2024

10th Annual

Plano West Senior High Programming Competition

"Code Your Move, Master the Game!"

February 17th, 2024 — Plano, Texas

Hail, Noble Lady

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 5

Problem

Print out the image of our dear noble lady!

Input Format

N/A

Output Format

Print the ASCII art of the queen chess piece. Trailing spaces will be ignored.

```
   _(_)_
  _/___\_
   /___\
  [|||||]
   \___/
   |||||
   |___|
   |___|
  /_____\
 (_______)
(_________)
/_________\
```


Author: Emily Lou

King's Chessboard Mapping

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 10

Problem

As the commander of his army, the King needs to keep track of his soldiers' movements on the chessboard for an effective military strategy. The chessboard is an 8×8 grid, and each square is represented by a unique integer. For instance, the square `A1` corresponds to the integer `1`, `B1` corresponds to `9`, and `H8` corresponds to `64`.

Given N test cases, each containing an integer representing a square on the chessboard, your task is to print the corresponding chess notation for each square.

Input Format

The first line of input will contain N ($N \leq 10^4$), the number of test cases. The following N lines will each contain a single integer, representing a square on the chessboard.

Output Format

The output should consist of N lines. Each line should contain the chess notation corresponding to the integer in the input.

Sample Case Input

```
3
1
9
64
```

Sample Case Output

```
A1
B1
H8
```

Sample Case Explanation

- The number 1 corresponds to the first square on the chessboard, which is `A1`.
- The number 9 corresponds to the first square in the second row, which is `B1`.
- The number 64 corresponds to the last square of the chessboard, which is `H8`.

Author: Khosraw Azizi

Pawn's Pawnshop

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 15

Problem

A pawn runs a pawnshop (no pun intended). The pawn accepts any items with a value of less than or equal to N and sells them for a value that is twice the value they were bought for.

Given a list of K items, with their respective values, your task is to determine the maximum profit the pawn can make. If any item is not sold, it does not contribute to the expenses or the profit – it is as if the item was never bought.

Input Format

The first line of the input will contain two space-separated integers N and K ($N, K \leq 10^3$). The next line will contain K space-separated integers representing the value of each item.

Output Format

The output should be a single line containing the maximum possible profit the pawn can make, assuming all items bought are sold.

Sample Case Input

```
100 5
50 90 120 30 80
```

Sample Case Output

```
250
```

Sample Case Explanation

- The pawn can buy the items valued at \$50, \$90, \$30, and \$80 (since their values are less than or equal to \$100). The item valued at 120 is not bought as its value exceeds \$100.
- The pawn resells the bought items for \$100, \$180, \$60, and \$160 respectively (double the original cost).
- The total profit is the difference between the price the items were bought at and the price the items were sold at, which is \$250.

Author: Khosraw Azizi

Chess Language Translation

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 20

Problem

There is a magical chess kingdom where chess pieces speak a language unique to their kind. However, the language used among all chess pieces isn't uniform, causing a lot of confusion in their communication.

In the given language, every word is represented as a string of lowercase Latin letters. The chess pieces have a special dictionary to translate words from one piece's language to another. The dictionary can be represented as two strings, each consisting of 26 characters, where the i-th character in the first string is translated to the i-th character in the second string.

Translate a sentence from the source piece's language into the target piece's language.

Input Format

The first and second lines of input will contain two strings of 26 lowercase Latin letters, the source and target language, respectively. The third line of input will contain the sentence in the source language, consisting of only lowercase Latin letters and spaces.

Constraints

The length of the sentence does not exceed 1000 characters. It is guaranteed that all letters appear in both dictionary strings.

Output Format

A single line containing the translated sentence in the target language.

Sample Case Input

```
abcdefghijklmnopqrstuvwxyz
qwertyuiopasdfghjklzxcvbnm
the quick brown fox jumps over the lazy dog
```

Sample Case Output

```
zit jxoea wkgvf ygb pxdhl gctk zit sqmn rgu
```

Sample Case Explanation

Translating "the quick brown fox jumps over the lazy dog" using the provided mapping gives "zit jxoea wkgvf ygb pxdhl gctk zit sqmn rgu".
Each letter in the sentence is replaced with its counterpart from the target language's alphabet. For example, `t` → `z`, `h` → `i`, `e` → `t`, resulting in "zit" as the translation for "the", and so on for the rest of the sentence. Spaces between words are retained, ensuring the structure of the sentence remains intact.

 Author: Khosraw Azizi

Binary Boards

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 25

Problem

After a long night of playing board games, a group of friends get ready to take their respective board games with them and return home. However, while cleaning up, they discover that the names of their board games have been converted into binary. Please help them return each board game to their rightful owner!

Input Format

The first line will contain N ($1 \leq N \leq 26$), the number of friends. The next N lines will each contain a friend's name and the unscrambled, **fully capitalized** name of the board game that belongs to them (**guaranteed to be a single word**).

The next N lines after that will contain the board game number and each letter of the binary-converted board game name, all space-separated.

For the encryption, let $A = 0,\ B = 1,\ C = 2, \ldots,\ Z = 25$. These numbers are in base 10.

Output Format

N lines, each containing the friend's name and the corresponding board game number that belongs to them. Print the friends and board game numbers, separated by a space, in the order they are provided.

