icpc international collegiate programming contest

ICPC North America Contests

Pacific Northwest Regional Contest

Official Problem Set

nac

2022
2023

icpc.foundation

icpc global sponsor
programming tools

upsilon pi epsilon
honor society

Pacific Northwest Regional Programming Contest
Division 1
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 many of the problems; use it at your discretion.

Problem A
Three Dice
Time Limit: 5 sec

Given a list of three-letter words, generate one possible set of three, six-sided dice such that each word can be formed by the top faces of some arrangement of the three dice. You must distribute 18 distinct letters across the 18 total faces of the dice. There may be multiple possible sets of dice that satisfy the requirement; any correct set will be accepted.

Input

The first line of input contains an integer n $(1 \le n \le 1{,}000)$, which is the number of words.

Each of the next n lines contains one three-letter word made up only of lowercase letters (`a`–`z`). There may be duplicate words in the list, and the words might contain identical letters.

Output

Output a single line. If there exists a set of dice that can form all of the words, output any such set. Output the set of dice as one line with three space-separated strings, each consisting of six lowercase letters. If no such set of dice can be formed, output a single line with the number 0.

Sample Input 1	Sample Output 1
3 lad fin sly	zounds plight fakery

Sample Input 2	Sample Output 2
1 dad	0

Sample Input 3	Sample Output 3
11 aft cog far irk kit yes tau rag own uke via	vortex whacky fusing

Problem B
Alchemy

Time Limit: 1 sec

You just finished day one of your alchemy class! For your alchemy homework, you have been given a string of lowercase letters and wish to make it a palindrome. You're only a beginner at alchemy though, so your powers are limited. In a single operation, you may choose exactly two adjacent letters and change each of them into a different lowercase letter. The resulting characters may be the same as or different from one another, so long as they were both changed by the operation.

Formally, if the string before the operation is s and you chose to change characters s_i and s_{i+1} to produce string t, then $s_i \neq t_i$ and $s_{i+1} \neq t_{i+1}$ must be true, but $t_i = t_{i+1}$ is permitted.

Compute the minimum number of operations needed to make the string a palindrome.

Input

The single line of input contains a string of n $(2 \leq n \leq 100)$ lowercase letters, the string you are converting into a palindrome.

Output

Output a single integer, which is the minimum number of operations needed to make the string a palindrome.

Sample Input 1	Sample Output 1
ioi	0

Sample Input 2	Sample Output 2
noi	1

Sample Input 3	Sample Output 3
ctsc	1

Sample Input 4

```
fool
```

Sample Output 4

```
2
```

Sample Input 5

```
vetted
```

Sample Output 5

```
2
```

Problem C
Champernowne Count

Time Limit: 1 sec

The nth Champernowne word is obtained by writing down the first n positive integers and concatenating them together. For example, the 10th Champernowne word is "12345678910".

Given two positive integers n and k, count how many of the first n Champernowne words are divisible by k.

Input

The single line of input contains two integers, n $(1 \leq n \leq 10^5)$ and k $(1 \leq k \leq 10^9)$.

Output

Output a single integer, which is a count of the first n Champernowne words divisible by k.

Sample Input 1	Sample Output 1
4 2	2

Sample Input 2	Sample Output 2
100 7	14

Sample Input 3	Sample Output 3
314 159	4

Sample Input 4	Sample Output 4
100000 999809848	1

This page is intentionally left blank.

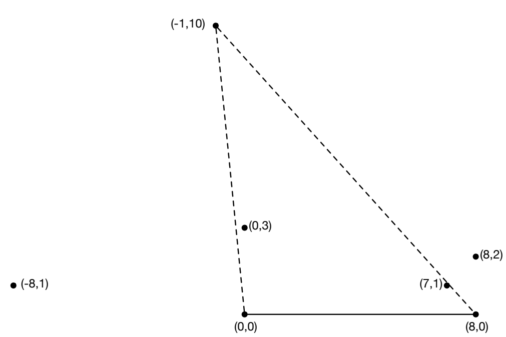
Problem D
Triangle Containment
Time Limit: 4 sec

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 ($1 \le n \le 10^5$) and x ($1 \le x \le 10^9$), 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 ($-10^9 \le x \le 10^9$, $1 \le y \le 10^9$, and $1 \le v \le 10^9$) 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 E
Color Tubes

Time Limit: 1 sec

There is a new puzzle generating buzz on social media—Color Tubes. The rules are relatively simple: you are given $n + 1$ tubes filled with $3n$ colored balls. Each tube can hold at most 3 balls, and each color appears on exactly 3 balls (so there are n colors).

