HIIT open 2024

November 16, 2024

Problems

A Adventure
B Budgetting bridges
C Conspiracies everywhere
D Dominoes
E Equilateral numbers
F Forgotten measurements
G Gerbil's run
H Hiitism
I Inheritance
J Just do it
K Key cutting
L Light rail connections

Adventure

You are visiting Hiitland, a country with n states and m two-way connections between them. When you arrive to one of the states from another state, you need to present certain legal documents at the border control. More precisely, there are 64 legal documents that are used in Hiitland, and passing each border check requires some specific subset of them.

Your travel plan consists of three steps. First, you arrive to some state by an airplane and stay there at a hotel. Then, you visit some other states, at least three states in total. Finally, you return back to the state where your hotel is and fly back home. Traveling to the same state multiple times is boring, so you would never do that, apart from returning the state you started from and flying home. Further, obtaining the legal documents is a hassle, so you plan your trip such that the number of legal documents you have to gather is not significantly larger than it has to be. That is, if a travel plan can be made with some number of legal documents, you want to gather at most twice that many documents.

Input

The first line has two integers, n and m.

The following m lines start with three numbers, v, u and c, describing that the you need c documents to travel between states v and u. The line continues by c integers describing the needed documents.

Output

Output the states you visit on your trip in the order you visit them. After the last state, you return to the state you started from and fly back home.

Constraints

- $3 \leq n \leq m \leq 1000$
- Each document is numbered by an integer between 1 and 64
- There's at most one connection between two states.

Example

Input:

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

Output:

```
2 5 1
```

Explanation: You fly to state 2, then travel to state 5, which requires document 3. Then you continue your trip to state 1, requiring document 2. Finally, you return back to state 2, which requires document 1. There exists a travel plan for which only two legal documents are needed, so obtaining three documents is not too many.

Budgetting bridges

The city of Hiitsburgh consists of n islands. In the past, people of Hiitsburgh have travelled between the islands by boats, but now the city wants to modernize and connect its islands by bridges. However, the budget for the project is limited, and so the bridges should be built in such a way that it minimizes the total cost while enabling travel between any pair of islands. The city council has received a report describing the cost of building a bridge between some pairs of islands and is now discussing which bridges should be built. Some council members are lobbying for certain bridges to be included to the plan, but there are concerns that this would increase the total cost of the project. Can you help the city council to figure out if the given sets of bridges can be included so that the islands can still be connected with minimum cost?

Input

The first line of the input has three integers n, m, and k, the number of islands, the number of possible bridges, and the number of plans the council members have.

The following m lines have three integers v, w, and c, stating that the cost of building a bridge between islands v and w is c. Each pair of islands appears at most once.

Finally, there are descriptions of the plans of the council members. The first line of the plan has a single integer b, describing the number of bridges in the plan. The following b lines have two integers v and w, meaning that we must build a bridge between islands v and w. The islands v and w are in the same order as when the possible bridges were described.

Output

For each plan, output either "YES" or "NO" depending on whether there exists a minimum-cost solution that connects all islands and contains the desired bridges.

Constraints

- $1 \leq n, m, k \leq 2 \cdot 10^5$
- You may assume that building all bridges connects all islands.
- The costs of the bridges are at most 10^9.
- The sum of all b is at most $2 \cdot 10^5$.

Example

Input:

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

Output:

```
YES
NO
```

Explanation: For the first plan, we can build a bridge between islands 1 and 2 and islands 1 and 3, resulting in a total cost of 2. For the second plan, the cost is at least 3.

Conspiracies everywhere

You suspect that the pyramids were built by aliens. A pyramid is a structure consisting of n layers of stones such that the ith layer of stones from the top has i stones in it. You observe that the quality of the building materials increases the higher the stones are, with the topmost stone having quality q. If aliens had built the pyramids, you believe that the qualities of the stones would be positive integers such that the quality of a stone is exactly the sum of the qualities of the two stones below it. Can you find the qualities of the stones in the bottom layer of a given pyramid to prove that aliens built them?

Input

The only line of the input has two integers n and q.