Sample Case Input

```
4
A BATTLESHIP
B GAMEOFLIFE
C RISK
D MONOPOLY
1 10001 1000 10010 1010
2 110 0 1100 100 1110 101 1011
1000 101 100
3 1100 1110 1101 1110 1111 1110
1011 11000
4 1 0 10011 10011 1011 100 10010
111 1000 1111
```

Sample Case Output

```
A 4
B 2
C 1
D 3
```

Sample Case Explanation

For board game 1, the binary converts to '17 8 18 10' in base 10. According to the key ($A = 0,\ B = 1,\ C = 2$, etc.), this translates to 'R I S K'. When put together, the board game name is `RISK`, which is C's board game. So, board game number 1 belongs to C.

For board game 2, the binary converts to '6 0 12 4 14 5 11 8 5 4' in base 10, which yields `GAMEOFLIFE`, the game that belongs to B. Proceeding likewise reveals the sample output.

Author: Annie Li

Cryptography I

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 30

Problem

The Rooks have devised a key generating algorithm that they want you to test. They want to use the Fibonacci numbers to help generate new keys; these are defined by

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}.$$

When the key generating algorithm is begun, it will pick a random Fibonacci number and find its remainder when divided by a given number P. The algorithm then picks the next term of the Fibonacci sequence and outputs its remainder when divided by P. (So if the first key provided is based on the N-th Fibonacci number, the next will be based on the $(N+1)$-th Fibonacci number.) However, the Rooks suspect that this algorithm is not secure. Can you prove to them that it isn't?

Input Format

The first line of input will contain the given number P $\left(P \leq 10^9\right)$. The second line of input will contain two keys that have been generated consecutively. There is no constraint on N.

Output Format

A single line containing the next key that would be generated.

Sample Case Input

```
7
6 5
```

Sample Case Output

```
4
```

Sample Case Explanation

In the sample case, the keys were generated with the consecutive Fibonacci numbers of 55 and 89, which leave remainders of 6 and 5, respectively, when divided by 7. The next Fibonacci number in the sequence would be 144, which leaves a remainder of 4 when divided by 7.

Author: Aaron Zhou

Connect 4

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 35

Problem

Carl Fredrickson enjoys watching his two grandchildren, Tim and Sally, play Connect 4. However, his grandchildren often argue about who won. Carl has poor eyesight and can not see the game pieces clearly. He remembers, however, that a player has won Connect 4 when four of their pieces are in a row either vertically, horizontally, or diagonally. Can you help Carl determine if either of his grandchildren won?

Input Format

The first line contains an integer N ($N \leq 100$), the number of lines to follow, followed by N lines each representing one row of 7 space-separated characters in the Connect 4 board.
Each character will be an `X`, representing Tim's move, or an `O`, representing Sally's move.

Constraints

It is guaranteed that at most one player will have won.

Output Format

The output will consist of a single line, containing "Tim Won" if Tim won, "Sally Won" if Sally won, or "Tie" if neither won.

Sample Case Input

```
6
X X O X X O O
X O X X O O X
O O X O X X X
X O O X O O O
O X X O O X O
X O O X X O X
```

Sample Case Output

```
Sally Won
```

Sample Explanation

There are 4 `O`'s in a row diagonally, so Sally has won.

Author: Emily Lou

Landing On Squares

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 40

Problem

In a certain strange board game, dice with N sides and irregular numbering are used. All dice used in the game are identical. Like in many games, two dice are rolled and their sum is the number of squares you move forward by. You have identified T detrimental squares that you do not want to land on. For each square, can you determine if it is possible to land on it on your next turn?

Input Format

The first line contains N ($1 \leq N \leq 2 \cdot 10^4$) and T ($1 \leq T \leq 100$). The next N lines each contain the numbers on each dice. The next T lines each contain the number of squares away from you each detrimental square is.

Output Format

T lines, with the i-th line containing "yes" if it is possible to land on the i-th detrimental square, and "no" if it is not.

Sample Case Input

```
5 4
9
5
10
4
8
11
14
16
23
```

Sample Case Output

```
no
yes
yes
no
```

Sample Case Explanation

The two dice each have sides with values 9, 5, 10, 4, 8.

11 and 23 cannot be created by summing the values on the two dice, while 14 and 16 can by using $14 = 9+5$, $16 = 8+8$, respectively.

Author: Andrew Yu

Chest of Greed

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 45

Problem

Four players are participating in a *Lairs and Lizards* campaign. After defeating the dragon's challenge, they see a chest. As they get closer, however, they see a goblin squatted next to the chest, seemingly waiting for someone. Approaching the goblin, the players ask what it is waiting for. The goblin then tells them about the chest, enchanted to only allow the winner of the chest's game to claim the gold. He says:

> There are a total of N coins in the chest. You and I will take turns. On each of our turns, we will take either 1, 3, or 4 coins. You must never take exactly two coins! The side that takes the last coin wins and gets to keep all the gold. Out of the kindness of my heart, I will let you guys go first. How many coins will you take?

Given the number of coins in the chest, N, determine whether the players can win if both sides play optimally.

Input Format

A single number N ($1 \leq N \leq 10^6$) denoting the number of coins in the chest.

