New Zealand Programming Contest 2025

PROBLEM SET

Problem	Points	Title
A	3	Golf Tournament
B	3	Change
C	3	Panel Area
D	3	Jumbled Words
E	10	Phone Ban
F	10	Graph
G	10	Seven Wonders
H	10	Game of Darts
I	30	Noah’s Unique Greetings
J	30	Bot Testing
K	30	Flip Grid
L	30	Highway
M	100	Tāwhirimātea
N	100	Krypto’s Chain of Commands
O	100	New Science Centre
P	100	Roadside Produce

PREAMBLE

Please note the following very important details relating to input and output:

- Read all input from the keyboard, i.e. use stdin, System.in, cin, input, Console.ReadLine or equivalent. Input will be redirected from a file to form the input to your submission.
- **Do NOT prompt for input** as this will appear in your output and cause a submission to be judged as wrong.
- Write all output to the screen, i.e. use stdout, System.out, cout, print, Console.WriteLine or equivalent. Do not write to stderr.
- Unless otherwise stated, all *integers* will fit into a standard 32-bit computer word. If more than one integer appears on a line, they will be separated by a space.
- An *uppercase letter* is a character in the sequence 'A' to 'Z'. A *lower case letter* is a character in the sequence 'a' to 'z'. A *letter* is either a lower case letter or an upper case letter.
- Unless otherwise stated, a *name* is a continuous sequence of from 2 to 30 characters (printed or written letters or symbols).
- If it is stated that 'a line contains no more than *N* characters', this does not include the character(s) specifying the end of line.
- Input files are sometimes terminated by a 'sentinel' line. This line should not be processed.

Please also note that:

- You do not need to check that input data meets any specifications given in the problem statement, unless the problem specifically states that you should.
- The filenames of your submitted programs may need to follow a particular naming convention, for example the name of a Java file containing a public class needs to be the name of the class followed by the '.java' extension.
- DOMjudge will reject a submitted file which has any spaces in its file name.
- Problems have a time limit shown on the Problem Set page of DOMjudge. It is usually 1 second, but may be longer. A *TIMELIMIT* error will be issued for submissions that exceed that limit on a single test run.
- Each problem description takes up at least 2 pages, one of which may be empty.

PROBLEM A **GOLF TOURNAMENT** **3 POINTS**

Golf is a popular game around the world. To make a game of golf more competitive between players of different skill levels, each player may be given a handicap which reflects their ability. Golf handicaps range from 0 for the top players up to 36 for complete beginners.

To help attract new members, a local club has organised a one round handicap tournament for local golfers. After adding up the strokes taken to complete the 18 holes, players subtract their handicap from that total to give their net score.

Your task in this problem is to rank players in the tournament based on their net scores.

Input

Input will consist of data describing tournament entrants.

Each line will consist of a name, as defined in the preamble, followed by two space separated integers, H and S.

H represents that player's handicap (in the range 0 to 36 inclusive). S represents their score for the round (in the range 60 to 120 inclusive).

The final line will be X 0 0. Do not process this line. There will be no more than 100 lines of input before the terminating line.

Output

The output will list each player on a single line, in order of net score, lowest first. Those with the same score will appear in their input order.

Each line must show the name of the player, separated by a space from their net score.

Sample Input

```
Gabby 12 90
Felicity 6 80
Charlotte 12 83
Deng 10 82
Hettie 11 92
Albert 3 71
Ewing 15 87
Benny 5 74
X 0 0
```

Output for Sample Input

```
Albert 68
Benny 69
Charlotte 71
Deng 72
Ewing 72
Felicity 74
Gabby 78
Hettie 81
```

Explanation

Net scores are gross scores minus handicaps. In golf, the lowest score wins. Deng and Ewing have equal scores and appear in their input order.

PROBLEM B **WHAT IS MY CHANGE?** **3 POINTS**

Since 2006, the smallest valued coin in circulation in New Zealand has been the 10 cent piece, but shops do not have to make sure that prices of goods are exactly divisible by 10 cents.

Customers paying by EFTPOS do not have to worry as they will be charged the exact amount for their shopping, but what if they want to pay with cash?

Cash transactions are rounded to the nearest 10 cents; this is known as Swedish rounding. The rules are straightforward:

- If the price ends in anything from 1 to 4 cents, it is rounded down to 0
- If the price ends in anything from 5 to 9 cents, it is rounded up to 10

In this problem, you will have to correctly apply Swedish rounding to a shopping bill.

Input

Input will consist of a single shopping bill for which a customer wishes to pay in cash.

The first line will be N, a single positive integer less than 100. This is the number of individual items in the bill. This is followed by N lines, each containing the sale price of a single item in the format `d.cc`, ie at least 1 digit showing the dollar amount, a decimal point, then 2 digits showing the cents.

The final line will be the amount of cash tendered by the customer in the same format as the prices. This is guaranteed to be not less than the total cost of the shopping.