Output

Print any valid way to choose the stone qualities in the bottom layer. If there are no solutions, print "The truth is out there".

Constraints

- $1 \leq n \leq 60$
- $1 \leq q \leq 10^{18}$

Example 1

Input:

```
3 8
```

Output:

```
2 1 4
```

Explanation: the qualities of the stones are

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

Example 2

Input:

```
2 1
```

Output:

```
The truth is out there
```

Dominoes

You are playing with dominoes. Each domino tile can be represented as a pair of two distinct numbers between 1 and 6. You want to build a long chain of domino tiles without rotating them such that all adjacent numbers are distinct. Can you put all domino tiles into this chain?

Input

The first line of the input contains the integer n, the number of domino tiles.

The following n lines contain two distinct integers between 1 and 6, describing each domino tile.

Output

If a solution exists, output $2n$ numbers on a single line describing the sequence of numbers in the chain.

Otherwise, output "Impossible".

Constraints

- $1 \leq n \leq 10^5$

Example

Input:

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

Output:

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

Equilateral numbers

The kth equilateral number is obtained by taking the sum of the first k positive integers. For example, the 5th equilateral number is $1 + 2 + 3 + 4 + 5 = 15$.

Given a positive integer n, you want to represent it as a sum of equilateral numbers. How many numbers do you need at least?

Input

The only line of the input contains the integer n.

Output

Output the smallest amount of equilateral numbers whose sum is n.

Constraints

- $1 \leq n \leq 10^{12}$

Example

Input:

```
5
```

Output:

```
3
```

Explanation: You need at least three equilateral numbers (you can choose 1, 1, and 3).

Forgotten measurements

You want to help your grandparents build a fence at their summer cottage. You went to a store to purchase materials for that, but you forgot to measure how long the fence is going to be. Fortunately, you still remember the general shape of the fence and that the sections of the fence are aligned either horizontally or vertically on a map. To figure out the total length, you call to your grandparents and ask them to take the measurements for you, but to avoid disturbing them too much, you want to minimize the number of sections whose lengths they need to measure.

Input

The only line of the input has a string describing the shape of the fence. From the starting point, you first go some distance left ('L'), right ('R'), up ('U'), or down ('D'), then continue to the direction indicated by the second character, and so on. The path ends back at the starting point.

Output

Output the minimum number of measurements your grandparents need to take.

Constraints

- The length of the string is at most 10^6.

Example

Input:

```
ULDRUR
```

Output:

```
2
```

The fence from the example is illustrated below.

