icpc | north america contests

The 2023 ICPC
Pacific Northwest Regional Contest

Official Problem Set

2023
2024

nac

icpc

pacific
northwest

JET
BRAINS

icpc global sponsor
programming tools

upsilon pi epsilon
honor society

Pacific Northwest Regional Programming Contest
Division 1

24 February 2024

- The languages supported are C, C++ 20 (with Gnu extensions), Java, Python 3 (with pypy3), and Kotlin.
- Python 2 and C# are not supported this year.
- For all problems, read the input data from standard input and write the results to standard output.
- In general, when there is more than one integer or word on an input line, they will be separated from each other by exactly one space. No input lines will have leading or trailing spaces, and tabs will never appear in any input.
- Submit only a single source file for each problem.
- Python may not have sufficient performance for many of the problems; use it at your discretion.

W
UNIVERSITY of
WASHINGTON
TACOMA

BYU
HAWAII

DeAnza
College

George Fox
UNIVERSITY

UBC
THE
UNIVERSITY OF
BRITISH
COLUMBIA

A: ABC String

Time Limit: 1 second, Memory limit: 2G

You're given a string consisting of the characters `A`, `B`, and `C`. The string contains the same count of `A`, `B`, and `C` characters.

A string is *beautiful* if

- Its length is divisible by 3.
- The string can be split evenly into contiguous substrings of size 3, where each substring has one `A`, one `B`, and one `C`, in any order.

For example: `ABCCBA` is a beautiful string, but `ABCAB` and `CCBAAB` are not beautiful.

Given a string, you want to partition it into subsequences (not necessarily contiguous) such that each subsequence is a beautiful string.

For example, for the string `ABACBCAACCBB`, we can do the following:

```
AB   CA C B
  ACB  A C B
```

This partitions the string into two subsequences `ABCACB` and `ACBACB`, both of which are beautiful strings.

For the given string, find the minimum number of subsequences you can partition it into such that each subsequence is beautiful. It can be proven that there is always at least one such partition for all possible inputs that satisfy the input constraints.

Input

The first line of input contains a string s ($3 \leq |s| \leq 3 \cdot 10^5$). $|s|$ is divisible by 3. s contains an equal number of characters `A`, `B`, and `C`.

Output

Output a single integer, which is the minimum subsequences that s can be partitioned into so each subsequence is a beautiful string.

Sample Input 1	Sample Output 1
`ABACBCAACCBB`	`2`

B: Acceptable Seating Arrangements

Time Limit: 1 second, Memory limit: 2G

Charlie is managing a classroom. The seats in the classroom are arranged in a grid with rows and columns. Each student has a distinct height.

A configuration of students to seats is *acceptable* if the following conditions are met:

- Each student is assigned to exactly one seat.
- The students are seated in increasing order of height from left to right in each row.

The students are initially seated in an acceptable arrangement. Charlie wants to rearrange students into a potentially different acceptable arrangement. To do this, he can swap any two students. However, he wants to ensure that the configuration stays acceptable after each swap.

Help Charlie devise a strategy to move the students from the original arrangement to his preferred arrangement. You don't need to minimize the number of swaps, but you are limited to at most 10^4 swaps.

It can be proven that this is always possible for all possible inputs that satisfy the input constraints.

Input

The first line of input contains two integers r and c ($1 \leq r, c \leq 20$). Charlie's classroom has r rows and c columns of seats.

Each of the next r lines contains c integers h ($1 \leq h \leq r \cdot c$), representing the heights of the students in each row in the original arrangement. The heights are guaranteed to be distinct, and the arrangement is guaranteed to be acceptable.

Each of the next r lines contains c integers h ($1 \leq h \leq r \cdot c$), representing the heights of the students in each row in Charlie's desired arrangement. The heights are guaranteed to be distinct, and the arrangement is guaranteed to be acceptable.

Output

On the first line, output an integer k, which is the number of swaps to perform ($0 \leq k \leq 10^4$).

Then, output the k swaps which change the original arrangement to Charlie's preferred arrangement. On each of the next k subsequent lines, output four integers r_1, c_1, r_2, c_2 ($1 \leq r_1, r_2 \leq r$, $1 \leq c_1, c_2 \leq c$, $(r_1, c_1) \neq (r_2, c_2)$). This represents a swap of the student in row r_1, column c_1 with the student in row r_2, column c_2.

