icpc international collegiate programming contest

ICPC North America Contests

North Central NA Regional Contest

Official Problem Set

nac

2022
2023

icpc.foundation

icpc global sponsor
programming tools

upsilon pi epsilon
honor society

Problems

North Central North America Regional Programming Contest

25 February 2023

- The languages supported are C, C++ 17 (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 some of the problems; use it at your discretion.

This page is intentionally left blank.

Problem A
Repetitive Song
Time limit: 1 second

Your younger sibling is obsessed with a fairly repetitive song. They claim that it is not repetitive, so you decided to prove your point by finding the longest (in terms of the number of words) subsequence of the song that your sibling cannot definitively identify within the full song lyrics.

More formally, a length-ℓ subsequence of a song with n words is a tuple of ℓ integers $1 \le s_1 < s_2 < \cdots < s_\ell \le n$ identifying the words in the subsequence. The subsequence is non-identifiable if there exists a different length-ℓ subsequence $1 \le t_1 < t_2 < \cdots < t_L \le n$ (with $s_i \neq t_i$ for at least one index i) where word s_1 in the song is identical to word t_1, word s_2 is identical to word t_2, etc.

Given the lyrics for a song, print the length of the longest non-identifiable subsequence.

Input

The first line of input contains a single integer n ($1 \le n \le 10^5$) specifying the number of words in the song lyrics.

Each of the next n lines contains one word of the song lyrics, given in order. Each word consists of up to 20 uppercase (`A`–`Z`) and lowercase (`a`–`z`) letters. The same word may appear on multiple lines. If two words do not match exactly (including the use of upper and lower case) then they are considered to be different words.

Output

Output a single integer ℓ, the number of words in the longest non-identifiable song subsequence. If all of the song's subsequences are identifiable, print 0. When determining if a subsequence is identifiable, treat two words in the lyrics as identical if each of their corresponding characters are identical (in other words, case *does* matter).

Sample Input 1	Sample Output 1
10 bow bow chick chicka chicka bow bow chick chicka chicka	9

Sample Input 2	Sample Output 2
31 head shoulders knees and toes knees and toes head shoulders knees and toes knees and toes eyes and ears and mouth and nose head shoulders knees and toes knees and toes	29

This page is intentionally left blank.

Problem B
Exponent Exchange
Time limit: 1 second

Alice and Bob are playing a cooperative game. They hold b^p dollars between them, for given integers b and p. Alice initially holds x dollars, and Bob holds $b^p - x$ dollars. Alice and Bob want to consolidate their money, so one person holds all the money.

In each transaction, one player can choose to send the other player b^y dollars, for some integer y with $0 \leq y < p$. But each player wants to initiate as few transactions as possible. They are willing to cooperate such that the player that initiates the most transactions (the busiest player), initiates as few as possible.

Alice and Bob want to know the fewest number of transactions that the busiest player needs to initiate to complete the transfer.

Input

The first line of input contains two integers b ($2 \leq b \leq 100$) and p ($2 \leq p \leq 1000$), where b is the base, and p is the number of digits.

The next line contains p integers $x_{p-1}, x_{p-2}, \ldots, x_0$, separated by spaces, with $0 \leq x_i < b$ and $0 < x_{p-1}$. These are the base-b digits of the value of x, with the most significant digit first. Specifically, $x = \sum_{0 \leq i < p} b^i x_i$. Note that they are given in order from the highest power to the lowest. For example, in the sample, `4 2 7 8 6` with $b = 10$ represents the base 10 number $42{,}786$.

Output

Output a single integer, which is the minimum number of transactions the busiest player must initiate to transfer all the money to either Alice or Bob.

Sample Input 1	Sample Output 1
`10 5` `4 2 7 8 6`	`7`

Problem C
Chocolate Chip Fabrication
Time limit: 1 second

You are making a chocolate chip cookie using a machine that has a rectangular pan composed of unit squares. You have determined the shape of your cookie, which occupies some squares in that area. Each square of your cookie must be chocolate chipified.

To make the cookie you will repeatedly perform the following two steps:

1. You place cookie dough in some unit squares.
2. You expose the cookie dough to a shallow chocolate chip solution. Any cookie dough square that does not have all four adjacent squares (up, down, left, right) filled with cookie dough becomes chocolate chipified. Note that any cookie dough in a square on the boundary of the pan always gets chipified.