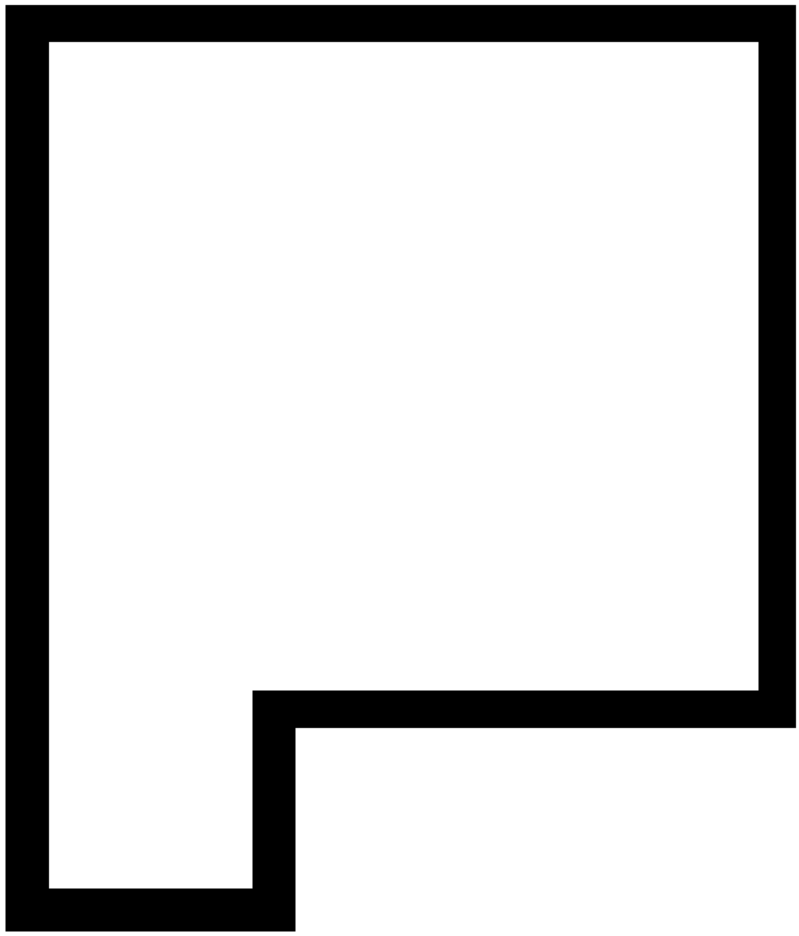

Figure 1: Illustration of the example

Gerbil's run

You have gotten yourself a gerbil and a nice large cage for it. You have even bought a lot of different toys for it to play with: a ball, a set of dominoes, and even a real Klein bottle! And still, to your great dismay, your gerbil seems to be upset with its new home! You ponder for a long time how you could make your gerbil happy. What could you still get your gerbil‽ And then it strikes you: The gerbil needs a wheel to run in! And no ordinary gerbil wheel suffices; you decide to order one with a custom paint job!

You know that the radius of the wheel needs to be exactly r units of length, and that the coloring must satisfy strict requirements. For simplicity, you model the wheel as a circle where each circular arc has a specific color. The coloring of the circle needs to satisfy the following requirements:

- The wheel needs to be fully painted with *blue* and *orange* colors. The blue color calms the gerbil down while the orange color encourages the gerbil to move faster!
- When the gerbil is standing on an *orange* point of the wheel, there must be a *blue* point at distance exactly 1 in either direction. Otherwise the gerbil might get too stressed.
- When the gerbil is standing on a *blue* point of the wheel, there must be an *orange* point at distance exactly 1 in both directions. Otherwise the gerbil might get bored.

To refer to the points of the wheel, you use the following convention: You mark one of the points of the wheel as the origin. You then refer to each point by their distance from the origin along the wheel in counter-clockwise orientation. Hence every point is assigned a real number between $[0, 2\pi r]$.

All distances, apart from the radius, are measured along the arc of the wheel.

Input

Input consists of one integer r, the radius of the gerbil wheel.

Output

Output a list of arc segments of the wheel that need to be colored *blue*. The rest of the wheel is colored *orange*.

Each line of the output consists of two rational numbers, x and y, describing a half-open interval $[x, y)$ of points on the wheel that are colored *blue*. They must satisfy $0 \leq x < y \leq 2\pi r$. Each rational number $\frac{p}{q}$ must be written in form `p/q`, where $1 \leq q \leq 10^6$.

The output may consist of at most 10^5 segments.

Constraints

- $1 \leq r \leq 1000$

Example

Input:

```
1
```

Output:

```
0/1 2/2
21/10 31/10
46/11 57/11
```

The example is illustrated below.

