KTH Challenge

KTH Challenge 2018

April 8, 2018

Problems

Do not open before the contest has started.

This page is intentionally left (almost) blank.

Problem A
Bus Planning
Problem ID: busplanning

Picture by AutoPhoto via Wikimedia Commons, cc by

An elementary school class is going on a road trip by bus to a computer factory. However, the driver is really tired of having to drive fighting kids all day, so she suggested that the class should split up and go to the factory in groups. She has already observed which kids dislike each other and are likely to fight, so she wants to make sure they do not end up in the same group. Of course, driving everyone one-by-one is a waste of time and money, so the driver wants to minimize the number of groups she has to drive. In addition, the bus is pretty small, and may only fit up to c kids at a time.

You are to write a program which helps her with the task of making these groups. Given the number of kids and their enemies, find the minimum number of groups required, as well as a division of the class into this minimum number of groups

Input

The first line contains three integers n, k, and c ($1 \leq n \leq 17$, $0 \leq k \leq \frac{n(n-1)}{2}$ and $1 \leq c \leq n$) – the number of kids, pairs of enemies, and the capacity of the bus, respectively. Then follow n lines with the kids' names. Each name consists solely of the characters `A-Z` and `a-z`, is non-empty, and at most 10 characters long. Then follow k lines, each containing a pair of space-separated names indicating a pair of kids that dislike each other. No pair of names appears twice, and no kid is their own enemy.

Output

On the first line, output the minimum number of groups, followed by one line per group containing the names of the children in that group (separated by spaces).

Sample Input 1	Sample Output 1
2 0 1 Alice Bob	2 Alice Bob

Sample Input 2	Sample Output 2
3 2 3 Alice Charlie Bob Alice Charlie Bob Charlie	2 Alice Bob Charlie

This page is intentionally left (almost) blank.

Problem B
Cars
Problem ID: cars

Picture by Sammmerbatter via Wikimedia Commons, cc by-sa

You work as an engineer at the leading self-driving car company Wayber. Wayber just got approval from the government of Sweden for their cars to drive in the streets of Stockholm. Unfortunately, all your tests so far have been with one car at a time on a closed dirt track. You are not really prepared for what will happen when two of the cars encounter each other in the city, and want to write some code to handle this.

The cars are already good at staying inside the lines and turning. However, they are not very good at detecting other moving vehicles. Fortunately, the government of Sweden is so enthusiastic about self-driving cars that they have banned all other forms of transportation in Stockholm, including human-driven cars, biking, and walking. If you can just detect whether two cars will collide you will be able to build a safe system.

You know that all cities consist of only north-south and east-west oriented streets in a perfect grid, and all cars are perfect rectangular prisms. When detecting collisions you only need to worry about cars traveling at a constant speed without turning.

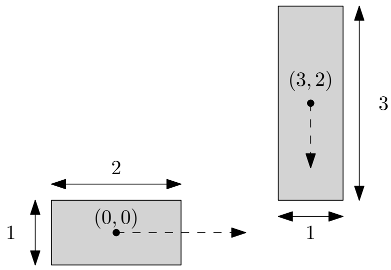


Figure B.1: Illustration of Sample Input 1.

Input

The first line of input contains an integer $1 \leq t \leq 10^4$, the duration in seconds of the trajectory of the two cars. Then follow the description of the trajectory of two cars.

A trajectory consists of a line containing a character d and five integers x, y, s, w and l. The starting position of the car is (x, y) $(0 \leq x, y \leq 10^4)$ and its direction is d which is either `N` (positive y direction), `S` (negative y direction), `W` (negative x direction) or `E` (positive x direction). The car is travelling at a speed of $1 \leq s \leq 10^4$ units per second, is $1 \leq w \leq 10^4$ units wide, and $1 \leq l \leq 10^4$ units long.

Cars start out centred on their starting coordinates and do not initially overlap in a nonzero area.

Output

For each line of input, output a line "`crash`" if the two cars will crash or "`safe`" if they will not crash. If two cars overlap in an area of size zero (only on an edge or corner) they do not crash.