The following example shows how to make a cookie of the shape shown on the left (s):

```
(s)      (a1)     (a2)     (b1)     (b2)
-X-X-    -D-D-    -C-C-    -C-C-    -C-C-
XXXXX    -D-D-    -C-C-    DCDCD    CCCCC
XXXXX    -DDD-    -CCC-    DCCCD    CCCCC
-XXX-    --D--    --C--    -DCD-    -CCC-
--X--    -----    -----    --D--    --C--
```

First you place cookie dough in 8 squares (a1). All squares become chipified after the first solution exposure (a2). You place cookie dough in 8 more squares (b1). The second exposure makes every square chipified and completes the cookie (b2).

Your chocolate chip solution is expensive, so you want to ensure that you perform the exposure as few times as possible. Given a cookie shape, determine the minimum number of chocolate chip solution exposures required to make the cookie.

Input

The first line of input contains two integers n and m ($1 \leq n, m \leq 1{,}000$), indicating the pan has n rows and m columns of unit squares.

Each of the next n lines contains a string of exactly m characters, where each character is either "X", representing a square occupied by your cookie, or "-", representing an empty square.

The shape of your cookie occupies at least one square. Note that the shape may consist of multiple pieces that are disconnected.

Output

Output the minimum number of chocolate chip solution exposures required to make your cookie.

Sample Input 1	Sample Output 1
5 5 -X-X- XXXXX XXXXX -XXX- --X--	2

Sample Input 2	Sample Output 2
4 5 --XXX --X-X X-XXX XX---	1

Sample Input 3	Sample Output 3
5 5 XXXXX XXXXX XXXXX XXXXX XXXXX	3

Problem D
Advertising ICPC
Time limit: 1 second

You're making a flag to try to advertise ICPC! The flag takes the form of a grid that is already filled with some "C", "I", and "P" letters. A flag is *advertising ICPC* if there exists at least one 2×2 subgrid that looks exactly like the following:

```
IC
PC
```

The flag cannot be rotated or reflected. Every square in the grid must be filled with either a "C", "I", or "P". Count the number of ways to fill the unfilled locations on the flag such that the flag is advertising ICPC.

Input

The first line contains two integers, n and m ($2 \leq n, m \leq 8$), where n is the number of rows and m is the number of columns in the grid.

The next n lines each contains a string of length m. Each character in the string is either a "C", "I", "P", or "?". A "?" means that that location is not yet filled with a letter.

These n lines form the grid that represents the flag.

Output

Output a single integer, which is the number of ways to fill the flag such that it is advertising ICPC, modulo $998{,}244{,}353$.

Sample Input 1	Sample Output 1
3 3 ??? ?I? ???	243

Sample Input 2	Sample Output 2
2 2 IC PC	1

This page is intentionally left blank.

Problem E
Sun and Moon
Time limit: 1 second

You recently missed an eclipse and are waiting for the next one! To see any eclipse from your home, the sun and the moon must be in alignment at specific positions. You know how many years ago the sun was in the right position, and how many years it takes for it to get back to that position. You know the same for the moon. When will you see the next eclipse?

Input

The input consists of two lines.

The first line contains two integers, d_s and y_s $(0 \leq d_s < y_s \leq 50)$, where d_s is how many years ago the sun was in the right position, and y_s is how many years it takes for the sun to be back in that position.

The second line contains two integers, d_m and y_m $(0 \leq d_m < y_m \leq 50)$, where d_m is how many years ago the moon was in the right position, and y_m is how many years it takes for the moon to be back in that position.

Output

Output a single integer, the number of years until the next eclipse. The data will be set in such a way that there is not an eclipse happening right now and there will be an eclipse within the next $5{,}000$ years.

Sample Input 1	Sample Output 1
3 10 1 2	7

This page is intentionally left blank.

Problem F
Streets Ahead
Time limit: 2 seconds

International Connecting Passage Causeway is a long, rutted two-way country road crossed by streets at different points.

There are many drivers, and each will drive along the country road starting at some intersection and ending at some other intersection. For each driver, how many intersections will they drive through?

Input

The first line contains two integers, n $(2 \leq n \leq 10^5)$ and q $(1 \leq q \leq 10^5)$, where n is the number of cross streets and q is the number of drivers.

Each of the next n lines contains a string of at most ten lowercase letters representing the name of one of the streets that crosses the country road. All street names are unique. Driving along the country road in some direction, one sees these streets in exactly the order provided.

Each of the next q lines contains two strings of at most ten lowercase letters representing the start and end intersection for each driver. Queries will be between distinct streets.