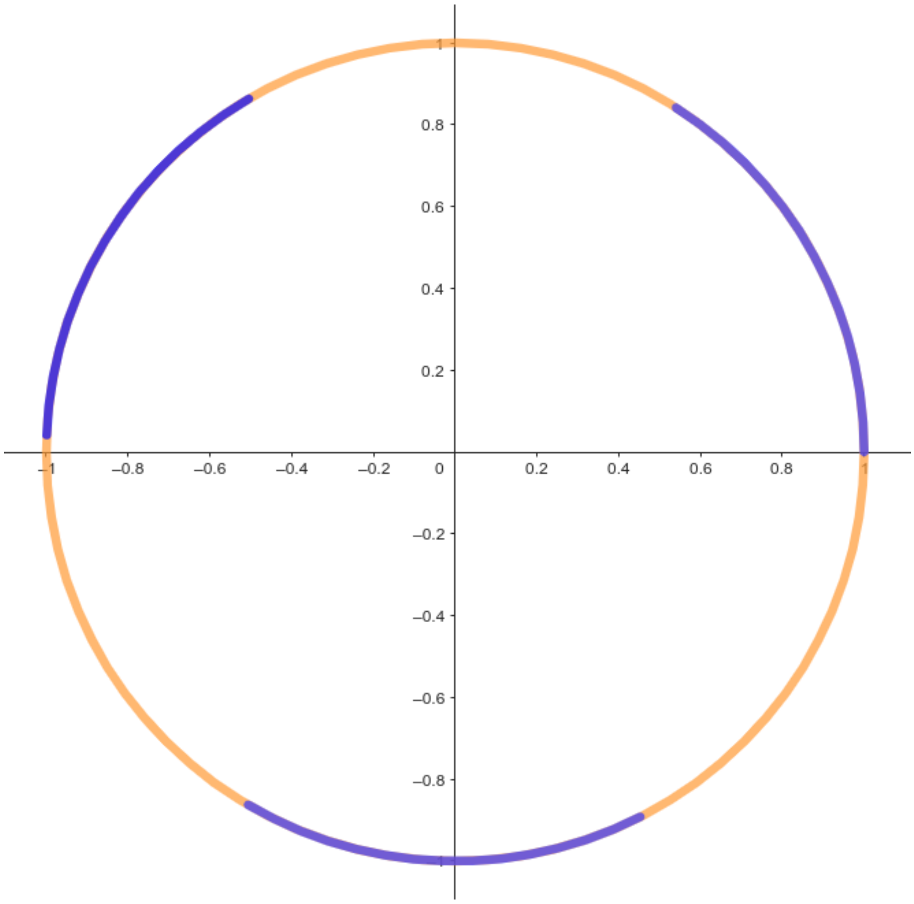

Figure 2: Illustration of the sample case.

Hiitism

Hiitism is a popular art movement in Hiitsburgh. In Hiitism, the paintings only use three colors, Honeydew, Indigo, and Teal. The canvas is divided into a number of rows and columns, and each brushstroke has to color either one column or one row with single color. A shady figure wants to sell you a painting that he claims to be Hiitist. Can you figure out if that's true?

Input

The first line of the input has two integers n and m, describing the height and the width of the painting.

The following n lines have strings of length m, describing each row of the canvas. The colors Honeydew, Indigo, and Teal are denoted by 'H', 'I', and 'T', and an unpainted part of a canvas by '.'.

Output

Output a sequence of brushstrokes that would produce the given painting or state that it is impossible by outputting "Impossible". If such a sequence of brushstrokes exists, first write the number of brushstrokes and then three values for each brushstroke: whether we paint a row ('R') or a column ('C'), its index, and the color of the brushstroke.

Constraints

- $1 \leq n, m \leq 1000$
- You may use at most 10^5 brushstrokes.

Example

Input:

```
3 3
.H.
IHI
TTT
```

Output:

```
3
R 2 I
C 2 H
R 3 T
```

The painting from the example is illustrated below. Unpainted parts of the canvas are colored in a light shade of brown.

