

# ANZAC CONTEST 4

MAY 27, 2023

---

## Contest Problems

---

A: Ace Arbiter  
B: Berry Battle  
C: Coffee Cup Combo  
D: Disc District  
E: Enigmatic Enumeration  
F: Foreign Football  
G: Graduation Guarantee  
H: Highest Hill  
I : Icy Itinerary  
J : Junk Journey  
K: Keyboard Queries

This page is intentionally left (almost) blank.



## Problem A

### Ace Arbiter

Alice and Bob are playing a friendly game of ping pong. The game consists of several rounds. At the end of each round, exactly one player gets one point. The first player to 11 points wins. When that happens, the game immediately ends. Alice starts serving the first round, then Bob serves for the next two rounds, then Alice serves twice, then Bob serves twice, and so on.

At any time in the game, the current score is written  $X-Y$ , where  $X$  is the number of points of the player who is currently serving, and  $Y$  is the number of points of the other player. This constant switching of the scores back and forth can be a little bit error-prone. Eve is the umpire, and she wrote down a log of the current scores at a few different times during the game. But since Eve is a bit unreliable, you worry that she wrote down the wrong scores. Write a program which, given the list of scores Eve wrote down, checks whether or not this list of scores can appear in an actual game.

### Input

The first line of input contains an integer  $n$  ( $1 \leq n \leq 100$ ), the number of lines in the log. Then follow  $n$  lines, each containing a string of the form  $X-Y$  ( $0 \leq X, Y \leq 11$ ), the list of scores Eve wrote down, in chronological order.

### Output

If the input is a valid log of a not necessarily finished game, output "ok". Otherwise, output "error  $i$ ", where  $1 \leq i \leq n$  is the largest value such that the list of the first  $i - 1$  entries is valid, but the list of the first  $i$  entries is invalid.

#### Sample Input 1

```
5
0-0
1-0
1-0
2-0
1-2
```

#### Sample Output 1

```
ok
```

#### Sample Input 2

```
2
1-0
0-0
```

#### Sample Output 2

```
error 2
```

#### Sample Input 3

```
4
11-0
11-0
11-1
11-11
```

#### Sample Output 3

```
error 3
```



The South Pacific Competitive  
Programming Association



**Sample Input 4**

```
4
0-0
1-0
0-2
2-1
```

**Sample Output 4**

```
error 3
```



## Problem B

### Berry Battle

Berry picking is hard work, but also a peaceful and relaxing experience. After a long day of picking, it is common to see nothing but berries once you close your eyes to sleep. As your mind drifts into unconsciousness, the berries will start living their own life and create all kinds of absurd scenarios.

You are given a tree with  $n$  vertices numbered from 1 to  $n$ . Initially, there is one berry in each vertex. There is also one ant in each vertex, guarding the berries. When picking the berry at vertex  $v$ , all the ants that are on different vertices will walk one step towards  $v$ . The ants already at  $v$  will stay where they are. Note that since the graph is a tree, there is always one unique path the ants will take.



Picture by Jonas Bergsten, public domain

Your goal is to pick all the berries in the tree. The ants are no danger to you as long as they stay separated. But if at any point all the  $n$  ants end up in the same vertex, they will attack you. Find a permutation of the vertices, so that if you pick the berries in that order, all the ants will *not* end up in the same vertex.

### Input

The first line contains an integer  $n$  ( $2 \leq n \leq 3 \cdot 10^5$ ).

The following  $n - 1$  lines each contain two integers  $u$  and  $v$  ( $1 \leq u \neq v \leq n$ ), meaning that an edge goes between vertices  $u$  and  $v$ .

### Output

If it is impossible to find an answer, print “NO”.

Otherwise, first print “YES” on one line. On the second line, print  $n$  integers  $p_1, p_2, \dots, p_n$ , the order in which to pick the berries ( $1 \leq p_i \leq n$ ). This means that the  $i$ :th berry you pick is the one in vertex  $p_i$ .

#### Sample Input 1

```
10
1 2
2 3
3 4
3 9
3 7
7 10
1 5
5 6
1 8
```

#### Sample Output 1

```
YES
1 5 6 3 4 9 8 7 10 2
```

#### Sample Input 2

```
3
1 2
2 3
```

#### Sample Output 2

```
NO
```

This page is intentionally left (almost) blank.