Constraints

The players all act as one entity; they do not get four turns.

Output Format

Print "We're rich!!!" if the players win; print "Catch that goblin!" if the players lose.

Sample Case Input

```
5
```

Sample Case Output

```
We're rich!!!
```

Sample Case Explanation

On their first turn, the players may take 3 coins. The goblin is then forced to take one coin, which leaves the players the last coin to take.

Author: Alex So

Board Game Sorting

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 50

Problem

Aaron has been tasked with sorting all the board games in his board game collection. Since he has a lot of board games, he decides to use a version of Quicksort to sort his games. In order to do this, Aaron randomly picks a game to use as the pivot, and puts all games that would come before the pivot alphabetically to the left of the pivot and all games that would come after the pivot alphabetically to the right of the pivot. After performing one step of this algorithm, Aaron realizes he put his phone in the board game that he picked as his pivot. Can you help him find his phone?

Input Format

The first line of the input contains N ($1 \leq N \leq 10^6$), the number of board games in Aaron's collection. The next N lines that follow will each contain the name of a board game, with the i-th entry representing the i-th book from the left after one step of the Quicksort algorithm.

Output Format

The first line of output should contain M, the number of possible pivots. The next M lines should contain each of the possible pivots, sorted in alphabetical order.

Sample Case Input

```
4
King of Tokyo
Hanabi
Pandemic
Terraforming Mars
```

Sample Case Output

```
2
Pandemic
Terraforming Mars
```

Sample Case Explanation

The only two board games that have all the games to their left before them alphabetically and all the games to their right after them alphabetically are Pandemic and Terraforming Mars, so those are the only possible pivots.

Author: Aaron Zhou

Point Calculations

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 55

Problem

Madeline is playing a series of games in which she wants to calculate the number of points she expects to get from all the games combined. Each game can be described by two numbers, a_i and b_i, and the expected number of victory points granted by a game with these parameters is $a_i/(b_i!)$. If M is the expected total number of points over all the games combined, help Madeline find the absolute value of $\log_{10}(M)$, rounded to 3 decimal places.

Input Format

The first line contains N ($N \leq 100$), the number of games Madeline wishes to play. Then, N lines follow, each containing two space-separated integers, a_i and b_i ($a_i, b_i \leq 1000$), respectively, detailing the parameters of one game.

Output Format

A single line containing the absolute value of $\log_{10}(M)$ rounded to 3 decimal places.

Sample Case Input

```
2
2 4
3 5
```

Sample Case Output

```
0.965
```

Sample Case Explanation

The first game provides a point value of 2/24 points. The second game provides a point value of 3/120 points. In total, both games provide 13/120 points together, which has a $\log_{10}$ of -0.965.

Author: Aaron Zhou

Battleship

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 60

Problem

In the game of Battleship, players place their $1 \times n$ rectangular ships on a grid horizontally or vertically. Each turn, a player must guess a square on the opponent's grid. The opponent must state whether a ship has been hit. Both players then mark this play. Once all squares that a player's ship occupies have been marked, the ship is **sunken**.

Brian has almost won a game of Battleship, but there is still one ship remaining that he has not sunk. On his guessing grid, he has marked all of his hits **on this ship** with an `X` and his misses with an `O`. Help Brian win by calculating the maximum number of guesses he needs to sink this last marked ship if he plays optimally. Assume that all `X`'s belong to the current ship and hits on already-sunken ships are ignored.

Input Format

The first line is an integer, N ($1 \leq N \leq 1000$), denoting the size of the $N \times N$ grid. The next N lines are strings of length N. Empty squares are marked `.`, "miss" squares are marked `O`, and "hit" squares are marked `X`.

Constraints

- There is at least one `X` on the grid.
- A valid ship must exist covering the `X`s.
- If the ship is guaranteed to already have been sunk, output 0.

Output Format

A single integer, denoting the maximum number of guesses Brian must make with optimal gameplay.

Sample Case Input

```
7
..O..O.
...O...
...XOO.
.O.....
....O..
...O...
O......
```

Sample Case Output

```
3
```

Sample Case Explanation

Brian can first guess the third square in the third row. If this is a hit, he knows the ship is horizontal, and at most two more guesses are needed to sink the whole ship. If it is a miss, then he knows the ship has to be vertical, and so at most two more guesses can sink the ship there as well.

 Author: Andrew Yu

Uno

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 65

Problem

After a long day of work, Alice, Brian, Casper, and Dorothy decide to play a modified version of Uno with only regular, plus fours, skips, and reverse cards.

To preserve their friendship, the four have decided on a few ground rules:

- plus four cards adds 4 cards to the next player and skips that player's turn.
- plus four cards cannot be played by consecutive players.
- skip cards skip the next player's turn.
- reverse cards reverse the turn order, **letting the original player who put down the reverse card play another card.**

Each player begins with 5 cards. Assuming that the turn order always starts off as Alice, Brian, Casper, and then Dorothy, find which of the four have the highest number of cards in their hand at the end of the game.

Input Format

The first line will contain an integer N ($1 \leq N \leq 100$) denoting how many games are played. The next N lines contain strings, one per line, that represent the cards played during a game. Regular cards are represented by either a `r`, `g`, `b`, or `y`. Plus 4 cards are represented by a `4`, skip cards are represented by a `0`, and reverse cards are represented by a `v`.