It can be proven that it's always possible to accomplish this in under 10^4 swaps for all possible inputs that satisfy the input constraints. Remember that the arrangement must be acceptable after each swap.

Sample Input 1

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

Sample Output 1

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

C: Candy Factory

Time Limit: 2 seconds, Memory limit: 2G

The International Consortium of Popular Candies (ICPC) is hosting a prestigious candy festival for candy lovers worldwide. The consortium has asked n candy factories to produce candies for the event. Each of the n factories has produced some quantity of a unique type of candy.

Packs of candies will be given to the participants at the festival. A candy pack must consist of exactly k candies of different types. Two candy packs may contain different sets of k candies.

There may be unavoidably some leftover candies given the quantities of candies that the n factories have already produced. The ICPC does not want to waste any of the candies produced, and is willing to create extra packs of candy to ensure this. The ICPC can order any of the n factories to produce additional candies. What is the minimum quantity of additional candies that must be ordered, so that there will be no leftover candies after packing?

Input

The first line of input has two integers n and k ($1 \leq k \leq n \leq 5\,000$).

The next n lines each have a single integer between 1 and 10^9. The integer on the i^{th} line is the quantity of candies that the i^{th} factory has produced.

Output

Output a single integer, the minimum quantity of additional candies that must be ordered.

Sample Input 1

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

Sample Output 1

```
3
```

This page is intentionally left blank.

D: Cramming for Finals

Time Limit: 10 seconds, Memory limit: 2G

It's final exam season and Ashley is heading to her favorite library to cram for finals.

The library has a dedicated floor for studying where there are r rows of c tables evenly spaced. Each table only has room for one student, and some students have already arrived and claimed their favorite tables.

Because the floor is usually very quiet, it is possible to hear sounds from other students who are nearby – for example, frustrated typing on a laptop keyboard or nervous leg shaking. Specifically, if one student is studying at the table in row i_1 and column j_1, and another student is studying at the table in row i_2 and column j_2, it is possible for the two students to hear sounds from each other if and only if $\sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2} \leq d$.

With this, Ashley wants to find an empty table where she can hear as few other students as possible. Compute the minimum number of students that Ashley can hear if she selects her table optimally.

Input

The first line of input has four integers r, c ($2 \leq r, c \leq 10^9$), d ($1 \leq d \leq 2\,500$), and n ($1 \leq n \leq 10^3$ and $n \leq r \cdot c - 1$).

Each of the next n lines contains two integers i ($1 \leq i \leq r$) and j ($1 \leq j \leq c$), indicating that a student is studying at the table at row i and column j. It is guaranteed that no two students are sitting at the same table.

Output

Output a single integer, which is the minimum number of students that Ashley can hear if she selects her table optimally.

Sample Input 1	Sample Output 1

3 2 1 3
1 1
2 2
3 1
	2

This page is intentionally left blank.

E: Eccentric Excursion

Time Limit: 9 seconds, Memory limit: 2G

Eddy is planning a cross-country trip across n different cities. There are $n - 1$ roads connecting the cities. Each road connects two cities and is bidirectional. The roads are laid out such that it is possible to travel between any two cities using only roads.

Eddy wants to plan a trip so that he visits each city exactly once. He may start or end at any city. It might not be possible to visit each city exactly once using only roads. Luckily, Eddy can take a flight between any two cities that aren't directly connected by a road. Eddy would like to take exactly k flights during his trip.

Help Eddy plan his trip.

Input

The first line contains two integers n, k $(0 \le k < n \le 500)$ where n is the number of cities Eddy is visiting and k is the number of flights Eddy would like to take.

The next $n - 1$ lines each contain two integers a, b $(1 \le a < b \le n)$ indicating that there is a road between cities a and b. It is guaranteed it is possible to travel from any city to any other city only using roads.

Output

Output n integers that specify the sequence of cities that Eddy shall visit in order. The sequence must visit each city exactly once and use exactly k flights. If there are multiple possible itineraries, output the lexicographically smallest sequence. If there is no possible itinerary, output -1.