## Problem C

### Coffee Cup Combo

Jonna is a university student who attends  $n$  lectures every day. Since most lectures are way too simple for an algorithmic expert such as Jonna, she can only stay awake during a lecture if she is drinking coffee. During a single lecture she needs to drink exactly one cup of coffee to stay awake.

Some of the lecture halls have coffee machines, so Jonna can always make sure to get coffee there. Furthermore, when Jonna leaves a lecture hall, she can bring at most two coffee cups with her to the following lectures (one cup in each hand). But note that she cannot bring more than two coffee cups with her at any given time.

Given which of Jonna's lectures have coffee machines, compute the maximum number of lectures during which Jonna can stay awake.

#### Input

The first line contains the integers  $n$  ( $1 \leq n \leq 10^5$ ), the number of lectures Jonna attends.

The second line contains a string  $s$  of length  $n$ . The  $i$ 'th letter is 1 if there is a coffee machine in the  $i$ 'th lecture hall, and otherwise it is 0.

#### Output

Print one integer, the maximum number of lectures during which Jonna can stay awake.

##### Sample Input 1

```
10
0100010100
```

##### Sample Output 1

```
8
```

##### Sample Input 2

```
10
1100000000
```

##### Sample Output 2

```
4
```

##### Sample Input 3

```
1
0
```

##### Sample Output 3

```
0
```

This page is intentionally left (almost) blank.





## Problem D

### Disc District

You live in the Disc District of Flatland and work for the Nearest Convenient Plot Company (NCPC). Your job is to find the nearest convenient plot of land outside of the Disc District to build on.

The Disc District can be described as a disc on a plane with center  $(0, 0)$  and some radius  $r$ . So a point is outside of the Disc District if the (Euclidean) distance from it to the origin is strictly larger than  $r$ . A point  $(x, y)$  on the plane is a *convenient plot* if  $x$  and  $y$  are integers.

#### Input

The only line of the input contains a single integer  $r$  ( $1 \leq r \leq 10^6$ ).

#### Output

The output should contain a single line with two integers, the  $x$  and  $y$  coordinates of a convenient building location that is closest to the Disc District. If there are more than one answer, output any of them.



Mount Taranaki in New Zealand. Image by NASA.

#### Sample Input 1

1

#### Sample Output 1

1 1

#### Sample Input 2

8

#### Sample Output 2

4 7

#### Sample Input 3

90210

#### Sample Output 3

69551 57450

This page is intentionally left (almost) blank.



## Problem E

### Enigmatic Enumeration

Your friend Cajsa had a graph with  $n$  vertices, and she needed to find its shortest cycle. To do this, she just took a random sequence of vertices and luckily this happened to be a shortest cycle. “What are the odds?”, she asked herself and wrote another program to calculate this probability.

To do this, Cajsa needed an algorithm to count the number of shortest cycles in a graph. She uses a homemade randomized algorithm that runs in  $\mathcal{O}(1)$ . But you suspect that this algorithm is incorrect, because surely it would have to consider every vertex of the graph (right?). You think that Cajsa’s algorithm just prints random numbers that happen to be correct on some small graphs.

In response to these doubts, Cajsa generated a bunch of graphs, and challenges you to check that her answers are correct.

You are given an undirected graph with  $n$  vertices and  $m$  edges, and your task is to count the number of shortest cycles in it.

A *cycle* in a graph is a path of **distinct** vertices where, additionally, there is an edge between the first and last vertices of the path. Two cycles are considered distinct if they don’t consist of the same set of edges. Thus the cycles 3, 1, 2 and 3, 2, 1 are the same, and the cycles 1, 2, 3 and 2, 3, 1 are considered the same.

### Input

The first line of input contains two integers  $n$  and  $m$  ( $3 \leq n \leq 3000$ ,  $3 \leq m \leq 6000$ ), the number of vertices and the number of edges.

The following  $m$  lines each contain two integers  $u_i$  and  $v_i$  ( $1 \leq u_i \neq v_i \leq n$ ), indicating that an undirected edge goes between  $u_i$  and  $v_i$ . The graph will not contain duplicate edges or self-loops.

It is guaranteed that the graph contains at least one cycle. However, note that the graph does *not* have to be connected.

### Output

Print one integer, the number of shortest cycles in the graph.