Using a series of moves, you are supposed to reach a Color Tubes state—each tube should either hold balls of a single color or it should be empty.

The only move allowed is to take the top ball from one tube and place it into a different tube that has room for it (i.e. holds at most two balls before the move).

You want to write a program to solve this puzzle for you. Initially, you are not interested in an optimal solution, but you want your program to be good enough to solve any puzzle configuration using at most $20n$ moves.

Input

The first line of input contains a single integer n ($1 \leq n \leq 1{,}000$), which is the number of colors.

Each of the next $n + 1$ lines contains three integers b, m and t ($0 \leq b, m, t \leq n$), which are the descriptions of each tube, where b is the color of the ball on the bottom, m is the color of the ball in the middle, and t is the color of the ball on the top.

The tubes are numbered from 1 to $n + 1$ and are listed in order. The colors are numbered from 1 to n. The number 0 describes an empty space. It is guaranteed that no empty space will be below a colored ball.

Output

On the first line output an integer m, the number of moves that your program will use to solve the puzzle. Remember, m has to be at most $20n$.

On the next m lines, output two space-separated integers u and v that describe a move ($1 \leq u, v \leq n + 1$). In each move, you are taking the uppermost ball out of tube u and placing it in tube v, where it will fall until it hits the uppermost ball already in that tube, or the bottom of the tube if the tube is empty.

Your solution will be deemed incorrect if it uses more than $20n$ moves, or any of the moves are not allowed, or the final configuration is not a Color Tubes state.

Sample Input 1	Sample Output 1
3 2 2 0 1 3 1 3 1 2 3 0 0	6 3 1 2 3 2 4 3 2 3 2 3 4

Sample Input 2	Sample Output 2
1 0 0 0 1 1 1	0

Problem F
Food Processor

Time Limit: 2 sec

You have a food processor with a variety of blades that can be attached to it, as well as some food you would like to process into smaller pieces.

The food processor can have one blade attached at any time. Each blade processes food by reducing its average piece size at a particular exponential rate, but it also has a maximum average piece size requirement; if the average piece size of the food is too big for the blade, the food processor will get stuck. Given a starting average food piece size, a target average piece size, and a set of blades for your food processor, determine the minimum amount of processing time needed to process your food into the target average piece size.

Note that we only care about the time spent actively processing food; we do not track time spent switching out blades or loading/unloading the food processor.

Input

The first line of input contains three integers s, t, and n ($1 \leq t < s \leq 10^6, 1 \leq n \leq 10^5$), where s is the starting average piece size, t is the target average piece size, and n is the number of blades.

Each of the next n lines contains two integers m and h ($1 \leq m, h \leq 10^6$). These are the blades, where m is the maximum average piece size of the blade and h is the number of seconds the blade needs to halve the average piece size.

Output

Output a single number, which is the minimum amount of time in seconds needed to process the food to the target average piece size. If it is not possible to reach the target, output -1. Your answer should have a *relative* error of at most 10^{-5}.

Sample Input 1	Sample Output 1
10 1 2 10 10 4 5	23.219281

Sample Input 2

```
10000 9999 1
10000 1
```

Sample Output 2

```
1.4427671804501932E-4
```

Problem G
Digits of Unity
Time Limit: 5 sec

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 H
Branch Manager
Time Limit: 5 sec

You are managing a transportation network of one-way roads between cities. People travel through the transportation network one by one in order all starting from the same city, and each person waits for the person before them to stop moving before starting. The people follow a simple algorithm until they reach their destination: they will look at all the outgoing roads from the current city, and choose the one that leads to the city with the smallest label. A person will stop when they either reach their destination, or reach a city with no outgoing roads. If at any point someone fails to reach their destination, the rest of the people still waiting in line will leave.

Before each person enters the transportation network, you can permanently close down any subset of roads to guarantee they reach their destination. The roads that you choose to close down will not be available for future people.

There are n cities, labeled from 1 to n. There are $n - 1$ directed roads, and each road will always be from a lower labeled city to a higher labeled one. The network will form a rooted tree with city 1 as the root. There are m people that want to travel through the network. Each person starts from city 1, and has a specific destination city d in mind. These people will line up in the given order. What is the maximum number of people you can route correctly to their destination if you close roads optimally?

Input

The first line of input contains two integers n and m ($2 \leq n, m \leq 2 \times 10^5$), where n is the number of cities and m is the number of people.