Sample Input 1	Sample Output 1

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

Sample Input 2

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

Sample Output 2

```
-1
```

F: Item Selection

Time Limit: 1 second, Memory limit: 2G

You are browsing a website that lists items for sale. The website has a paging UI that displays a fixed number of items per page, one page at a time.

For example, if there are 55 items and the page displays exactly 20 at a time, then there are 3 pages in total. Items 1 through 20 are on page 1, items 21 through 40 are on page 2, and items 41 through 55 are on page 3.

You may navigate and select items using these UI elements:

- A checkbox for every item on the current page. After you click a checkbox, a selected item becomes unselected, and an unselected item becomes selected. You cannot click a checkbox for an item that is not on the current page.
- A "Select All" button. All unselected items on the current page become selected after you click this button.
- A "Deselect All" button. All selected items on the current page become unselected after you click this button.
- A "Next Page" button. Clicking it navigates to the next page and increments the current page number by one. This button is disabled on the last page.
- A "Previous Page" button. Clicking it navigates to the previous page and decrements the current page number by one. This button is disabled on the first page.

The website has pre-selected some items for you based on its machine learning recommendation algorithm. The recommendation may or may not work for you. You know exactly the items that you want to purchase, which may differ from the pre-selected items. What is the minimum number of checkbox and button clicks required to select exactly the items you actually want?

Input

The first line of input has five integers n, m ($1 \le m \le n \le 10^3$), s ($1 \le s \le \lceil \frac{n}{m} \rceil$), p, q ($0 \le p, q \le n$), where:

- n is the number of items. The items have item numbers from 1 to n.
- m is the fixed number of items displayed per page.
- s is the number of the page currently displayed.
- p is the number of preselected items.
- q is the number of items you want.

Each of the next p lines contains an integer i ($1 \le i \le n$). These are the item numbers of the preselected items. These p items are distinct and are listed in increasing order. It is possible that the website has pre-selected none of the items ($p = 0$), in which case the input has no lines for pre-selected items.

Each of the next q lines contains an integer j ($1 \le j \le n$). These are the item numbers of the items you want to buy. These q items are distinct and are listed in increasing order. It is possible that you want to buy none of the items ($q = 0$), in which case the input has no lines for items you want.

Output

Output a single integer, which is the minimum number of checkbox and button clicks required to select exactly the items you want.

Sample Input 1

```
11 4 1 5 5
1
4
9
10
11
1
3
6
7
8
```

Sample Output 1

```
7
```

G: Matrix Fraud

Time Limit: 2 seconds, Memory limit: 2G

For the purposes of this problem: A matrix is a *binary matrix* if all its entries are 0 or 1. A matrix is a *banded binary matrix* if its rows and columns satisfy the following properties:

1. Every row has at least one 1.
2. Every column has at least one 1.
3. All 1s in each row are contiguous.
4. For row i, if s_i is the leftmost column that has a 1 entry and t_i is the rightmost column that has a 1 entry, then it must satisfy $s_i \geq s_{i-1}$ and $t_i \geq t_{i-1}$ for $i > 1$.

Detecting banded binary matrices is an important method used in various fields like biology, paleontology, and linguistics to unearth clusters in data sets. Unfortunately, a group called the Immoral Cartel of Pure Cozeners (ICPC) has decided to do the unthinkable: manipulate data! The ICPC wishes to present their groundbreaking scientific results, but the scientific community will not take their results seriously because their matrices may not be banded. To have publishable results, they want to toggle some cells such that their data is a banded binary matrix.

The ICPC gives you its raw data, represented as a binary matrix. They want to toggle some cells (meaning, change a 0 to a 1 or a 1 to a 0) so that the resulting matrix is a banded binary matrix, as defined above. What is the fewest number of toggles needed to turn the given matrix into a banded binary matrix?

Input

The first line of input contains two integers r and c ($1 \leq r \times c \leq 2 \cdot 10^5$), which are the dimensions of the matrix. The matrix has r rows and c columns.

Each of the next r lines contains a string of binary digits of length c. This is the matrix.

Output

Output a single integer, which is the minimum number of entries in the matrix to toggle to make the input matrix a binary banded matrix.