Output

Output will consist of 3 lines as follows:

```
Total price: $d.cc
Rounded price: $d.cc
Change: $d.cc
```

Sample Input

```
5
6.09
6.09
9.95
18.50
8.58
50.00
```

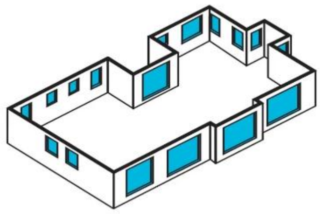
Output for Sample Input

```
Total price: $49.21
Rounded price: $49.20
Change: $0.80
```

Explanation

The total cost of the 5 items is $49.21. As it ends in 1 cent, it is rounded down to $49.20. With $50 tendered, the change is 80 cents.

PROBLEM C **NET WALL PANEL AREA** **3 POINTS**



In construction, the **Net Wall Panel Area (NWPA)** refers to the total area of wall panel excluding all openings. It is an important first calculation step in wall panel design, as windows are a major source of heat loss in a home.

This problem requires you to calculate the NWPA of a given wall.

Input

The input consists of:

Line 1, two positive integers W and H, separated by a space, being the width and height of the overall wall area in millimetres.

Line 2, a positive integer N, the number of window openings which is not greater than 5.

The following N lines each contain two positive integers, separated by a space, being the width and height of one window opening.

All widths and heights will be in the range 100 to 5,000 inclusive.

Output

For each panel, output the **NWPA** in m^2 rounded to two decimal places.

If the total area of window openings equals or exceeds the total wall area, a mistake must have been made with measuring; display the error message shown.

Sample Input 1
```
3948 2420
2
1815 615
1815 615
```

Sample Output 1
```
The Net Wall Panel Area is 7.31
square metres.
```

Explanation for Sample Input & Output 1

- Gross Wall Panel Area, GWPA is $3.948 \times 2.420 = 9.55416$ m^2
- **Net wall panel area, NWPA** is 9.55416 $m^2 - 2 \times 1.12$ m^2
  $= 7.31416$ m^2
  = **7.31 m^2** (rounded to 2 d.p.)

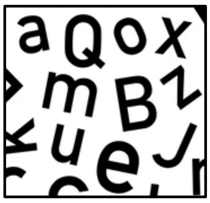
Sample Input 2
```
1948 2420
3
1815 1615
1815 1615
900 400
```

Sample Output 2
```
Please check the dimensions.
```

Explanation for Sample Input & Output 2

- Gross Wall Panel Area, GWPA is $1.948 \times 2.420 = 4.71416$ m^2
- **Net wall panel area, NWPA** is $4.71416 - (1.815 \times 1.615 + 1.815 \times 1.615 + 0.9 \times 0.4)$
  $= 4.71416$ $m^2 - 6.22245$ m^2
  = **-1.50829 m^2**

PROBLEM D | **JUMBLED WORDS** | **3 POINTS**



Did you konw taht the hamun biarn is so cevelr taht it can raed a scnetnee lkie tihs as lnog as ecah wrod has the ccerrot lrettes and the fsrit and lsat lrettes are ccerrot?

I'll translate! Did you know that the human brain is so clever that it can read a sentence like this as long as each word has the correct letters and the first and last letters are correct?

Albert is running a literacy pub quiz and needs help to create several sentences with jumbled words like this. To keep it simple, any word with 4 or more letters must have all letters except the first and last in reverse order.

Input

The first line of input will be a single positive integer no more than 50. It will show N, the number of lines of input to follow.

Each of the following N lines will contain no more than 250 characters consisting of lower case letters and spaces only. There will be a single space between each word, but no leading or trailing spaces.

Output

Output will be one line of text for every input line. In the output line all letters within each word will have been reversed, except for the first and last letters, which will be unchanged.

Sample Input

```
3
welcome to alberts literary pub quiz
did you know that the human brain is very clever
only a short sentence
```

Output for Sample Input

```
wmoclee to atrebls lraretiy pub qiuz
did you konw taht the hamun biarn is vrey cevelr
olny a sroht scnetnee
```

PROBLEM E **PHONE BAN** **10 POINTS**

The New Zealand government has banned cell phones in all schools; schools have to implement the policy.

Mr Hardman, the year 10 dean, decided on a regime of punishments for students who tried to contravene the new regulations. Students are required to hand in their phones at the school gate to the appropriate dean.

Mr Hardman listed the ways students might try to get round the rule and associated points with each strategy. If a student accumulated more than 30 points in a week, they would have to serve a lunch time detention. If they accumulated over 50 points, the student's parents were called in to "please explain".

He drew up a table of misdemeanours. Their corresponding codes and penalties are shown below.

Reason	**Code**	**Penalty Points**
Handing in your granddad's text-only phone from the 1990s and trying to keep yours.	GDX	5
Being caught trying to sneak your phone out of Mr Hardman's basket after handing it in.	SNK	15
Saying "You've already got mine. Sir".	PRV	25
Handing in your late model phone but having a 2nd smart phone hidden.	TWO	20
Lying with "I don't own a phone Sir; you can ask my mother".	LNO	15
Saying "I left it at home", "my dog ate it with my homework", or other obvious fibs.	LRB	10

Input

Input will start with a single line containing a positive integer, W, which is the week number. There then follow several lines, each containing an infringement which occurred during the week. Each line will contain a name, the name of the student who infringed, a space then a 3 letter code from the table above.