Each of the next $n - 1$ lines contains two integers a and b ($1 \leq a < b \leq n$), denoting a directed road from city a to b. These roads will describe a rooted tree with city 1 as the root.

Each of the next m lines contains a single integer d ($2 \leq d \leq n$), denoting the destination city of the next person in line.

Output

Output a single integer, which is the maximum number of people you can route to the correct destination.

Sample Input 1

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

Sample Output 1

```
5
```

Sample Input 2

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

Sample Output 2

```
1
```

Problem I
Counting Satellites

Time Limit: 1 sec

Nick likes satellites. He likes them so much that he looks for them everywhere. One day he found a string of letters and counted a lot of instances of the word "SATELLITE" among all subsequences of the string. However the next day he forgot this string. Can you help him construct such a string?

String s is a subsequence of string t if and only if it is possible to delete some (possibly zero) characters from t to get s. Two subsequences are considered different if some character at a given position in t is deleted in one subsequence but not the other.

Input

The single line of input contains a single integer k ($1 \leq k \leq 10^{18}$), which is the number of instances of the word "SATELLITE" in the string Nick forgot.

Output

Output a string of at most $5{,}000$ uppercase letters. The string must have exactly k instances of the word "SATELLITE" among all its subsequences. It can be proven that under the given constraints a solution always exists. Note that the length of the string does *not* have to be minimized.

Sample Input 1	Sample Output 1
1	SATELLITE

Sample Input 2	Sample Output 2
2	NICKLIKESSATELLITES

Sample Input 3	Sample Output 3
3	SSSATELLITE

Sample Input 4	Sample Output 4
19	SATELLITESATELLITE

This page is intentionally left blank.

Problem J
Sun and Moon

Time Limit: 1 sec

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 K
Advertising ICPC
Time Limit: 1 sec

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

Problem L
Exponent Exchange

Time Limit: 1 sec

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.

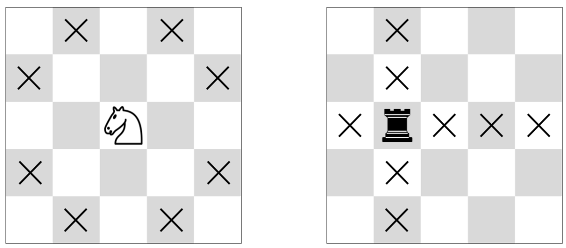
Sample Input 1	Sample Output 1
10 5 4 2 7 8 6	7

This page is intentionally left blank.

Problem M
Lone Knight

Time Limit: 7 sec

In the game of chess, a knight moves as shown in the picture below; each move is one square horizontally and two squares vertically or two squares horizontally and one square vertically. A rook can move any number of squares horizontally or vertically, but not both in the same move. If a square can be reached by a rook in one move, that square is said to be attacked by the rook.



Consider an infinite chess board, with squares that can be indexed by integer coordinates. There is a white knight on the board on a square, and it wants to go to another square. However, there are also a number of black rooks on the board. The knight can make as many moves as it needs to get to its target square, but it cannot stop on a square that is attacked by or occupied by a rook. The rooks don't move.

Can the white knight reach its target square? You are to answer that question many times!

Input

The first line of input contains two integers n and q ($1 \leq n, q \leq 1{,}000$), where n is the number of black rooks and q is the number of queries.

Each of the next n lines contains two integers x and y. This indicates that there is a black rook at (x, y). No two rooks share the same square.

Each of the next q lines contains four integers x_s, y_s, x_t and y_t. This is a query, where the white knight starts at square (x_s, y_s) and wants to move to square (x_t, y_t).

All square coordinates in the input are no larger than 10^9 in absolute value. It is guaranteed that in every query the knight's initial and target squares are not attacked by or occupied by any rook, and the target square is not the same as the initial square.

Output

For each query, output on a single line `1` if the knight can reach the target square, or `0` otherwise.

Sample Input 1

```
6 6
10 14
1 0
0 1
4 9
9 13
5 9
2 2 3 4
2 2 2 4
2 2 6 4
2 2 2 10
7 11 6 2
6 2 8 12
```

Sample Output 1

```
1
0
0
1
0
1
```

Sample Input 2

```
8 10
0 0
1 1
5 5
8 8
11 11
14 14
18 18
19 19
17 10 13 9
15 15 15 9
7 3 17 4
15 15 12 3
9 17 3 3
4 4 9 4
12 12 2 6
10 15 6 6
15 17 4 16
-1000000000 -999999999 15 7
```

Sample Output 2

```
1
0
0
0
0
0
0
0
0
0
```