#### Sample Input 1

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

#### Sample Output 1

```
1
```



**Sample Input 2**

```
5 10
1 2
1 3
1 4
1 5
2 3
2 4
2 5
3 4
3 5
4 5
```

**Sample Output 2**

```
10
```

**Sample Input 3**

```
6 6
1 2
2 3
3 1
4 5
5 6
6 4
```

**Sample Output 3**

```
2
```



## Problem F

### Foreign Football

You are on vacation in a foreign country. This country has a local football league, and you don't know any of the team names. However, you have found a table of all the results from this season, and next to every match is the concatenated names of the two teams that played.

There are  $n$  teams in total, named  $s_1, s_2, \dots, s_n$ . You are given the concatenation  $s_i + s_j$  for every ordered pair  $i \neq j$ . Find the teams names  $s_1, s_2, \dots, s_n$ . Team names must be nonempty, but they do not need to be distinct.

#### Input

The first line of input contains the integer  $n$  ( $2 \leq n \leq 500$ ).

The following  $n$  lines each contain  $n$  strings, the table of concatenated team names. The  $j$ :th string on the  $i$ :th of these lines will contain the string  $s_i + s_j$  if  $i \neq j$ , and "\*" if  $i = j$ . The concatenated team names will consist of lower case characters a-z.

The total number of characters in concatenated team names is at most  $10^6$ .

#### Output

If there is no solution, print "NONE".

If there is more than one solution, print "MANY".

If there is one unique solution, print "UNIQUE", followed by  $n$  lines containing  $s_1, s_2, \dots, s_n$ .

##### Sample Input 1

```
3
* difaik difhammarby
aikdif * aikhammarby
hammarbydif hammarbyaik *
```

##### Sample Output 1

```
UNIQUE
dif
aik
hammarby
```

##### Sample Input 2

```
2
* aaaa
aaaa *
```

##### Sample Output 2

```
MANY
```

##### Sample Input 3

```
3
* a ab
a * b
ba b *
```

##### Sample Output 3

```
NONE
```

##### Sample Input 4

```
2
* zz
zz *
```

##### Sample Output 4

```
UNIQUE
z
z
```

This page is intentionally left (almost) blank.



## Problem G

### Graduation Guarantee

Gustav is studying to become an interpreter. In this line of work, knowing several languages is a must, and Gustav is already fluent in French, Mandarin, Nahuatl and even Finnish. But there is one language that Gustav struggles with: Norwegian. Before Gustav can graduate, he must complete the infamous Norwegian Conclusive Proficiency Check.

This exam consists of  $n$  yes/no questions. Answering correctly gives +1 point, answering incorrectly gives -1 point, and not answering gives 0 points. To pass the exam, at least  $k$  points are needed. For each question, Gustav has a guess that he knows is correct with probability  $p_i$  ( $0.5 \leq p_i \leq 1$ ). Note that  $p_i \geq 0.5$ , because otherwise guessing the opposite would be better.

What is the maximum possible probability of passing the exam, assuming we carefully choose which questions to answer and which ones to leave unanswered? Note that unlike a programming contest, the exam does not have instant feedback. So Gustav will answer the questions, hand in his answers, and only later be told the results.



Picture by Milford, public domain

### Input

The first line contains the two integers  $n$  and  $k$  ( $1 \leq k \leq n \leq 5000$ ). The second line contains  $n$  real numbers  $p_i$  ( $0.5 \leq p_i \leq 1.0$ ). These numbers will have at most 6 digits after the decimal point.

### Output

Output the probability that we pass the exam. Your answer will be considered correct if it has an absolute or relative error of at most  $10^{-6}$ .

#### Sample Input 1

```
3 3
0.5 0.5 0.5
```

#### Sample Output 1

```
0.125
```

#### Sample Input 2

```
4 1
0.9 0.5 0.9 0.9
```

#### Sample Output 2

```
0.972
```

This page is intentionally left (almost) blank.

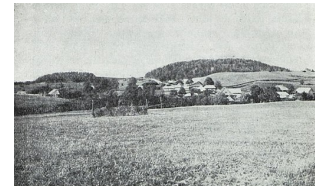




## Problem H

### Highest Hill

Sweden may not have a particularly impressive mountain range compared to other NCPC countries such as Norway and Iceland, but at least it beats the flatlands of Denmark. The situation is not so clear when comparing other member countries though. For example, is Estonia more mountainous than Lithuania<sup>1</sup>? To settle this question, you want to determine which of the two countries has the most impressive mountain peak.