The last line will be the word END – this should not be processed.

Output

The first line of output will be the week number from the input.

The remaining output will consist of 2 lists, each occupying two lines. The first line will define the list, the second will consist of a space separated list of names of the students to which the definition applies. Students will be listed in the order they appear in the input.

Continued

The first list will be students who will serve a detention, the second list will be those whose parents will be contacted. Those in the second list will not also appear in the first list. If either list has no students, it must say “None”.

See the sample output for the required wording.

Sample Input 1

```
7
Monica GDX
John PRV
Sam TWO
Pam SNK
John LNO
Pam LNO
Abbey SNK
Chris SNK
Sam LNO
John SNK
Pam LRB
Abbey LNO
END
```

Output for Sample Input 1

```
Week 7
Students for detention:
Sam Pam
Parental meeting for:
John
```

Explanation

Monica and Chris are well below the threshold for punishment.

Abbey has accumulated 30 points so just avoids a detention this week.

Sam has gained 35 points so has to serve a detention.

Pam has been given 40 points so is also in detention.

John has accumulated 55 points, so his parents are informed.

Sample Input 2

```
12
Wilky GDX
Yaz PRV
Ziva TWO
Wilky SNK
Tracey LPR
Ziva LNO
END
```

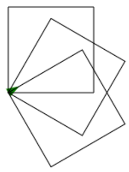
Output for Sample Input 2

```
Week 12
Students for detention:
Ziva
Parental meeting for:
None
```

Explanation

Ziva reaches 35 points and gets a detention, but no student scores over 50 points.

PROBLEM F **GRAPH PLOTTING** **10 POINTS**



Modern computers and printers can usually plot graphics quite easily. However, in the NZPC you will be working with console applications that have far more primitive capabilities.

In this problem you will be required to render a simple graph from pairs of input numbers.

Input

Input will consist of the specification for one graph.

The first line will contain 2 positive integers, C and R, representing the number of columns and rows required for the graph. Both numbers will be in the range 5 to 20 inclusive.

The specification for each curve on the graph will then follow, each on a separate line. The line will start with a single character being the symbol used to represent each point on the curve. This is followed by several pairs of coordinates (c, r) specifying the column and row where that point must appear. The coordinates will be in the range specified by C and R, and are space separated. Input for the line is terminated by the pair `0 0` which should not be processed.

Each graph will have at least 1 curve, and each curve will have at least one point.

The final line will be `#` on its own. This marks the end of input and must not be processed.

Output

Each curve is to be plotted on the graph showing the required symbol at each of the specified c and r coordinates. For all curves, point 1,1 will be at the top left of the canvas with the horizontal axis (the columns) running left to right and the vertical axis (the rows) top to bottom.

IMPORTANT

Because DOMjudge ignores most white space, any part of the graph that does not contain a curve point must be marked with a dot ('.').

Turn over for Sample Input and Output

Sample Input 1

```
11 11
\ 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 11 11 0 0
/ 1 11 2 10 3 9 4 8 5 7 6 6 7 5 8 4 9 3 10 2 11 1 0 0
+ 6 6 0 0
#
```

Output for Sample Input 1

```
\........./
.\......./.
..\...../..
...\.../...
....\./....
.....+.....
..../.\....
.../...\...
../.....\..
./.......\.
/.........\
```

Note that the point 6,6 occurs on all 3 curves, the last one drawn (with +) overwriting the others.

Sample Input 2

```
11 5
* 1 2 2 2 3 2 4 2 5 2 6 2 6 3 7 3 8 3 9 3 10 3 11 3 0 0
% 1 5 2 5 3 5 4 5 5 5 6 5 7 5 8 5 9 5 10 5 11 5 0 0
#
```

Output for Sample Input 2

```
...........
******.....
.....******
...........
%%%%%%%%%%%
```

PROBLEM G **SEVEN WONDERS SCORING** **10 POINTS**

In the game *Seven Wonders*, players construct various structures, including wonders, civilian buildings, commercial buildings, military structures, and scientific achievements. The goal of this problem is to compute the total score of a player based on multiple scoring categories, including their constructed wonders, the buildings they've built, their military strength, their coins, and their science achievements.

Input

Input will consist of 8 lines.