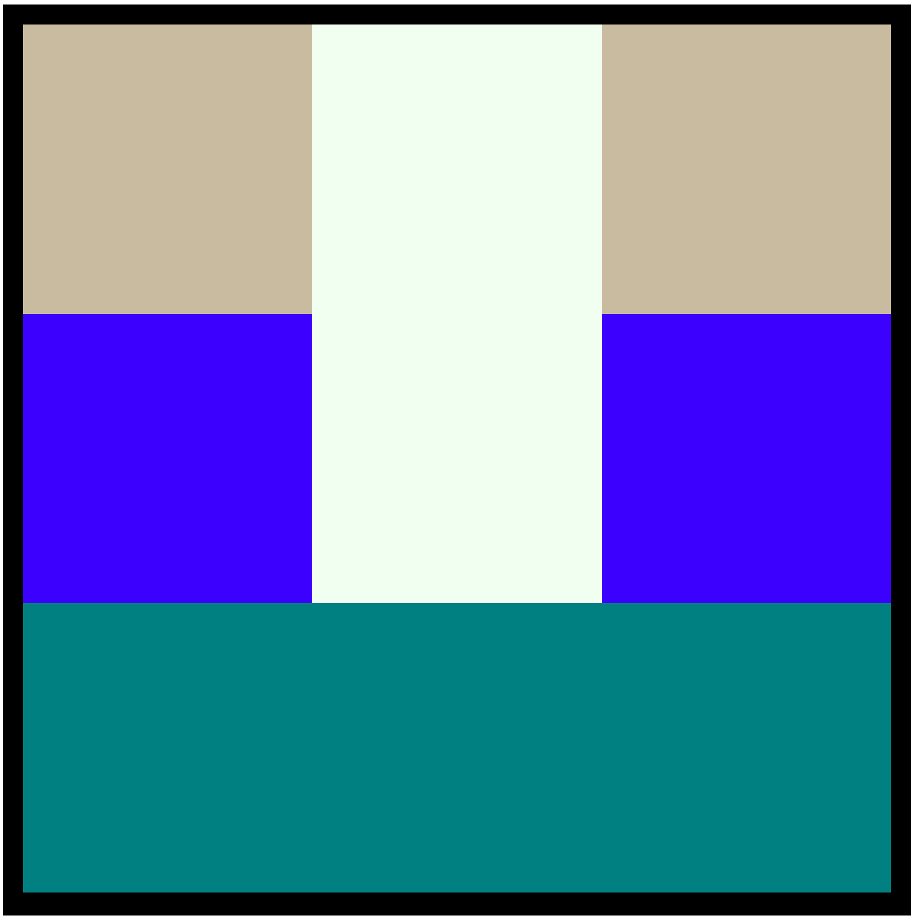

Figure 3: Example of a Hiitist painting.

Inheritance

You and your sister have received inheritance. Unfortunately, you are not on speaking terms with your sister, and you need to divide the inheritance evenly between you two. You may give some of the items to charity if you otherwise cannot split them evenly.

Input

The first line has a single integer n describing the number of items you have inherited.

The second line has n positive integers describing the value of each individual item.

The total value of the items is at most $2^n - 2$.

Output

If a solution exists, output two lines with the first describing the values of the items that you receive and the second line describing the values of items that your sister receives. Neither of you should receive zero items. The unlisted items are given to charity. If no solution exists, output “Impossible”.

Constraints

- $2 \leq n \leq 40$
- $1 \leq \sum_{i=1}^{n} x_i \leq 2^n - 2$

Example

Input:

```
6
3 2 5 1 7 8
```

Output:

```
3 2
5
```

Explanation: If you take items with values 3 and 2, and your sister takes the item with value 5, you both get equally valuable inheritances in total.

Just do it

Your colleague at Hiitsburgh Inversion Innovation Technologies has come up with the following sorting algorithm:

```
// Sort an array. Initialize with low = 0 and high = len(array)
algorithm HiitSort(array, low, high)
    if high - low > 1
        pivot <- median(array[low], array[(low + high) / 2], array[high - 1])

        a <- low
        b <- high - 1
        loop
            while array[a] < pivot      // (*)
                a <- a + 1
            while array[b] > pivot      // (*)
                b <- b - 1
            if a >= b
                break loop
            swap(array[a], array[b])
            a <- a + 1
            b <- b - 1

        call HiitSort(array, low, a)
        call HiitSort(array, a, high)
```

Your colleague claims that their algorithm uses only $O(n \log n)$ comparisons against the pivot in total. You are sceptical of this claim and want to prove them wrong. In particular, you want to find an array of n distinct elements such that the algorithm makes at least $n^2/4$ comparison operations against the pivot.

For the purposes of this exercise, we are only interested in the comparisons against the pivot, that is the comparisons executed on lines 9 and 11 of the code above (marked with `(*)` in the code).

The array is 0-indexed and the division operator rounds down. Function `median` returns the value that is the median of the three arguments; it does not count towards comparisons.