An old photograph of Suur Munamägi, public domain

A mountain range is defined by sampling the heights  $h_i$  of  $n$  equidistant points.

Within a mountain range, we call a triple of indices  $1 \leq i < j < k \leq n$  a *peak* if  $h_i \leq \dots \leq h_j \geq \dots \geq h_k$ . The *height* of a peak is defined as the smaller of  $h_j - h_i$  and  $h_j - h_k$ .

Given a mountain range, can you find the height of its highest peak?

### Input

The first line contains a single integer  $N$  ( $3 \leq n \leq 200\,000$ ), the number of sampled points of the mountain range.

The second and final line contains the heights  $h_1, \dots, h_N$  ( $0 \leq h_i \leq 318 \cdot 10^9$ ) of the sampled points, in nanometers above sea level.

It is guaranteed that the mountain range contains at least one peak.

### Output

Output a single integer: the height of the highest peak.

#### Sample Input 1

```
11
0 1 2 3 4 5 4 3 2 1 0
```

#### Sample Output 1

```
5
```

#### Sample Input 2

```
10
29 85 88 12 52 37 19 86 7 44
```

#### Sample Output 2

```
67
```

#### Sample Input 3

```
3
2147483647 318000000000 2147483647
```

#### Sample Output 3

```
315852516353
```

#### Sample Input 4

```
3
1 1 1
```

#### Sample Output 4

```
0
```

<sup>1</sup>Yes, but not by much! The highest point in Lithuania is Aukštojas Hill, 293.84 meters above sea level, while Estonia has the highest peak in the Baltics: Suur Munamägi is 318 meters above sea level.

This page is intentionally left (almost) blank.



## Problem I Icy Itinerary

It is a harsh and cold winter, in a town consisting of  $n$  houses. Everybody is asleep in the early morning hours. The snow lies deep on the ground, except on the  $m$  roads that connect some pairs of houses. Only Thomas is awake.

Thomas the mailman needs to deliver mail to each of the  $n$  houses in the town. The houses are numbered from 1 to  $n$ . Thomas will start in his own house (house 1), and then visit all the other houses in some order. He can use a bicycle to get between two houses connected by a road, and he can use skis if there is no road between the houses. But it is not possible to use skis on roads and the bicycle outside of roads. Switching between bicycle and skis is tedious, so Thomas would like to do this at most once.



Picture by Stöhrfall, public domain

Your task is to find an ordering  $a_1, a_2, \dots, a_n$  of the  $n$  houses so that  $a_1 = 1$ , and there is at most one index  $i$  ( $2 \leq i \leq n - 1$ ) such that either

1.  $a_{i-1}$  and  $a_i$  are connected by a road, but  $a_i$  and  $a_{i+1}$  are not, or
2.  $a_{i-1}$  and  $a_i$  are not connected by a road, but  $a_i$  and  $a_{i+1}$  are.

In other words, the sequence starts at 1 and switches between using roads and non-roads at most once.

### Input

The first line contains two integers  $n, m$  ( $2 \leq n \leq 3 \cdot 10^5$ ,  $0 \leq m \leq 3 \cdot 10^5$ ), the number of houses and the number of roads.

The next  $m$  lines each contain two integers  $u_i, v_i$  ( $1 \leq u_i \neq v_i \leq n$ ), meaning that  $u_i$  and  $v_i$  are connected by a road. The roads can be used in both directions, and all the unordered pairs  $\{u_i, v_i\}$  are distinct.

There exists a valid ordering for every case in the input.

### Output

Print  $n$  integers  $a_1, a_2, \dots, a_n$  on one line, the order in which to visit the houses.

Remember that the first number  $a_1$  should be 1.

#### Sample Input 1

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

#### Sample Output 1

```
1 4 3 2
```

#### Sample Input 2

```
5 0
```

#### Sample Output 2

```
1 2 3 4 5
```

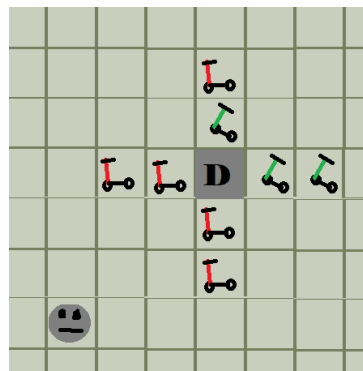
This page is intentionally left (almost) blank.