1. The first line contains a positive integer **N** between 2 and 7 inclusive; the number of players.
2. The second line of input will be the names of **N** players, separated by spaces.
3. The third line will contain the corresponding points of Constructed Wonders for each player in order, separated by spaces.
4. The fourth line will contain the number of Military Tokens from their military strength of each player in order, separated by spaces. Each player will have a line of the form `WnXnYnZn`, where n represents a single digit integer. Type W are worth 1 point, X worth 3 points, Y worth 5 points and Z worth minus 1 points.
5. The fifth line will contain the scoring of each player's Civilian Building points based on the completed blue cards, separated by spaces.
6. The sixth line will contain the scoring of each player's Commercial Building points based on the completed yellow cards, separated by spaces.
7. The seventh line will contain the total number of Science cards for each player in order, in the form `AnBnCn`, where n represents a single digit integer. These numbers show how many of each type of card the player has. Each of the 3 sets of card points are squared, and an 8 point bonus is given for each complete set (1 each of A, B and C).
8. The eighth line contain the coins each player has accumulated. Each group of 3 coins contributes one point to the final score

Output

Output the total score for each player and the winner as described in the sample output. The score is found by adding the wonder points, the points from their military strength, the points based on Commercial and Civilian Buildings, the points from the completed sets of science cards and the points based on the number of coins. With coins, every three coins counts for one point.

Where two or more players have equal points, the one with the most coins wins. If still equal it is a tie with players being listed in input order and space separated. In this case `'Winners:'` should be used instead of `'Winner:'`.

Sample Input

```
2
James Samuel
11 15
W1X0Y0Z2 W0X1Y1Z1
2 3
3 4
A3B1C1 A0B2C0
11 5
```

Output for Sample Input

```
James's total score: 37
Samuel's total score: 34
Winner: James
```

Explanation

Player	James	Samuel
Wonders	James has **11 wonder points**.	Samuel has **15 wonder points**
Military	James has total military strength of **-1 point** (1 token of +1 point, 0 tokens of +3 points, 0 tokens of +5 point and 2 tokens of -1 point).	Samuel has military strength of **7 points** (0 tokens of +1 point, 1 token of +3 points, 1 token of +5 point and 1 token of -1 point).
Civilian	**2 points**	**3 points**
Commercial	**3 points**	**4 points**
Science Cards	James has 3 identical type "A" Science symbols, 1 type "B" Science symbol and 1 type "C" Science symbol, so they gain **19 points** from the science cards (3^2 + 1^2 + 1^2 + 8 points for one complete set of Science cards = 9 + 1 + 1 + 8).	Samuel has 0 type "A" Science symbols, 2 type "B" Science symbols and 0 type "C" Science symbols, so they gain **4 points** from the science cards (0^2 + 2^2 + 0^2 + 0 points for no complete set of Science cards = 0 + 4 + 0 + 0)
Coins	James has 11 coins, which equals 3 points for every 3 coins, i.e., **3 points**.	Samuel has 5 coins, which equals **1 point**
Total	11 (Wonder) + -1 (Military) + 2 (Civilian) + 3 (Commercial) + 19 (Science) + 3 (Coins) **James has 37 points**	15 (Wonder) + 7 (Military) + 3 (Civilian) + 4 (Commercial) + 4 (Science) + 1 (Coins) **Samuel has 34 points**

PROBLEM H **A GAME OF DARTS** **10 POINTS**

Darts is a competitive sport where players throw darts at a circular dartboard (Google tells us). As shown in the image, a dartboard is divided into 20 sections each with a value of between 1 and 20 inclusive. A dart landing in a section scores the points shown.

There are 2 rings on the board which appear in different colours. A dart landing in the outer ring doubles the points score while one landing in the middle ring trebles[1] the score. In the centre of the board are two smaller rings making up the bullseye. The outer bull is worth 25, the central bull worth double, 50.

Players take turns to throw 3 darts at a time and win by scoring exactly 501 with their final dart landing in the double ring. If a player scores more in one turn than they need, they have "busted" and score 0 for that turn. They also score 0 if their turn's total would leave them needing less than 2, the lowest possible double. This includes scoring the required amount but not finishing with a double.

Input

The first line of input will be the names of 2 players, separated by a space. Each line thereafter will contain the scores for 3 darts by one of the players. The first named player throws first, then turns alternate.

Each score will start with T if the dart landed in the treble ring, D if it landed in the double ring or S (for single) if it landed in neither. This is followed by a number from 1 to 20, or 25. The scores are separated by spaces. There are two exceptions. 0 means the dart did not land in a scoring section, and X means a dart was not thrown. A second or third dart would not be thrown if the player had won or busted with the dart or darts already thrown.

The final line will contain just END; this marks the end of input and should not be processed.

Output

If neither player has won, the output must state how many points each player must still score and whose turn it is to play. The players must be listed in the order of input.

If one player has scored exactly 501 and their last dart was a double, the output must state that that player has won.

Follow the wording shown in the sample.

Turn over for Sample Input and Output

[1] Treble means multiply by 3, like triple.

Sample Input 1

```
Michael Luke
S20 T20 S20
S20 T20 S20
S20 S20 T20
S20 T20 S12
T20 S18 S18
T19 S25 S7
T20 T20 T20
T20 T1 S17
S9 D16 X
T20 S20 S5
S9 S8 0
S15 D20 X
END
```

Output for Sample Input 1

```
Luke has won.
```

Explanation

The odd numbered turns are Michael’s, the even ones Luke’s.