Output

Output q lines, the ith line containing the number of intersections that the ith driver drives through.

Sample Input 1	Sample Output 1
3 3 first second third first second third first second third	0 1 0

This page is intentionally left blank.

Problem G
Restaurant Opening
Time limit: 1 second

Image by wirestock on Freepik

It is said that the three most important factors for determining whether or not a business will be successful are location, location, and location. The Incredible Cooks Preparing Cuisine are opening a new restaurant in the International City Promoting Cooking, and they have hired you to find the optimal location for their restaurant.

You decide to model the city as a grid, with each grid square having a specified number of people living in it. The distance between two grid squares (r_1, c_1) and (r_2, c_2) is $|r_1 - r_2| + |c_1 - c_2|$. In order to visit the restaurant, each potential customer would incur a cost equal to the minimum distance from the grid square in which they live to the grid square containing the proposed location of the restaurant. The total cost for a given restaurant location is defined as the sum of the costs of everyone living in the city to visit the restaurant.

Given the current city layout, compute the minimum total cost if the Incredible Cooks Preparing Cuisine select their next restaurant location optimally.

Input

The first line of input contains two integers, n and m ($1 < n, m \leq 50$), where n is the number of rows in the city grid and m is the number of columns.

Each of the next n lines contains m integers g_{ij} ($0 \leq g_{ij} \leq 50$), which specifies the number of people living in the grid square at row i, column j.

Output

Output a single integer, which is the total cost if the restaurant is selected optimally.

Sample Input 1	Sample Output 1
2 2 1 2 3 4	7

Sample Input 2	Sample Output 2
1 10 3 49 4 31 10 31 50 24 10 42	591

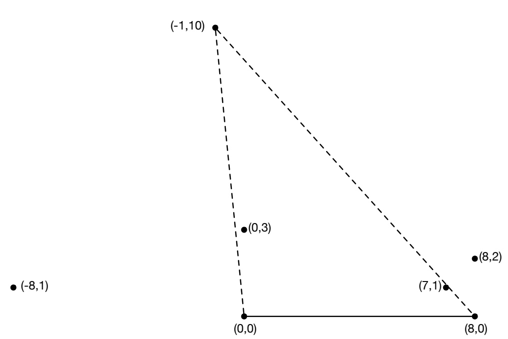
Problem H
Triangle Containment
Time limit: 3 seconds

You recently discovered there is treasure buried on your farm land. A **lot** of treasure! You quickly decide to put a fence around the land.

Alas, you have but a single fence post! You will have to drive to town to get more fencing material. But you can't just leave the land as open as it is, so you decide to create a makeshift fence to protect some of the treasure while you are gone. You will place the post in the ground and run some wire in a straight line between two sides of your barn wall and the fence post to section off a triangular area. Also, the ground is very hard: only places that were dug up to bury a treasure are soft enough for you to quickly place the fence post.

To figure out the best option, you first calculate the following. For each of the treasures in your field, if you were to place the fence post at that treasure and complete the fence as described, then what is the total value of all treasures that would be enclosed by the fence? Note that the treasure under the post you place is not considered enclosed by the fence (it might not be safe since someone could dig around the post).

Sample Input 1 is illustrated below. The triangle that includes the point $(-1, 10)$ encloses exactly two other treasure points which have total value $4 + 8 = 12$.



Input

The first line of input contains two integers n ($1 \leq n \leq 10^5$) and x ($1 \leq x \leq 10^9$), where n is the number of treasure points and x fixes the two corners of the barn wall at locations $(0, 0)$ and $(x, 0)$.

Each of the next n lines contains three integers x, y, and v ($-10^9 \leq x \leq 10^9$, $1 \leq y \leq 10^9$, and $1 \leq v \leq 10^9$) giving the location (x, y) and value v of one of the treasure points. All of these points are distinct. It is also guaranteed that for each point, the triangle formed with the barn wall will not contain any other treasure point on its boundary.

Output

Output n lines, one for each treasure point in the order of the input. For each point output a single integer, which is the total value of all points in the interior of the triangle that point forms with the barn wall. Note that the value of the point itself should be excluded from this sum.

Sample Input 1

```
5 8
-8 1 1
-1 10 2
0 3 4
7 1 8
8 2 16
```

Sample Output 1

```
0
12
0
0
8
```

Sample Input 2

```
6 6
0 1 1
2 3 10
2 5 100
3 1 1000
3 5 10000
4 5 100000
```

Sample Output 2

```
0
1000
1010
0
1010
1000
```

Problem I
Digits of Unity
Time limit: 6 seconds

At the beginning of the school year, the students in the International College of Paper Cutters (ICPC) choose their student IDs. The students can choose any positive integer less than or equal to some maximum number for their IDs, but no two students can choose the same student ID.