Sample Input 1

```
5
E 0 0 2 1 2
S 3 2 1 1 3
```

Sample Output 1

```
crash
```

Sample Input 2

```
1
E 0 0 1 1 1
N 0 2 1 1 1
```

Sample Output 2

```
safe
```

Sample Input 3

```
2
N 0 0 7 3 1
S 3 20 12 4 1
```

Sample Output 3

```
crash
```

Sample Input 4

```
1
N 0 0 7 3 1
S 3 20 12 4 1
```

Sample Output 4

```
safe
```

Problem C
Color Codes
Problem ID: colorcodes

Picture by designmilk on Flickr, cc by-sa

Gray codes are a classic topic in information theory with a number of practical applications, none of which we are concerned with in this problem. An n-bit Gray code is an ordering $(x_1, x_2, \ldots, x_{2^n})$ of all n-bit binary strings, with the property that any consecutive pair of strings differ in exactly 1 bit. More formally, for every $1 \leq i < 2^n$, it holds that $d(x_i, x_{i+1}) = 1$, where $d(\cdot, \cdot)$ denotes the Hamming distance between two binary strings. For instance, for $n = 3$, the sequence $(000, 001, 011, 010, 110, 111, 101, 100)$ is a Gray code.

While Gray codes are great, they are also a bit, well... gray[1]. In this problem, we look at a much more colorful variant.

For an integer $n \geq 1$ and set of integers $P \subseteq \{1, \ldots, n\}$, we say that an ordering $(x_1, \ldots, x_{2^n})$ of all n-bit binary strings is an *n-bit color code with palette P*, if for all $1 \leq i < 2^n$, it holds that $d(x_i, x_{i+1}) \in P$, i.e., the number of bits by which any consecutive pair of strings differ is in P.

Note that for some palettes, color codes do not exist. For instance, if $n = 6$ and $P = \{6\}$, the second string must be the binary negation of the first one, but then the third string must be the negation of the second one, i.e., equal to the first string.

Given n and P, can you construct an n-bit color code with palette P?

Input

The first line of input consists of two integers n ($1 \leq n \leq 16$) and p ($1 \leq p \leq n$). Then follow a line with p distinct integers $s_1, \ldots, s_p$ ($1 \leq s_i \leq n$ for each i) – the elements of P.

Output

If there is an n-bit color code with palette P, output 2^n lines, containing the elements of such a code, in order. If there are many different codes, any one will be accepted. If no such code exists, output "`impossible`".

Sample Input 1	Sample Output 1
6 1 6	impossible

[1] With apologies to Frank Gray.

Sample Input 2

```
3 1
1
```

Sample Output 2

```
000
001
011
010
110
111
101
100
```

Sample Input 3

```
4 2
3 2
```

Sample Output 3

```
0110
1101
1011
0001
0111
1100
1001
0000
0101
0011
1111
1010
0100
1000
0010
1110
```

Problem D
#exclude<scoring>
Problem ID: excludescoring

You are participating in a programming contest cup. The cup consists of a series of programming contests, followed by a final at the end of the season for the 15 top ranked contestants in the cup. With only one contest left to go before the final, you are starting to wonder if your performance in the earlier contests has been good enough to already secure you a spot in the finals. If so, you could succumb to your laziness and skip the last contest.

The ranking of the cup works as follows. In each contest, a contestant earns some number of points between 0 and 101 (the details of this are described below). Their *aggregate score* is then defined to be the *sum of the four highest scores* achieved. For instance if a contestant got 45, 15, 32, 0, 30, and 20 points over 6 contests, their aggregate score is $45 + 32 + 30 + 20 = 127$. The *rank* of a contestant X *in the cup* is defined to be 1 plus the number of contestants that have a strictly larger aggregate score than X.

The score a contestant earns from a contest is based on the rank they achieve *in that contest*, according to the following table.