At the start of turn 9, Michael needs 25. He scores 9 to leave 16 but then hits double 16 instead of double 8. As he has now scored more than 501, he has “busted” and none of his darts on this turn count (so he does not throw his last dart). He still needs 25.

On turn 11, Michael again scores 9 but then scores single 8 to require double 4. His dart misses the double and does not score.

On turn 12, Luke needs 55. He scores 15 leaving 40 which he scores with double 20 thus winning, and does not need to throw his last dart.

Sample Input 2

```
Eric Keith
S20 S20 T18
S20 T20 S20
S20 T19 0
END
```

Output for Sample Input 2

```
Eric needs 330.
Keith needs 401.
Keith to throw next.
```

Explanation

This game has only just started. Eric has had 2 turns. His final throw on his second turn hit a wire and bounced off the board scoring 0. Keith has only had 1 turn so far.

PROBLEM I **NOAH'S UNIQUE GREETINGS** **30 POINTS**

Noah, a friendly Kiwi bloke, has built a reputation around town for his signature greetings. Whether it's `"bro"`, `"bruh"`, or a well-timed `"skibidi"`, he's always got a vibe ready.

But lately, folks have started to notice — "Didn't he call me 'bro man' yesterday too?"
Noah's not in trouble or anything, but he figures it's time to freshen things up. After all, if every greeting feels like déjà vu, where's the flair?

Each morning, Noah picks 3 words he feels like using that day, and how many times he's willing to use each. His goal is simple: greet as many people as he can, with each one getting a totally unique combo of words.

But of course, there are rules to the Noah greeting system:

- Every visitor must get a greeting with at least one word.
- He can't repeat the same word within the same sentence — it wouldn't make sense to say "bro bro".
- Noah can't greet two people with the same set of words — once he's used `"bro bruh"`, that combination is off the table for the rest of the day.
- The order of words in a greeting doesn't matter — `"bruh bro"` is the same as `"bro bruh"`.
- Each word can be used across multiple greetings, but only up to its allowed usage count for the day.

Given these rules, what's the greatest number of people Noah can greet each day?

You will be given `T`, the number of days Noah has greetings planned for.
For each day, you will be given a list of `3` integers, where the i^{th} number represents how many times the i^{th} word can be used that day.

Turn over for input and output.

Sample Input

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

Sample Output

```
3
4
3
1
```

Explanation

On Day 1, Noah has 3 words with usage limits [2, 1, 1].
One possible way Noah can greet 3 people (with order not mattering, and no repeats within a greeting) is:

- `[word0]`
- `[word0, word1]`
- `[word2]`

So on this day, Noah can greet a maximum of **3** people.

On Day 2, with limits [2, 2, 2], he has more options. One possible way Noah can greet 4 people is:

- `[word0]`
- `[word1]`
- `[word2]`
- `[word0, word1]`

This set uses each word within its allowed limit, and all greetings are unique.
There are other sets that also allow Noah to greet 4 people — but no set allows him to greet more.
So the maximum number of people Noah can greet on this day is **4**.

On Day 3, with limits [1, 1, 1]. One possible way Noah can greet 3 people is:

- `[word0]`
- `[word1]`
- `[word2]`

So the output is **3**.

On Day 4, with limits [1, 0, 0]. Noah can only say 1 word.

So the output is **1**.

PROBLEM J **BOT TESTING** **30 POINTS**

The flood of refugees from Auckland to Christchurch is putting pressure on supermarket supply chains. To increase capacity, Woolstuffs is setting up a new robotic supply depot in Rolleston and they need to test the robots.

A robot test environment consists of a wide aisle with 100 numbered storage bays on each side. A super-high-speed conveyor belt runs down the centre of the aisle and there is one robot on each side of the belt, responsible for taking selected items from the storage bays and putting them on the conveyor belt.

Storage bays are 1 metre wide.The robots are named Buzzy and Oona but we'll call them B and O for short. Both robots start at bay 1 on their respective sides of the aisle.

Over the period of one second, a robot can do one of the following:

1. walk one meter in either direction, to the neighbouring bay, or
2. put an item from the storage bay it's currently at onto the conveyor belt, or
3. stay where it is, doing nothing.

To complete the test, the robots need to load a particular sequence of items onto the belt in a specified order. Both robots know the full sequence in advance. How fast can they complete it?

For example, let's consider the following item sequence:

`O 2, B 1, B 2, O 4`

Here, `O 2` means an item from bay 2 on Oona's side, `B 1` means an item from bay 1 on Buzzy's side, and so on.

The robots can satisfy the sequence in 6 seconds using the following strategy.

Time	Oona	Buzzy
1	Move to bay 2	Stay at bay 1
2	Load item from bay 2	Stay at bay 1
3	Move to bay 3	Load item from bay 1
4	Move to bay 4	Move to bay 2
5	Stay at bay 4	Load item from bay 2
6	Load item from bay 4	Stay at bay 2

Turn over for input and output.

