists are quite choosy: they didn' t want to be same with others. That means, every tourist wants his or her route to have a unique path compared with others. A route is made by paths from one spot to another, starts from one spot and return to the same spot without using the same path twice. So Tracy wants to know how many satisfied route this town has at most so that she can determine how many tourist she can take in. Can you help her?

Input:

The input file contains several test cases. For each test case:

The first line contains two numbers n(1<=n<=100000), m(1<=m<=n*(n-1)/2), standing for the number of spots and the number of paths. Then the following m lines: Each of the following m lines contains 2 number x, y(1<=x,y<=n), means there is a two-way path between spot x and spot y.

The size of the input file will be less than 2M.

Output:

only one number, the number of different route.

Sample Input:

```
5 7
1 2
1 3
1 4
2 4
2 3
3 4
5 4
```

Sample Output:

```
3
```

Walk

Time Limit : 1 seconds Memory Limit : 10MB

One day, Tracy felt tired after a whole day’s programming. She decided to leave from computer and have a walk. The room is a rectangle. Tracy started from computer, choosing an angle as she like, then start walking. Tracy would walk straight forever. If Tracy collides with a wall, she would be rebounded off it. The collision reduces no energy, and if we consider the route as light, the incident angle of the light is equal to the reflection angle. Tracy will keep walking until inspiration comes. The speed is constant. But, mum wanted to know the walking time so that she could remind Tracy go back to work.

Tracy wants to test mum, so she recorded the sequence of collision and told them to mum. Can you help mum to work out the total length of Tracy’s route?

Input:

The input file contains several test cases. For each test case:

The first line contains two numbers w, l (0<=w,l<=1000), the width and length of the room, respectively.

The second line contains two numbers x0, y0, representing the position of the computer is (x0, y0).

The third line contains two numbers x1, y1, representing the position of the inspiration is (x1, y1).

The last line is a string. Each character is one of ’F’,’B’,’L’,’R’, means the wall is in front, back, left or right. The length of the string will be less or equal than 1000.

All the coordinates are real numbers.

Output:

one real number, the total length of the route. The result should be accurate up to 4 decimals.

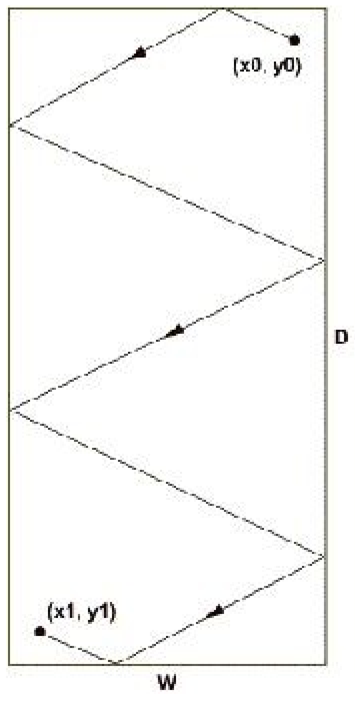
Sample Input:

```
10 20
9 1
1 19
FLRLRB
```

Sample Output:

```
52.8015
```

The picture below shows the sample:


(x0, y0)
D
(x1, y1)
W