Rank	**Points**	**Rank**	**Points**	**Rank**	**Points**
1	100	11	24	21	10
2	75	12	22	22	9
3	60	13	20	23	8
4	50	14	18	24	7
5	45	15	16	25	6
6	40	16	15	26	5
7	36	17	14	27	4
8	32	18	13	28	3
9	29	19	12	29	2
10	26	20	11	30	1

If a contestant gets a worse rank than 30, they get 0 points. If two or more contestants get the same rank in the contest, they are instead assigned the average points of all the corresponding ranks. This average is always rounded up to the closest integer. For example, if three contestants are tied for second place they all receive $\lceil \frac{75+60+50}{3} \rceil = 62$ points, and the next contestant will have rank 5 and receives 45 points (or less, if there is a tie also for 5'th place). This applies also at rank 30, e.g., if $4\,711$ contestants are tied for 30'th place, they all receive 1 point.

Contestants may participate in every contest either on-site or online. If they compete on-site, they get 1 extra point, no matter their original number of points. If a contestant does not participate in a contest, they get 0 points.

Input

The first line of input contains two integers n and m ($2 \le n \le 10$, $1 \le m \le 10^5$), where n is the number of contests in the cup (excluding the final), and m is the number of people who participated in any of the first $n - 1$ contests.

Then follow m lines, each describing a contestant. Each such line consists of $n - 1$ integers $0 \le s_1, \ldots, s_{n-1} \le 101$, where s_i is the score that this contestant received in the ith contest.

The first contestant listed is you. The point values in the input might not correspond to actual points from a contest.

Output

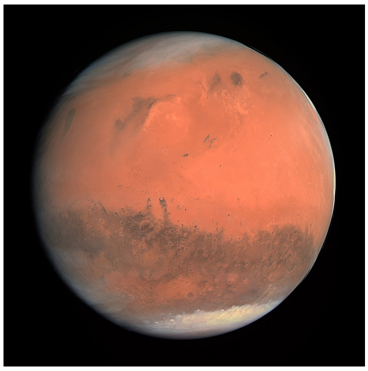
Output a single integer r, the worst possible rank you might end up in after the last contest, assuming you do not participate in it.

Sample Input 1	Sample Output 1
4 2 50 50 75 25 25 25	2

Sample Input 2	Sample Output 2
5 2 50 50 50 50 25 25 25 25	1

Sample Input 3	Sample Output 3
2 4 90 1 3 2	3

Problem E
Mars Window
Problem ID: marswindow


Picture by the European Space Agency, cc by-sa

You are an eccentric billionaire with an affinity for sending stuff to Mars. In an ideal world you would send stuff to Mars all the time, but your engineers tell you that it is most efficient to do it when Earth and Mars are somewhat close to each other.

Your engineers have calculated that optimal launch windows occur once every 26 months, and that one of them occurs in April 2018. They also tell you that they will not have any Big Finished Rockets by then, so you will have to wait for a later launch window.

Since your rocket scientists apparently can not be bothered to tell you about the optimal launch windows before it is too late, you have to keep track of that yourself. Write a program that determines if there is an optimal launch window in any given year.

Input

The only line of input contains an integer y ($2018 \leq y \leq 10000$), the year you are interested in.

Output

Output "`yes`" if there is an optimal launch window in the year y, otherwise output "`no`".

Sample Input 1	Sample Output 1
`2018`	`yes`

Sample Input 2	Sample Output 2
`2019`	`no`

Sample Input 3	Sample Output 3
`2020`	`yes`

Sample Input 4	Sample Output 4
`2028`	`no`

This page is intentionally left (almost) blank.

Problem F
Mathemagicians
Problem ID: mathemagicians

There are n mathemagicians standing in a circle. Each mathemagician wears either a blue hat or a red hat. A mathemagician can cast a color change charm which changes the color of their hat to the same color as the hat of the mathemagician directly to the left or to the right of them (the caster of the spell may choose which one of them). Note that mathemagicians are polite people and do not like interrupting each other, so only one mathemagician at a time may perform mathemagic.

The mathemagicians are not happy with their current hat configuration, so they would like to use the color change charm repeatedly to enter another hat configuration. Time isn't an issue because they can conjure cookies to eat.

Input

The first line contains an integer n ($3 \leq n \leq 10^5$), the number of mathemagicians. The next contains a string of length n. If the ith mathemagician wears a blue hat in the beginning, the ith character of the string is '`B`', otherwise the ith character is '`R`'. Finally, the third line contains a string of length n. If the ith mathemagician would like to wear a blue hat in the end, the ith character of the string is '`B`', otherwise the ith character is '`R`'.