Input

The input consists of a single line beginning with a positive integer **N** ≤ 100, representing the number of items to be loaded. This is followed by a single line containing **N** space-separated terms of the form "$\mathbf{R_i}$ $\mathbf{P_i}$", where $\mathbf{R_i}$ is the robot (always 'O' or 'B') and $\mathbf{P_i}$ is a bay position, 1 ≤ $\mathbf{P_i}$ ≤ 100. A single space separates the robot and the bay number.

Output

Output is a single integer giving the minimum number of seconds required for the robots to load all the items, in order.

Sample Input 1

```
4
O 2 B 1 B 2 O 4
```

Output for Sample Input 1

```
6
```

Sample Input 2

```
3
O 5 O 8 B 100
```

Output for Sample Input 2

```
100
```

Sample Input 3

```
2
B 2 B 1
```

Output for Sample Input 3

```
4
```

PROBLEM K — FLIP GRID — 30 POINTS

Marama's architecture class has been asked to suggest possible landscaping for Christchurch's cathedral square, in the unlikely event it ever gets turned back into a people-friendly area.

0	1	1	0
1	0	0	0
0	1	1	0
1	1	1	1

To help with the design, Marama has envisaged the square as being divided into a 4 x 4 grid. Each sub-square is either left empty for recreational use or has some amenity on it (tree, artwork, fountain, etc). Marama represents the layout as a binary grid where 0 denotes an empty sub-square and 1 an amenity.

Marama has written a program that can manipulate such binary grids using the following operations:

1. Rotate the grid clockwise by 90 degrees.
2. Flip a row in the grid, i.e. change all 0's in the chosen row to 1's, and 1's to 0's.
3. Flip a column in the grid, i.e. change all 0's in the chosen column to 1's, and 1's to 0's.

Marama wants to convert a starting grid, A, into a desired target grid, B, using only those operations. Output the minimum number of operations that Marama will need, or -1 if it is impossible to find such a sequence of operations.

Hint: Think of each grid configuration as a ‘state’ in a graph, and each operation (rotate, flip row, flip column) as an edge connecting two states. Can you find a way to explore these states to get from state A to B in the fewest steps?

Input

The first 4 lines of the input each contain a string of length 4, defining the rows in grid A. Each character in the string is either 0 or 1.

The remaining 4 lines of the input each contain a string of length 4, defining the rows in grid B. Each character in the string is either 0 or 1.

Output

The first and only line of the output should contain the minimum number of operations Marama needs to convert grid A to grid B, or -1 if it's impossible to do so, using the operations in the statement.

Sample Input

```
0001
1000
0001
1000
0101
0000
0000
0101
```

Output for Sample Input

```
2
```

Explanation

One possible sequence of operations that converts the first grid to the second grid is:

1. Rotating A clockwise
2. Flipping the first row of A.

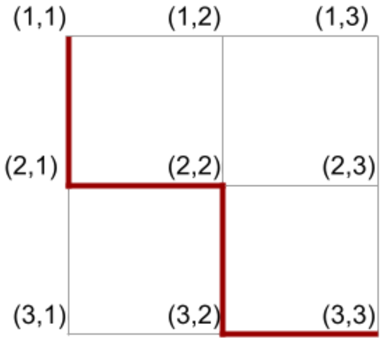
```
0001                 1010                 0101
1000      ⇨          0000      ⇨          0000
0001   Rotate A      0000     Flip        0000
1000   clockwise     0101   first row     0101
```

PROBLEM L **HIGHWAY** **30 POINTS**

Auckland Council is planning to build a new highway. For simplicity, suppose that Auckland is modelled as an *N* by *N* grid (like Manhattan), and each street is a one-way street going either down or to the right. This means that a car at the position *(i,j)* can only move to *(i+1,j)* (down) or to *(i,j+1)* (right). Additionally, a car is not allowed to drive outside of the city, that is, it is not allowed to move to a position where either of the coordinates is bigger than *N*. For example, if *N=10* and a car is at the position *(8,10)*, then it can only move to the position *(9,10)*. Note that the car at the position *(N,N)* cannot go anywhere.

You are given an *N* by *N* matrix *M*, where the cell $M_{i,j}$ denotes how many people from the position *(i,j)* drive to work each day. The goal of the new highway is to minimise the total time **all** the people spend before they get on to the highway.

The highway is modelled as a path that starts at the position *(1,1)* and ends in the position *(N,N)*, and in each step goes either to the right or down. Once the highway is built, each car will take a shortest path to get to the highway. As the streets are one way, a path from a position *(i,j)* to the highway is a sequence of steps, where each step is either down or right and after following these steps we end up in a position through which the highway goes. We denote with $s_{i,j}$ the length of a shortest path to the highway. Note that if the highway passes through a position *(i,j)*, then $s_{i,j}=0$. In the figure below, a highway is denoted with red. The shortest path from the position *(1,2)* to the highway is of length 1 (go down one step); the shortest path from *(1,3)* is of length 2 (go down two steps); the shortest path from *(3,1)* is of length 1 (go one step to the right).