Output Format

A single line, with the name of the player who is holding the most cards at the end of each game played. If there is a tie, output "Tie".

Sample Case Input

```
1
rr4gggg04v
```

Sample Case Output

```
Dorothy
```

Sample Explanation

For this game, players Alice and Brian each play a red card, bringing their card total down to 4. Then, Casper plays a plus 4, adding 4 cards to Dorothy's total and skipping her turn. The four all play one card, reducing their respective card totals by 1. Alice then plays a skip, skipping Brian's turn, and then Casper plays a plus 4, adding 4 cards to Dorothy's hand and skipping her turn. Alice plays a reverse card, and the game ends. Tallying up their card totals yields 1, 3, 2, and 12 for Alice, Brian, Casper, and Dorothy respectively. Therefore, Dorothy is the player with the most cards.

 Author: Annie Li

Mao

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 70

Problem

In the game of *Mao*, players play with a standard 52-card deck of cards and create the rules they go. At the end of every round, a new rule is added. The Plano West CS Officers are playing a game of Mao. However, since the officers are particularly unoriginal people, in every game of Mao they play, the first three rules they use are the same:

1. *Suits Rule*: A club must be followed by a club or diamond. A diamond must be followed by a heart or spade. A heart must be followed by a heart or club. A spade must be followed by a diamond or spade.
2. *Difference Rule*: Consecutive cards played must have a difference of d or $d + 1$, with wraparound allowed. (Thus, a Jack and an Ace can be said to have a difference of 3 or 10).
3. *Divisibility Rule*: Cards must alternate being divisible and not being divisible by x.

To keep things interesting, the order these three rules are added and the values of d and x are changed every time the officers play a game of Mao. It is currently round 3 of their game (so 2 rules have been added), and Alex has seen 5 cards played. However, Alex has suddenly forgotten which two rules have been instituted, as well as the values of d and x if either rule 2 or 3 has been instituted. It is now Alex's turn. Given the four cards in Alex's hand, can you tell which card is playable.

As a reminder, the card ranks are A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, in that order. For the purposes of the divisibility rule, A = 1, J = 11, Q = 12, and K = 13.

Input Format

The first line will contain 5 space-separated strings. Each string will have the rank of a card and its suit, in that order. `C` will be used to denote clubs, `D` for diamonds, `H` for hearts, and `S` for spades. These are the five cards in play, given in the order that they were played. The second line of input will have 4 space-separated strings, representing the cards in Alex's hand.

Output Format

A single line containing the playable card in Alex's hand. If no card is playable, then output "The Mao gods have befallen me!". Alex is guaranteed to have at most one playable card, and it will be always possible to determine which rules are in play, as well as the values of d and x, if applicable, from the 5 cards given.

Sample Case Input

```
AH 3H 7C QD 5S
2C 6D KD 4S
```

Sample Case Output

```
6D
```

Sample Case Explanation

From the data given, it can be determined that the two rules that apply are the Difference Rule with $d = 2$ and the Suits Rule. Out of the cards in Alex's hand, the only playable card is the 6 of Diamonds following these rules.

Author: Aaron Zhou

Simultaneous Exhibition

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 75

Problem

Top chess grandmaster Stock Fish is playing a simultaneous exhibition match: he is playing each of N opponents at the same time. In fact, Mr. Fish is so good at chess that he knows exactly how many moves he needs to beat each opponent, at which point they will immediately begin playing him again. (For example, if Stock can win in 2 moves, then he can win 2 games from this opponent in 4 moves).

How long will it take Mr. Fish to win G games?

Input Format

The first line contains positive integers N ($N \leq 10^4$) and G ($G \leq 10^5$), the number of opponents and the number of games Stock needs to win, respectively. The next line contains N integers, where the i-th integer is M_i ($M_i \leq 10^4$), the number of moves Mr. Fish needs to beat the i-th opponent.

Output Format

A single integer, representing the minimum number of moves required for Stock Fish to win G games.

Sample Case Input

```
3 7
3 2 5
```

Sample Case Output

```
8
```

Sample Explanation

In 8 moves, Stock wins 2, 4, and 1 game against the three opponents respectively, for a total of 7 games won. In only 7 moves, he would not yet have won the 8th game.

Author: Alex Zheng

Knight's Conquest

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 80

Problem

Horsey the knight is on an chessboard and needs to visit his friend Jorsey! He moves by going two squares in one direction, and then one square in a perpendicular direction (forming an L or J shape).

Input Format

The first line of input will consist of two space-separated integers M, N ($M, N \leq 5000$), representing the size of the $M \times N$ chessboard. The next M lines of input will contain N space-separated characters each. A `.` will denote a square that is able to be visited, while an `X` will denote a square that is not able to be visited. An `H` will denote Horsey's start square, while a `J` will denote Jorsey's square (Horsey's end square).

Output Format

A single integer, representing the minimum number of moves required for Horsey to reach Jorsey, or "IMPOSSIBLE" if he cannot do it.