It is guaranteed that not every mathemagician is happy with their hat color in the beginning.

Output

Output "`yes`" if it is possible for the mathemagicians to achieve the desired hat configuration after a finite number of color change charms, otherwise output "`no`".

Sample Input 1	Sample Output 1
5 BRBBR RBBRR	yes

Sample Input 2	Sample Output 2
6 RBRBRB BRBRBR	no

This page is intentionally left (almost) blank.

Problem G
Pairing Socks
Problem ID: pairingsocks

Picture by Villy Fink Isaksen, cc by-sa

Simone's mother often complains about how Simone never helps with chores at home. In return, Simone often points out that many of the chores her mother assigns her are NP-complete to perform optimally (like cleaning the house, seating her little brothers around the dinner table in a conflict-free way, splitting the brothers' Halloween loot in a fair manner and so on).

Being a computer scientist, her mother finds this a fair objection. Looking over her list of potential chores, she picked one she thinks should be easy to solve – pairing a number of different kinds of socks.

In the beginning, there are $2n$ socks stacked in a pile. To pair the socks, Simone can repeatedly make one of three moves:

1. Move the sock from the top of the original pile to the top of an auxiliary pile (which is originally empty).
2. Move the sock from the top of the auxiliary pile to the top of the original pile.
3. Pair the top socks from each pile together, if they are of the same type.

Simone only has one auxiliary pile, for a total of two piles. There may be more than two socks of each type. In this case, Simone can pair them up however she wants.

Your task is to help Simone to determine the least number of moves she needs to pair the socks, if it is possible at all.

Input

The first line of input contains the integer n ($1 \leq n \leq 10^5$) as described above. The next line contains $2n$ integers $a_1, \ldots, a_{2n}$ ($1 \leq a_i \leq 10^9$ for each i), where a_i denotes the type of sock number i. Initially, sock 1 is at the top of the pile and sock $2n$ is at the bottom.

Output

If Simone can pair all the socks, output the least number of moves she needs to do this. If it is impossible to do so, output "`impossible`" (without the quotes).

Sample Input 1	Sample Output 1
2 1 2 2 1	4

Sample Input 2	Sample Output 2
1 3 7	impossible

Sample Input 3

```
3
5 5 5 5 5 5
```

Sample Output 3

```
6
```

Problem H
Tree Hugging
Problem ID: treehugging

Once, two trees forgot their place and started to grow into each other. One of the trees grew from the left, and the other from the right. On n points, they collided.

Numbering the points $1, 2, \ldots, n$ from left to right, the left tree ended up connecting all of them in a single subtree rooted in node 1, such that every node's children had larger numbers than the node itself. We can describe this subtree with a list of $n - 1$ edges.

Similarly, the right tree also connected all nodes in a single subtree rooted in node n, with every node's children having smaller numbers than the node itself. This yields an additional $n - 1$ edges.

Now, given the full list of $2(n-1)$ edges, it is not necessarily easy to tell which edge belongs to which tree. Can you figure out a possible assignment, or determine that it is impossible for this collection to have been the union of two trees?

Input

The first line of input contains the integer n ($2 \leq n \leq 10^5$). The next $2(n-1)$ lines each contain two integers u, v ($1 \leq u < v \leq n$) indicating an edge joining the two nodes u and v. A pair (u, v) may be connected by more than one edge.

Output

If it is possible for the edges to be the union of two trees that grow left-to-right and right-to-left, output a string of length $2(n-1)$, where the i's character is `L` if the i'th edge should come from the left tree, or `R` if it should come from the right tree. Otherwise, output the word "`impossible`" on a single line. If there are multiple solutions, you may output any one of them.

Explanation of Sample Inputs

In the first example, there are two solutions: `LLRRRRLL` and `LLRLRRLR`.

In the second example, there are no solutions. Note that `LRLR` is not valid, because it would involve the right tree growing backward, from left to right.

Sample Input 1	Sample Output 1
5 1 2 2 5 2 3 1 3 3 5 4 5 3 4 1 3	LLRRRRLL

Sample Input 2

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

Sample Output 2

```
impossible
```