Once you build a highway, the *total travel time* denotes the sum

$$M_{1,1} * s_{1,1} + M_{1,2} * s_{1,2} + \ldots + M_{1,N} * s_{1,N} +$$
$$M_{2,1} * s_{2,1} + M_{2,2} * s_{2,2} + \ldots + M_{2,N} * s_{2,N} + \ldots$$
$$M_{N,1} * s_{N,1} + M_{N,2} * s_{N,2} + \ldots + M_{N,N} * s_{N,N} .$$

Your task is to build a highway with the smallest possible total travel time.

Turn over for input and output.

Input

The first line of the input contains one integer N ($1 \le N \le 100$). Then N lines follow, each containing N numbers (each number is separated from the next by one or more spaces). The j-th number of the i-th line denotes the cell $M_{i,j}$. For every cell we have $0 \le M_{i,j} \le 10$.

Output

Output a single integer which denotes the smallest possible total travel time. You do not need to specify a highway which achieves such a total travel time.

Sample Input 1

```
3
0  10 3
10 0  0
0  5  0
```

Output for Sample Input 1

```
16
```

Explanation

The highway shown on the figure above achieves the total travel time of 16. There are 10 cars leaving from the position *(1,2)* and $s_{1,2} = 1$; there are 3 cars leaving from the position *(1,3)* and $s_{1,3}=2$. For every other position we either have $M_{i,j}=0$ (no cars leave from that position) or $s_{i,j}=0$ (the highway passes through that position).

Sample Input 2

```
4
0   1 0 0
0   0 0 6
10  0 5 0
10 10 0 0
```

Output for Sample Input 2

```
20
```

PROBLEM M **TĀWHIRIMĀTEA** **100 POINTS**

The weather god, Tāwhirimātea, wants to reach all the islands in the ocean. Tāwhirimātea can reach an island either by being on it, or by sending one of the four winds in the appropriate direction (N/S/E/W) from one his locations. How many different locations does Tāwhirimātea need to be in so that he can reach all the islands?

Input

The input consists of fifteen lines each containing fifteen characters depicting the ocean. Each character is either a period (.), representing water, or a hash (#), representing an island that is not connected to any other island.

Output

The output consists of a single integer, the minimum number of locations from which Tāwhirimātea can reach every island depicted by a hash. An island is reachable if Tāwhirimātea is himself on the island or if it lies exactly North, South, East or West of him.

Sample Input 1

```
...............
...............
...............
...............
...............
...............
...............
.......#.......
...............
...............
...............
...............
...............
...............
...............
```

Output for Sample Input 1

```
1
```

Explanation

It is enough for Tāwhirimātea to be on the single island.

Turn over for more sample data.

Sample Input 2

```
#..............
#..#...........
.#.............
.#.#...........
..#............
..##...........
...............
...............
...............
...............
...............
...............
...............
...............
...............
```

Output for Sample Input 2

```
3
```

Explanation

If *(i,j)* means row *i* (1 is the topmost) and column *j* (1 is the leftmost), then one possibility is for Tāwhirimātea to be on positions *(5,1)*, *(4,2)*, and *(6,4)*.

PROBLEM N **KRYPTO'S CHAIN OF COMMANDS** **100 POINTS**

When Superman isn't saving the world, he trains his trusty companion, the superdog **Krypto**. But as it turns out, training a dog with Kryptonian powers is no walk in the park.

A training session consists of a sequence of **N** commands that must be followed *in order*.
Every command is represented by an integer between **1** and **MaxValue** (inclusive).

Superman has a problem though:
Krypto will only listen to a command if its number is divisible by the previously given command. That is, if the previous command is `x`, then the next command must be a multiple of `x`.

Superman has come to you, asking how many different valid training sequences there can be, such that Krypto will obey every command in the sequence.

It can be assumed that Krypto will always listen to the first command, as it has no previous command to check divisibility against.

Since the number of valid sequences can be very large, print the result modulo $10^9 + 7$.

Turn over for input and output.

Input

The input consists of a single line, with 2 integers: **N** and **MaxValue**.

N, (*1 ≤ N ≤ 10000*), representing the number of commands a sequence must have.
MaxValue, (*1 ≤ MaxValue ≤ 10000*), representing the maximum value a command can have. (Each command is an integer between **1** and **MaxValue**, inclusive.)

Output

Output is a single integer representing the number of valid sequences of commands, modulo $10^9 + 7$.

Sample Input 1	Output for Sample Input 1
`3 3`	`7`

Explanation

For Sample Input 1, the possible valid sequences of length 3 using values from 1 to 3 are:

- `[1, 1, 1]`
- `[1, 1, 2]`
- `[1, 1, 3]`
- `[1, 2, 2]`
- `[1, 3, 3]`
- `[2, 2, 2]`
- `[3, 3, 3]`

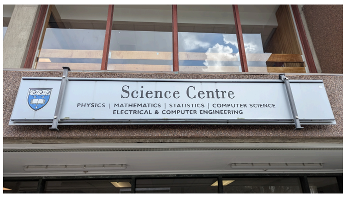
So there are **7** valid sequences in total. **7** should be the output.

A sequence like `[1, 2, 3]` is **not valid** because 3 is *not* divisible by the previous number, 2.