Sample Case Input

```
4 5
. . X . X
. X . . .
. . H J .
. X . . .
```

Sample Case Output

```
IMPOSSIBLE
```

Sample Case Input

```
4 5
. . X X X
X X . X X
. H J . .
. . . . .
```

Sample Case Output

```
5
```

Sample Case Explanation

In the first sample case, Jorsey is unreachable by any sequence of knight's moves contained in the 4×5 grid beginning at Horsey's start square. In the second sample case, Horsey can reach Jorsey in 5 knight's moves contained within the grid.

Author: Alex Zheng

Intercontinental Ballistic Missile

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 85

Problem

After the opponent blunders a queen, you suddenly gain the ability to launch an **Intercontinental Ballistic Missile** (ICBM) at the $N \times N$ chessboard. However, you must be careful where you launch the missile, because if any of your own pieces are within the blast zone, you will lose material. For each of the Q given blast zones, calculate the **net point value** of the zone.

Input Format

The first line of input will contain two integers: the size of the square chessboard, N $\left(1 \leq N \leq 10^3\right)$ and the number of blast zones, Q $\left(1 \leq Q \leq 10^6\right)$. The next N lines will each contain a string of length N, representing a row of the chess board. The following Q lines after that will each contain four integers, indicating the two corners of the rectangular blast zone: (x_1, y_1) and (x_2, y_2) $(1 \leq x_1 \leq x_2 \leq N, 1 \leq y_1 \leq y_2 \leq N)$.

Pieces that are represented by a lowercase letter are opposition pieces and will gain you material if destroyed, while pieces represented by an uppercase letter are your pieces and will lose you material if destroyed. As a reminder, pawns (`p`) are worth 1 point, knights (`n`) and bishops (`b`) are worth 3 points, rooks (`r`) are worth 5 points, and queens (`q`) are worth 9 points. Empty squares are denoted with an underscore.

Output Format

The net point value of each of the Q blast zones on separate lines.

Sample Case Input

```
3 2
bnr
p_P
NBR
1 1 1 3
1 1 3 3
```

Sample Case Output

```
11
0
```

Sample Case Explanation

In the sample case, the 3×3 chessboard is given as input. The first query, which is the blast zone with corners at $(1, 1)$ and $(1, 3)$, contains an opposition bishop, knight, and rook, totaling $3 + 3 + 5 = 11$ points. The second query has a net point value of all pieces added together, totaling to 0 points.

Author: Randy Guo

Bishops' Alarm Defense Strategy

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 90

Problem

The Castle of Bishops is under siege by knights. There are N knights outside the castle, each having a unique attack value $(a_1, a_2, \ldots, a_N)$. The knights will attack sequentially, in the order they are listed, inflicting damage equal to their respective attack values.

The bishops inside the castle have a unique defense mechanism: an alarm system. This system is represented by a counter that starts at 0. With each knight's attack, the bishops can increase the alarm counter by 1, 2, or 3. If the alarm counter reaches or exceeds 8, the attacking knight's damage is nullified (set to 0), and the alarm counter is reset to 0.

Your task is to help the bishops determine the minimum total damage that the knights can inflict.

Input Format

The first line of input will contain N ($1 \leq N \leq 10^5$), the number of knights. The second line will contain N space-separated integers, each representing a_i ($1 \leq a_i \leq 10^9$), the attack value of a knight.

Output Format

A single line representing the minimum possible total damage the knights can inflict.

Sample Case Input

```
5
2 3 5 6 2
```

Sample Case Output

```
12
```

Sample Case Explanation

The bishops can raise the alarm counter by 2 each time, causing the alarm counter to hit 8 when the knight with attack 6 is attacking. This leads to a total damage of $2 + 3 + 5 + 0 + 2 = 12$.

Author: Aaron Zhou

Lairs and Lizards

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 95

Problem

In the hit game *Lairs and Lizards*, character creation is done on a board of N spaces, numbered conveniently from 1 to N. Each space i has three statistics: power, denoted P_i, smartness, denoted S_i, and funniness F_i, each ranging from 0 to 20. In order to create a character, two tokens are placed on the board. If the tokens are placed on squares i and j with $i \leq j$, then the character will have power equal to $\max(P_i, P_{i+1}, \ldots, P_j)$ and have analogous statistics for smartness and funniness. A character is considered be **overpowered** if at least two of its three statistics are at least 15. Given a board, determine the maximum value of $|j - i|$ over all i and j satisfying the condition that a character created from tokens placed on squares i and j is not overpowered.

Input Format

The first line will contain N ($N \leq 10^6$), the number of squares on the board. Then, N lines follow, each detailing the P_i, S_i, and F_i of a square, respectively.

Output Format

A single line containing the maximum value of $|j-i|$ over all i and j satisfying the condition that a character created from tokens placed on squares i and j is not overpowered.

Sample Case Input

```
6
10 16 12
15 8 6
19 10 4
5 6 14
13 12 11
8 20 17
```

Sample Case Output

```
3
```

Sample Case Explanation

It can be seen that putting tokens on squares 2 and 5 will lead to a character with statistics $P = 19$, $S = 12$, and $F = 14$, which is not overpowered. It can be shown that any other choice of token that causes a larger $|j - i|$ will be overpowered.