## Problem J Junk Journey

You are MALL-E, the mall scooter scooper robot. Your job is to fetch all the misplaced mall scooters at the end of the day and return them to the scooter depot. The mall is an infinite grid, containing  $n$  scooters that need to be returned. You can move in one of four directions: up, down, left, or right, and each move takes one second. If you move to a location that contains a scooter, that scooter moves to the next location in the same direction you were moving. If a scooter moves into a location that contains another scooter, the same thing happens, so there is never more than one scooter at each location and you never occupy the same location as a scooter. If a scooter moves to the scooter depot, it disappears. However, the robot can freely move over the scooter depot.

The mall will soon open. You need to find a way to get all the scooters to the depot in at most  $10^5$  seconds.



A picture of Sample Input 2. The two leftmost scooters can be pushed in the depot by moving up three times and moving right twice.

### Input

The first line of the input contains a single integer  $n$  ( $1 \leq n \leq 50$ ). The next line contains four integers  $x_0, y_0, x_t, y_t$  ( $0 \leq x_0, y_0, x_t, y_t \leq 30$ ). This indicates that you start at  $(x_0, y_0)$  and the depot is located at  $(x_t, y_t)$ . Then follow  $n$  lines, each containing two integers  $x$  and  $y$  ( $0 \leq x, y \leq 30$ ). This indicates that there is a scooter located at  $(x, y)$ . You, the depot, and all the scooters will all have distinct locations.

### Output

The output should contain a sequence of instructions, each on its own line. An instruction is one of the following strings: “up”, “down”, “left”, or “right”.

#### Sample Input 1

```
1
0 0 2 0
1 0
```

#### Sample Output 1

```
right
```



**Sample Input 2**

```
8
1 1 4 4
5 4
6 4
3 4
2 4
4 5
4 6
4 3
4 2
```

**Sample Output 2**

```
up
up
up
right
right
down
down
down
right
up
up
right
right
right
up
left
left
up
up
up
left
down
down
```



## Problem K

### Keyboard Queries

Katrín and her friends are university students, and they go to a seminar every week. At the start of each seminar, the professor divides the students into groups randomly. Katrín and her friends dislike random groups, they want to form a group together so they can chat with each other and not get to know the other students. The professor has a secret string  $S$  consisting of  $n$  characters on their computer. This string acts as a seed for the random group generation. When generating groups, a program `manager` is ran with a substring of  $S$  as input. But sometimes, the professor mistypes and writes `manacher` instead. This will indicate that the substring is a palindrome. Maybe Katrín can use this information to predict what the group division will look like?

There is a secret string  $S$  with  $n$  characters in an unknown alphabet. You are given  $q$  queries of two types.

- 1  $l$   $r$ : This means that the substring of  $S$  at indices  $l$  through  $r$  is a palindrome.
- 2  $a$   $b$   $x$   $y$ : This means that you should determine whether the substring at indices  $a$  through  $b$  is equal to the substring at indices  $x$  through  $y$ , given the information in the previous queries.

### Input

The first line of input contains two integers  $n$  and  $q$  ( $1 \leq n \leq 10^5$ ,  $1 \leq q \leq 2 \cdot 10^5$ ), the number of characters in the unknown string and the number of queries.

The following  $q$  lines each start with a 1 or a 2 indicating the type of query. For queries of type 1, two integers  $l$  and  $r$  follow ( $1 \leq l \leq r \leq n$ ), meaning that the corresponding substring of  $S$  is a palindrome. For queries of type 2, four integers  $a, b, x, y$  follow ( $1 \leq a \leq b \leq n$ ,  $1 \leq x \leq y \leq n$ ), meaning that you should determine whether those two substrings are equal or not.

### Output

For each query of the second kind print “Equal” if the substrings must be equal, “Not equal” if the substrings can’t be equal, and “Unknown” if either possibility could be true given the information thus far.

#### Sample Input 1

```
6 8
1 1 6
2 1 1 6 6
2 1 2 5 6
2 1 3 5 6
1 1 3
2 1 3 4 6
2 4 4 6 6
2 2 3 4 5
```

#### Sample Output 1

```
Equal
Unknown
Not equal
Equal
Equal
Unknown
```

This page is intentionally left (almost) blank.