Sample Input 2	Output for Sample Input 2
`2 5`	`10`

Sample Input 3	Output for Sample Input 3
`5 3`	`11`

PROBLEM O **NEW SCIENCE CENTRE** **100 POINTS**



The University of Auckland is planning to replace the old swimming complex with a new Science Centre. To prepare the bulding site, the swimming complex first has to be demolished.

The swimming complex consists of N floors, and on each floor there is a swimming pool with some amount of water in it. The amount of water (in litres) in the swimming pool on the ground floor is denoted by W_0, on the first floor by W_1, on the second by W_2, and so on. Each swimming pool has a maximum capacity that determines how much water it can hold (in litres) without collapsing the floor it's on. If the capacity is exceeded and the floor collapses, all the water goes down to the pool on the floor beneath it. This may cause that floor to collapse, and so on. The pool on the i-th floor has a capacity of L_i litres.

The entire building collapses if the ground floor collapses.

A floor can collapse for two reasons: either we demolish it using explosives, or the amount of water in its pool is larger than the capacity. Demolishing the i-th floor using explosive costs C_i NZD. Your goal is to calculate the cheapest way of collapsing the ground floor - which makes the whole building collapse and the development of the new Science Center can begin!

For example, suppose there are five floors (including the ground floor) and the pool on each floor has the same capacity of 1000 litres and 400 litres of water in it. Suppose the prices of demolishing floors using explosives are $C_0 = 200$, $C_1 = 90$, $C_2 = 80$, $C_3 = 70$, $C_4 = 60$. Demolishing the ground floor using explosive costs 200NZD. If we demolish the second floor using explosive, then we pay 80NZD and all 400 litres of water go down to the first floor. The first floor now has 800 litres of water in it. If we further demolish the first floor, all 800 litres of water will go down to the ground floor, which then collapses because it has in total 1200 litres of water but the capacity is only 1000 litres. Demolishing the first floor using explosive costs us 90NZD, which in total sums up to 90 + 80 = 170NZD, which is cheaper than blowing up the ground floor. But there is an even cheaper way: demolish floors 3 and 4. This costs us 70 + 60 = 130NZD. The pool on floor 2 now has 1200 litres of water in it, thus it collapses. All this water goes down to floor 1, and then the ground floor. This turns out to be the cheapest way.

Turn over for input and output.

Input

The first line of the input contains one integer N ($1 \le N \le 200000$), the number of floors. Then N lines follow, where the i-th line contains three integers W_i, L_i, and C_i, which are the amount of water in the pool on the i-th floor, the capacity of the pool on the i-th floor, and the cost of demolishing it using explosives, respectively.

It is guaranteed that $0 \le W_i \le 500$, $1 \le L_i \le 20000000$,$C_i \le 300000000$ and $W_i \le L_i$.

Output

Output a single integer which denotes the cheapest possible price for demolishing the building.

Sample Input 1

```
5
400 1000 200
400 1000  90
400 1000  80
400 1000  70
400 1000  60
```

Output for Sample Input 1

```
130
```

Explanation

This is the example from the description of the problem.

Sample Input 2

```
8
0   40  100
8   9   9
7   8   8
6   7   7
5   6   6
9   100 5
9   10  15
100 101 100
```

Output for Sample Input 2

```
20
```

Explanation

Demolish floors 5 and 6 (recall that floors are indexed from 0).

PROBLEM P **ROADSIDE PRODUCE** **100 POINTS**

Every day you travel from Te Anau to Timaru and prepare your famous layered salad en route. On each road connecting two towns there is a roadside stall selling one type of produce, which you must always buy and immediately add to your salad. This being a layered salad, it is paramount that you buy and add the ingredients in the precise order that the recipe specifies.

During your long and healthy life you would like to visit many towns. How many different towns will you be able to visit if you adhere to this plan?

Every day you must prepare a full and ordered salad, but you do not need to take the same route every day. You can pass through the same town or road more than once on a given day. Each distinct town is only counted once, even if you visit it multiple times in the same day or on different days.

Input

The first input line consists of two integers n, m ($2 \le n \le 100$) representing the number of towns and the number of roads. Te Anau is town 0, and Timaru is town $n - 1$. The following m lines consist of a triple of integers x, y, z indicating that there is a road between x and y with a stall selling produce z ($0 \le x, y \le n$, $1 \le z \le 100000$). There is at most one road between any two towns. The next line consists of an integer t ($0 \le t \le 100$) representing the number of ingredients. The next line consists of t integers, the list of ingredients.

Output

The output must consist of a single integer, the maximum number of different towns that can be visited.

Sample Input 1

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

Output for Sample Input 1

```
4
```

Explanation

On day 1, you first travel to town 1, then back to Te Anau, and finally to Timaru. On day 2, you first travel to town 2, then back to Te Anau, and finally to Timaru.

Turn over for more sample input/output.

Sample Input 2

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

Output for Sample Input 2

```
0
```

Explanation

You can only travel to town 1 but not reach Timaru.