Author: Aaron Zhou

Heap Stack

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 100

Problem

Bob has quite a large board game collection with N rectangular games. He wants to put them all on his board game shelf, but he wants to make sure none of them get damaged. To ensure the ultimate preservation of his games, he follows the following rules:

1. No board game can be rotated.
2. No board game with a larger width can be placed on top of a board game of smaller width.
3. No board game can be placed on top of two other board games simultaneously.
4. No board game can have more than one other board game above it.

Given the sizes of each of his board games and the width of the board game shelf, can you help Bob find the maximum number of games he can place on his shelf?

Input Format

The first line contains two space-separated integers, N ($N \leq 10^5$) and W ($W \leq 10^9$), the number of games Bob owns and the width of his board game shelf, respectively. Then, N lines follow, each detailing w_i ($w_i \leq 10^9$), the width of a single game.

Output Format

A single integer representing the number of games he can fit on the shelf.

Sample Case Input

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

Sample Case Output

```
3
```

Sample Case Explanation

3 games can be placed on the shelf. The game of width 2 can be placed on top of the game of width 3, and the game of width 1 can be placed alone. This leads to 3 games total. It can be easily shown that it is not possible to fit all 4 games on the shelf.

Author: Aaron Zhou

Numbers Game

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 105

Problem

Ryan wanted to buy a copy of Scrabble, but he got it off of Wish and the tiles are all wrong! Instead of letter tiles, he gets a set of number tiles instead. Determined to make this a fun game, he takes out two sets of numbers from 1 to 9. He rearranges one of the sets to form a sequence of nine numbers, $a_1, a_2, \ldots, a_9$. In order for him to score a point, he needs to create another 9-digit number N with his second set of tiles such that for all i from 1 to 9, the number formed by taking the first i digits of N are divisible by a_i. Given the sequence formed by the first set of tiles, can you help Ryan score a point?

Input Format

The first line contains T ($T \leq 1000$), the number of test cases to follow. Then, each of the next T lines contains a 9-digit number M, representing the sequence formed by the first set of tiles.

Output Format

T lines, each containing a valid sequence that scores Ryan a point, or -1 if no such sequence exists.

Sample Case Input

```
1
123456789
```

Sample Case Output

```
381654729
```

Sample Case Explanation

It can be confirmed that 3 is divisible by 1, 38 is divisible by 2, 381 is divisible by 3, 3816 is divisible by 4, etc. Thus, this is a valid sequence.

Author: Aaron Zhou

Knights' Layer Defense Strategy

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 110

Problem

The Knights are under attack! The Bishops, having successfully defended against the Knights' onslaught, are now marching towards the Knights' castle. In order to mount a defense, the K Knights have arranged themselves on an $N \times N$ chessboard to protect each other in **layers**, where a knight defending another knight is one layer, and a knight defending a knight defending another knight is two layers of protection. A knight in chess defends another knight if it is two squares in one direction, and then one square in a perpendicular direction (forming an L or J shape). The Knights have also arranged themselves in such a manner as to guarantee **at most one path** between every pair of Knights. Now, the Knights need to know how many layers of protection they have against the Bishops.

Input Format

The first line of input will contain two integers: the size of the square chessboard, N, and the number of knights, K $\left(1 \leq N \leq 10^3, 1 \leq K \leq 10^5\right)$. The next K lines will each contain two integers, indicating the rank r and file f $(1 \leq r, f \leq N)$ of a knight.

Output Format

A single integer representing the number of layers of protection of the most protected knight.

Sample Case Input

```
5 5
1 1
2 3
3 2
5 3
5 5
```

Sample Case Output

```
3
```

Sample Case Explanation

In the sample case, the knight at $(1,1)$ defends, and is defended by, the knights at $(2,3)$ and $(3,2)$; and the knight at $(3,2)$ defends, and is defended by, the knights at $(1,1)$ and $(5,3)$. Thus, the knight at $(2,3)$ has 3 layers of protection through the knights at $(1,1)$, $(3,2)$, and $(5,3)$. The knight at $(5,3)$ similarly has 3 layers of protection. The knight at $(5,5)$ is not defended by anybody and has 0 layers of protection. Therefore, the knight at $(2,3)$ and $(5,3)$ have the most protection, with 3 layers each.

Author: Randy Guo

Broken Sensing

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 115

Problem

Treasure has been buried on the chessboard, and all the pieces are scrambling to find it. The Grand Pawn Society (GPS) thinks it has found a way to reliably determine the correct location. The GPS views the chessboard as a large $N \times N$ grid of squares, with rows and columns each labeled from 1 to N. The society has placed four sensors at various different locations around the chessboard. Each sensor is able to calculate the distance from it to the treasure, but not precisely where the treasure is. In order to scope some additional information from each sensor, the GPS has programmed the sensor to use a specialized distance formula. If a sensor is on row X and column Y and the treasure is on row x and column y, then the sensor will measure a distance of

$$D = \sqrt[3]{(x-X)^3 + (y-Y)^3}.$$

However, due to data limitations, the GPS has only been able to get the sum of the four distances produced by each of the sensors, but not the individual distances. Given this limitation, help the GPS determine the location of the treasure.