Input

The input contains a single integer n, the number of elements in the array.

Constraints

- $1 \leq n \leq 1000$

Output

The output contains n distinct integers from range $[1, n]$, the array to be sorted.

Example

Input:

```
3
```

Output:

```
1 2 3
```

Explanation: The array takes 7 operations to sort, which is more than $3^2/4 = 2\frac{1}{4}$.

Input:

```
5
```

Output:

```
1 2 4 3 5
```

Explanation: The array takes 17 operations to sort, which is more than $5^2/4 = 4\frac{1}{4}$.

Key cutting

You forgot your key inside your house and are now locked outside. Fortunately, you remember precisely what your key looks like and have key cutting tools for producing a new one. A key can be described as a sequence of n integers describing the depth at which each segment is cut. To construct a new key, you start with a key blank without any cuts. In other words, each segment has depth 0 initially. Then, on a single operation you choose an interval of segments and remove metal within that interval from the key blank until reaching some chosen depth. What is the minimum number of operations you need to perform?

Input

The first line of the input has the number of segments n.

The second line of the input has n nonnegative integers describing the desired depths of each segment.

Output

Output a single integer, the minimum number of operations needed.

Constraints

- $1 \leq n \leq 10^5$
- Each depth is at most 10^9.

Example

Input:

```
5
1 2 1 4 1
```

Output:

```
3
```

Explanation: you can do the following operations:

- choose interval $[1, 5]$, cut to depth 1
- choose interval $[2, 2]$, cut to depth 2
- choose interval $[4, 4]$, cut to depth 4

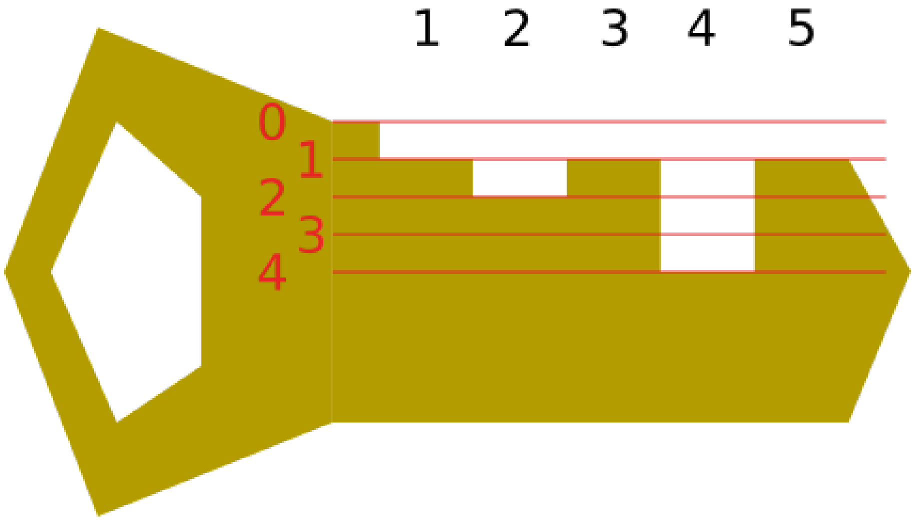

Figure 4: Visualization of the example input

Light rail connections

The city of Hiitsburgh has n light rail stations and m connections between them. Due to increased expenses in other building projects, the city council wants to reduce the maintenance costs of the light rail network by dismantling some of the connections such that at most $3n$ connections remain. Complicating the matter, they also want to ensure that the network remains reliable after dismantling the connections: If any two connections could fail in the initial network and there would still be a route between two stations, then that should still be true after dismantling the connections. Further, any pair of stations with a route between them should still have one afterwards.

Input

The first line of the input has two integers n and m.

The following m lines have two integers v and w, stating that there is a light rail connection between stations v and w.

Output

Output any valid set of light rail connections that can be kept. First, output the number of remaining connections and then the stations for each remaining connection on separate lines.

Constraints

- $1 \leq n \leq 10^3$
- $0 \leq m \leq 10^4$
- There is at most one direct connection between any pair of stations.

Example

Input:

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

Output:

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