Sample Input 1

```
3 4
1100
0101
0011
```

Sample Output 1

```
1
```

H: On-Call Team

Time Limit: 2 seconds, Memory limit: 2G

An IT company has formed an on-call team of software engineers who will manage their backend services and make sure that these services run without interruption. When services go down, for each service that is down the on-call team must dispatch one member who is familiar with that service to take care of its issue. One team member can handle at most one service at a time. The company wants to evaluate the *robustness level* of the on-call team, which is defined as the maximum value k such that any k services that go down simultaneously can be handled by the on-call team.

Input

The first line of input contains two integers n ($1 \leq n \leq 3 \cdot 10^4$) and m ($1 \leq m \leq 20$), where n is the number of engineers and m is the number of backend services.

Each of the next n lines contains a string of binary digits of length m, describing the n software engineers' familiarity with the m services. The j^{th} digit on the i^{th} line is 1 if software engineer i is familiar with service j, and 0 otherwise.

It is guaranteed that for each of the m services there exists at least one software engineer who is familiar with it.

Output

Output a single integer, which is the robustness level of the on-call team.

Sample Input 1

```
4 6
001101
111001
001110
100100
```

Sample Output 1

```
3
```

Sample Input 2

```
3 3
001
001
110
```

Sample Output 2

```
1
```

This page is intentionally left blank.

I: Range Editing

Time Limit: 8 seconds, Memory limit: 2G

You are editing a list of spreadsheet cells. Initially all cells are empty. You can perform two types of operations:

1. Select a consecutive range of cells and change their values to a positive integer of your choice. All these cells get the same value after this operation.
2. Select a consecutive range of cells and delete their values. All these cells become empty after this operation.

Given the final cell values that you would like to have in the spreadsheet, calculate the minimum number of editing operations required to obtain those values.

Input

The first line of input contains a single integer n ($1 \leq n \leq 800$), which is the number of cells you are editing. The cells are number from 1 to n.

Each of the next n lines contains a single integer between 0 and 10^9 inclusive. The integer on the i^{th} line is 0 if cell i should be empty after all operations. Otherwise, it is a positive integer that is the final value of cell i.

Output

Output a single integer, which the minimum number of editing operations required.

Sample Input 1	Sample Output 1
10 0 1 2 3 4 2 1 0 0 1	5

This page is intentionally left blank.

J: Segment Drawing

Time Limit: 8 seconds, Memory limit: 2G

You're given some points, paired with an equal number of line segments. All points are strictly above the x-axis. Each line segment lies completely on the x-axis. None of the line segments share any common points.

You would like to draw some new line segments – one for each given point/line segment pair. The i^{th} drawn segment must connect point i to its corresponding line segment i. The correspondence between point i and line segment i is fixed, you cannot rearrange which point connects to which line segment.

No two drawn segments may strictly intersect, but it is allowed to have one drawn segment touching another drawn segment at an endpoint.

You would like to find the minimum total length of all drawn segments, or determine that it is impossible to draw such segments.

Input

The first line contains a single integer n $(1 \le n \le 10^5)$, which is the number of given point/line segment pairs.

Each of the next n lines contains four integers x, y, l, r $(-10^6 \le x \le 10^6, 0 < y \le 10^6, -10^6 \le l \le r \le 10^6)$. This denotes a point at (x, y) and its corresponding line segment with endpoints $(l, 0)$ and $(r, 0)$. The line segments will be given in order from left to right. No two given line segments will share a common point. It is possible for two points to be at the same location.

Output

Output a single real number, which is the minimum total length of all drawn segments. The answer will be accepted if it is within an absolute or relative error of at most 10^{-6}. If there is no way to draw the new segments without intersecting, output -1.

Sample Input 1

```
3
0 6 -4 -2
-2 1 -1 0
0 4 1 4
```

Sample Output 1

```
11.9995169566
```

Sample Input 2

```
3
0 6 -4 -2
0 4 -1 0
-2 1 1 4
```

Sample Output 2

```
-1
```

Sample Input 3

```
2
0 5 -1000000 -1
0 5 1 1000000
```

Sample Output 3

```
10.1980390272
```

K: Sequence Guessing

Time Limit: 10 seconds, Memory limit: 2G

You are tasked with creating a secret sequence of integers that is difficult to guess.

The sequence is subject to the following constraints:

- The first number in the sequence must be 0.
- The last number in the sequence must be $100\,000$.
- Each number in the sequence must be either 1 or 2 greater than the one preceding it.