Input Format

The first line of the input will contain N ($1 \leq N \leq 10^5$), the size of the chessboard. The next four lines of the input will contain two space-separated integers, the row and column of each sensor, respectively. The final line will contain D ($|D| \leq 10^9$), the sum of all distances measured, *truncated* to three decimal places.

Output Format

A single line containing two space-separated integers, the row and column, respectively, of the treasure. If multiple such locations exist, output the first location in reading order (also known as numerical order). It is always guaranteed that one such location will exist.

Sample Case Input

```
5
1 1
4 2
2 4
3 5
-2.912
```

Sample Case Output

```
2 3
```

Sample Case Explanation

The distances of each of the sensors to the provided treasure location are 2.080, -1.912, -1, and -2.080, respectively. It can be seen that the sum of these are the desired distance, and this happens to be the only possible location.

Author: Aaron Zhou

Chess75600

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 120

Problem

Chess960, also known as Fischer random chess, is a chess variant where the starting position of the pieces is randomized with a few constraints: the bishops must be on opposite colors (one index is odd and the other is even), and the king must be between the rooks. Bored of this variant, some chess pieces have invented a new variant they call Chess75600, where an extra queen and knight are added, and the starting position of the pieces is randomized with no constraints, resulting in 75,600 possible positions. Now, the pieces are bored again and want to convert Chess75600 back into a Chess960-like position (Chess14000), with the knowledge that a swap between two pieces costs the absolute difference of their values, multiplied by the absolute difference of their indices. Determine the minimum cost required to convert the given Chess75600 position to a Chess14000 position that follows Chess960's constraints.

Input Format

The first line of input will contain a string of length 10, the starting Chess75600 position. As a reminder, knights (`n`) and bishops (`b`) are worth 3 points each, rooks (`r`) are worth 5 points, and queens (`q`) are worth 9 points. For the purposes of this problem, kings (`k`) are worth 200 points.

Constraints

There will be exactly 3 knights, 2 bishops, 2 rooks, 2 queens, and 1 king in the position.

Output Format

A single integer representing the minimum cost required to convert the given Chess75600 position to a valid Chess14000 position.

Sample Case Input

```
qrrnkbnqnb
```

Sample Case Output

```
6
```

Sample Case Explanation

In the sample case, the given starting position is invalid because the bishops are not on opposite colors and the king is not in between the rooks. This can be corrected with just 1 swap between the second rook from the left and the leftmost bishop, which would cost $|r - b| \times |i_r - i_b| = (5 - 3) \times (5 - 2) = 6$.

Author: Randy Guo

Spy Mission

Time Limit: 4 seconds

Memory Limit: 512 MB

Victory Points: 125

Problem

The Pawns and Rooks are on a spy mission deep behind enemy lines. The P Pawns and R Rooks are on an $N \times N$ chessboard, where the enemy King is sleeping. It is the Pawns and Rooks' turn to move, but they do not want to check the King and alert his guards. A check occurs when the King is one space in front of and one space to either the left or right of a pawn (diagonal to the pawn), or when the King is in the same row or column as a rook with no pieces in between. Determine the number of possible moves that will **not** result in a check. A rook can move horizontally or vertically through any number of unoccupied squares, and a pawn moves one space forward at a time. The King is guaranteed to not already be in check, and there will be no pawns initially on the N^{th} rank. Note that pawns are able to promote on the N^{th} rank only to a rook and advance 2 spaces on the second rank (as long as it is not also the $(N-1)^{th}$ rank).

Input Format

The first line of input will contain three integers: the size of the chessboard, N, the number of pawns, P, and the number of rooks, R $\left(1 \le N \le 10^5, 0 \le P, R \le 10^5\right)$. The second line will contain two integers, indicating the rank r and file f $(1 \le r, f \le N)$ of the enemy king. The next P lines will each contain the position of a pawn, and the following R lines after that will each contain the position of a rook, in the same format.

Output Format

A single integer representing the number of possible moves that will **not** result in a check.

Sample Case Input

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

Sample Case Output

```
16
```

Sample Case Explanation

In the sample case, the enemy king is at $(5, 5)$. The first pawn at $(2, 4)$ can advance 1 or 2 spaces, but moving 2 spaces would result in a check, so only 1 move is possible. The second pawn at $(3, 6)$ can advance 1 space, but that would result in a check, so no moves are possible. The first rook at $(1, 4)$ is stopped from moving on the fourth file by the pawn at $(2, 4)$, but has 6 possible moves along the rank, excluding 1 that results in a check. The second rook at $(6, 6)$ is blocked on the sixth file by the pawn at $(3, 6)$, but has 9 possible moves, excluding 2 that result in a check. In total, there are $1 + 0 + 6 + 9 = 16$ possible moves.

Rook-Man

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 130

Problem

Rook-Man is a classic game where a rook must move from the top left to the bottom right corner of an $N \times N$ chessboard, collecting points along the way. The rook can only move down and to the right, one square at a time, but it also has K uses of a special ability that allows them to move one square up or the left. Notably, when the rook collects the points on a square, the points are not removed from the square. Find the maximum score Rook-Man can earn.

Input Format

The first line of input will contain two integers: the size of the square chessboard, N ($1 \leq N \leq 400$) and the number of ability uses, K ($0 \leq K \leq 100$). The next N lines will each contain a string of length N, representing a row of the chess board.