After some deliberation among the ranks, the students decided they wanted to find some common ground between all their IDs. In particular, they want to choose their IDs such that the bitwise AND of all of their student IDs has at least some minimum number of 1-bits. The students of the ICPC are asking you to write a program to compute the number of ways to do this. Two assignments are different if there is at least one student that has a different student ID in each assignment.

The bitwise AND of two integers a and b is an integer c whose binary representation is as follows: the ith bit of c is 1 if and only if the ith bits of both a and b are 1. C, C++, Java, and Python all support computing the bitwise AND of two integers using the `&` operator.

This definition generalizes to sets of numbers. The bitwise AND of a set of integers S is an integer c whose binary representation is as follows: the ith bit of c is 1 if and only if the ith bit of each element of S is 1.

Input

The single line of input contains three integers n ($1 \leq n \leq 5 \times 10^5$), k ($1 \leq k \leq 5 \times 10^5$), and m ($n \leq m \leq 5 \times 10^6$), where n is the number of students, k is the required minimum number of common bits, and m is the maximum number any student ID could be.

Output

Output a single integer, which is the number of ways to choose n distinct student IDs from the range $[1, m]$ such that the number of 1-bits in the bitwise AND of the student IDs has at least k 1-bits. Since the answer may be large, output it modulo $998{,}244{,}353$.

Sample Explanation

There are 2 students, they want the bitwise AND of their student IDs to have at least 2 1-bits and the maximum allowed student ID is 10. The valid ID assignments are $\{(3, 7), (5, 7), (6, 7), (7, 3), (7, 5), (7, 6)\}$.

Sample Input 1	Sample Output 1
`2 2 10`	`6`

Sample Input 2	Sample Output 2
`3 4 14`	`0`

Sample Input 3	Sample Output 3
`2 1 100000`	`910073387`

Problem J
Bog of Eternal Stench
Time limit: 2 seconds

You are trying to reach the center of a Labyrinth, which means you must cross the Bog of Eternal Stench. Legend says that if you put so much as one toe in the Bog you will smell bad... forever. You would of course prefer to make it through the Bog with the minimum level of stench possible.

Luckily, you have previously determined that the stench does eventually wear off, and that some areas of the Bog transfer more stench to you than others. There are also small islands where you can rest, without any effect on your stench level. The first island is the starting location of your journey through the Bog of Eternal Stench, and the last is your destination at the other side.

From each island, established bridges allow you to travel to other islands, but these bridges can only be used in one direction. Because this is the Bog of Eternal Stench, traveling along most bridges will increase your overall stench by a specific amount. However, some bridges are quite pleasant, and will decrease your overall stench as you travel along them. But there is a catch—your stench level can never drop below 0. (A bridge that would decrease your stench level below 0 sets it to 0 instead).

You have carefully mapped out all of the islands and bridges, and measured the amount each bridge will increase or decrease your stench. As a result, it may be possible to traverse the Bog of Eternal Stench and emerge with no stench at all!

Your top priority is reaching the destination island with minimum stench; you are willing to take a circuitous path that visits some islands multiple times if doing so achieves this goal. Your path must end at the destination island, but you don't have to leave the Bog immediately the first time your reach your destination, if taking an additional detour and returning to the island later would decrease your final stench value.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 2{,}000$), where n is the number of islands and m is the number of direct bridges.

Each of the next m lines contains 3 integers u, v, and s ($1 \le u, v \le n$, $-10^9 \le s \le 10^9$), indicating that there is a direct bridge from island u to island v that changes your overall stench level by s. It is guaranteed that $u \ne v$, and that there is at most one direct bridge from u to v (but there can also be another direct bridge from v to u).

You may assume that it is possible to reach island n (your destination) from island 1 (your starting location).

Output

Output a single integer, which is the minimum stench level you can exit the Bog with, assuming you begin with 0 stench.

Sample Input 1	Sample Output 1
4 4 1 2 5 1 3 -2 2 4 1 3 4 10	6

Sample Input 2	Sample Output 2
5 5 1 2 1000 2 3 -3 3 4 1 4 2 0 2 5 2	3

Sample Input 3	Sample Output 3
3 3 1 3 -10 3 2 2 2 3 -1	0