At first, all you need to reveal is the length of the sequence.

Then, an adversary will guess the numbers in the sequence one at a time.

- If the guessed number is in your sequence, you must reveal exactly where in the sequence it appears.
- If the guessed number is not in your sequence, you must simply reveal that it is not in the sequence. This is considered a "miss".

Note that because you are not forced to write down the sequence in advance, you can "cheat" by changing the sequence you have in mind, so long as it does not contradict the information you have revealed so far. It turns out that under these conditions, you can always force the adversary to get $33\,333$ misses before they can guess every number in your sequence. Your job is to write a program that does so.

Interaction

This is an interactive problem.

Your program should begin by printing k $(2 \leq k \leq 100\,001)$, the length of your sequence, as a single integer on a single line. After this, you will receive one integer x on a single line. This integer is guaranteed to be between -1 and $100\,000$ inclusive.

- If $x = -1$, the adversary has given up; your program should print all k integers in your sequence in order, one line per integer, and then exit. The adversary is guaranteed to give this input after it has gotten $33\,333$ misses, though it may do so earlier. After printing all k integers, your program should exit. If you print a valid sequence consistent with your previous responses, your submission will be considered correct for this test case.

- If x is not in your sequence, print -1 on a single line.
- If x is in your sequence, print a single integer i on a single line, such that x is the ith (1-indexed) number in the sequence. If you have printed every integer from 1 to k, your program should now exit, and your submission will receive a Wrong Answer verdict.

Do not forget to flush the output after every integer you print.

After this, if your program has not yet exited, the process will repeat, with you receiving another single integer. The adversary is guaranteed to never repeat integers.

The adversary may employ different guessing strategies on different runs.

Sample Interaction 1

Read	Write
	50001
1	
	-1
0	
	1
-1	
	0 2 <omitted 49997 lines for brevity> 99998 100000

L: Training

Time Limit: 2 seconds, Memory limit: 2G

Ashley is training for a programming contest on Brandon's Online Judge. Brandon's Online Judge has a new feature which allows Ashley's coach, Tom, to load a list of problems for Ashley.

Tom has curated some problems for Ashley to work on. Each problem has two integers as a lower skill bound and an upper skill bound. Each programmer has an integer skill level. If someone with a skill level between the lower and upper bounds of a problem (inclusive), and they solve that problem, then his/her skill level goes up by 1.

Ashley will train on Tom's curated list of problems as follows – she will look at the first problem on the list and either solve it or skip it. She will repeat this for every problem on the list in the order Tom loaded the problems. Once she has skipped a problem, she can never go back to it.

Compute the maximum skill level Ashley can have if she chooses to solve or skip problems optimally.

Input

The first line contains two integers n and s $(1 \leq n \leq 10^5, 0 \leq s \leq 10^9)$, where n is the number of problems Tom has curated for Ashley, and s is Ashley's current skill level.

Each of the next n lines contains two integers l and r $(0 \leq l \leq r \leq 2 \cdot 10^9)$. These are the lower (l) and upper (r) skill bounds on each of Tom's problems, in the order that Tom loaded them.

Output

Output a single integer, which is the maximum skill level Ashley can attain.

Sample Input 1	Sample Output 1

3 0
0 2
0 0
1 1
	2

This page is intentionally left blank.

M: Triple Sevens

Time Limit: 1 second, Memory limit: 2G

Eddy is overseeing construction of some new slot machines. His slot machines consist of three wheels, each of which can show one of several different digits. When activated, each wheel shows a random digit. The goal is for all wheels to be able to show the digit 7.

A slot machine is good if each wheel is capable of showing the digit 7, and bad otherwise. For a given slot machine, determine if it is good or bad.

Input

The first line contains a single integer n $(1 \leq n \leq 10)$, which is the number of different digits on each wheel. Each wheel has the same number of digits.

Each of the next three lines contains n distinct digits (in the range from 0 to 9). These are the digits on each of the three wheels.

Output

Output a single integer, which is 777 if the slot machine is good, 0 otherwise.

Sample Input 1

```
2
0 7
7 3
7 0
```

Sample Output 1

```
777
```

Sample Input 2

```
2
0 7
7 3
3 9
```

Sample Output 2

```
0
```

This page is intentionally left blank.