As a reminder, pawns (`p`) are worth 1 point, knights (`n`) and bishops (`b`) are worth 3 points, rooks (`r`) are worth 5 points, and queens (`q`) are worth 9 points. Empty squares are denoted with an underscore.

Output Format

The maximum score Rook-Man can earn by taking the optimal path.

Sample Case Input

```
3 1
bnr
p_q
qrp
```

Sample Case Output

```
35
```

Sample Case Explanation

In the sample case, the 3×3 chessboard is given as input. The optimal path Rook-Man can take is represented by the string `RRDUDD`, where `R`, `D`, and `U` mean right, down, and up respectively. This path would earn Rook-Man a score of b + n + r + q + r + q + p = 3 + 3 + 5 + 9 + 5 + 9 + 1 = 35.

Author: Randy Guo

Settling Resources

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 140

Problem

In the game *Settlers of Catan*, players fight over a common pool of 5 resources: sheep, wood, brick, wheat, and ore. One way players can do this is by rolling a die whose result will grant them a specific set of resources.

Randy has obtained a magical die which he can control to help him gain resources. The die will allow Randy to pick the result that appears on its face. However, due its magical powers being too powerful for building materials to handle, the die is unable to produce brick or wood. Additionally, if rolling the die would cause Randy to hold more than 20 of any resource, his civilization becomes too advanced and becomes the Mongols (which is undesirable for the world at large). Finally, for each result Randy picks, he must pay a number of knights specific to the result he chose.

Randy is currently in a game of Settlers of Catan. He needs a specific set of resources to build his town; can you help him determine the minimum number of knights he must sacrifice to gain his resources without forcing his citizens to become the Mongols?

Input Format

The first line of input gives N ($N \leq 200$), the number of faces on the magical die, and Q ($Q \leq 100$), the number of queries Randy makes. Each of the N lines that follow contain 4 space separated integers, s_i, w_i, o_i, k_i ($s_i, w_i, o_i \leq 20$, $k_i \leq 1000$), the amount of sheep, wheat, and ore produced, respectively, followed by the number of knights sacrificed when Randy picks a face of the die. The last Q lines of input will each contain 3 space-separated integers: S, W, and O, the minimum amount of sheep, wheat, and ore, respectively, that Randy needs.

Output Format

Q lines containing the minimum number of knights needed to be sacrificed to build the town. If it is not possible to build the town without holding too many resources, instead print "Oh no, the Mongols!"

Sample Case Input

```
2 1
1 1 2 2
2 3 2 3
7 0 6
```

Sample Case Output

```
11
```

Sample Case Explanation

It can be confirmed the best way to acquire the desired resources is to pick the first face once and the second face 3 times, sacrificing 11 knights in total.

Author: Aaron Zhou

Rabbit's Carrot Quest

Time Limit: 4 seconds
Memory Limit: 512 MB
Victory Points: 150

Problem

In the enchanted land of Fiboria, a clever rabbit finds itself on a magical $N \times N$ grid. The rabbit has a unique way of moving – with each move, it hops a number of tiles, either down or right, equal to the next number in the Fibonacci sequence $(1, 1, 2, 3, 5, 8, 13, ...)$. The rabbit starts its journey from the top-left corner $(0, 0)$ of the grid, with the initial move being 1 tile.

Scattered across the grid are C delicious carrots, each located at a specific (x, y) coordinate. When the rabbit eats a carrot, its excitement resets its hopping pattern back to the first Fibonacci number (1 tile per move). If a required move would cause the rabbit to hop outside the boundaries of the grid, it will hit an invisible wall and lose the game.

Determine whether the rabbit can successfully navigate the grid and eat all the carrots without hitting any walls.

Input Format

The first line contains an integer N ($N \leq 1000$), the size of the grid. The second line contains an integer C, the number of carrots. The next C ($C \leq 100$) lines each contain two integers x and y ($x, y \leq N$), representing the position of each carrot.

Output Format

Print 'Yes' if the rabbit can eat all the carrots without hitting the walls. Otherwise, print 'No'.

Sample Case Input

```
5
2
0 4
4 4
```

Sample Case Output

```
Yes
```

Sample Case Input

```
5
3
2 0
2 2
4 3
```

Sample Case Output

```
No
```

Sample Case Explanation

In sample case 1, the rabbit can reach the carrot at $(0, 4)$ by moving right in hops of 1, 1, and 2 (the first 3 Fibonacci numbers), and then moving down in the same way to reach the carrot at $(4, 4)$.

In sample case 2, it can be shown that there is no way for the rabbit to eat all three carrots.

 Author: Khosraw Azizi

Credits and Acknowledgements

Cover Art:
The cover art for this contest packet was created by Iris Ko.

Problem Writing, Contest Setup, and Review:
Our problems were written and reviewed by a dedicated team of officers from the Plano West Computer Club:

- Khosraw Azizi
- Alex Zheng
- Aaron Zhou
- Randy Guo
- Emily Lou
- Alexander So
- Teagan Gao
- Annie Li
- Andrew Yu

Sponsors:
We gratefully acknowledge the support of Robin Bailey, Plano West Computer Club sponsor, who made this contest possible through her contributions.