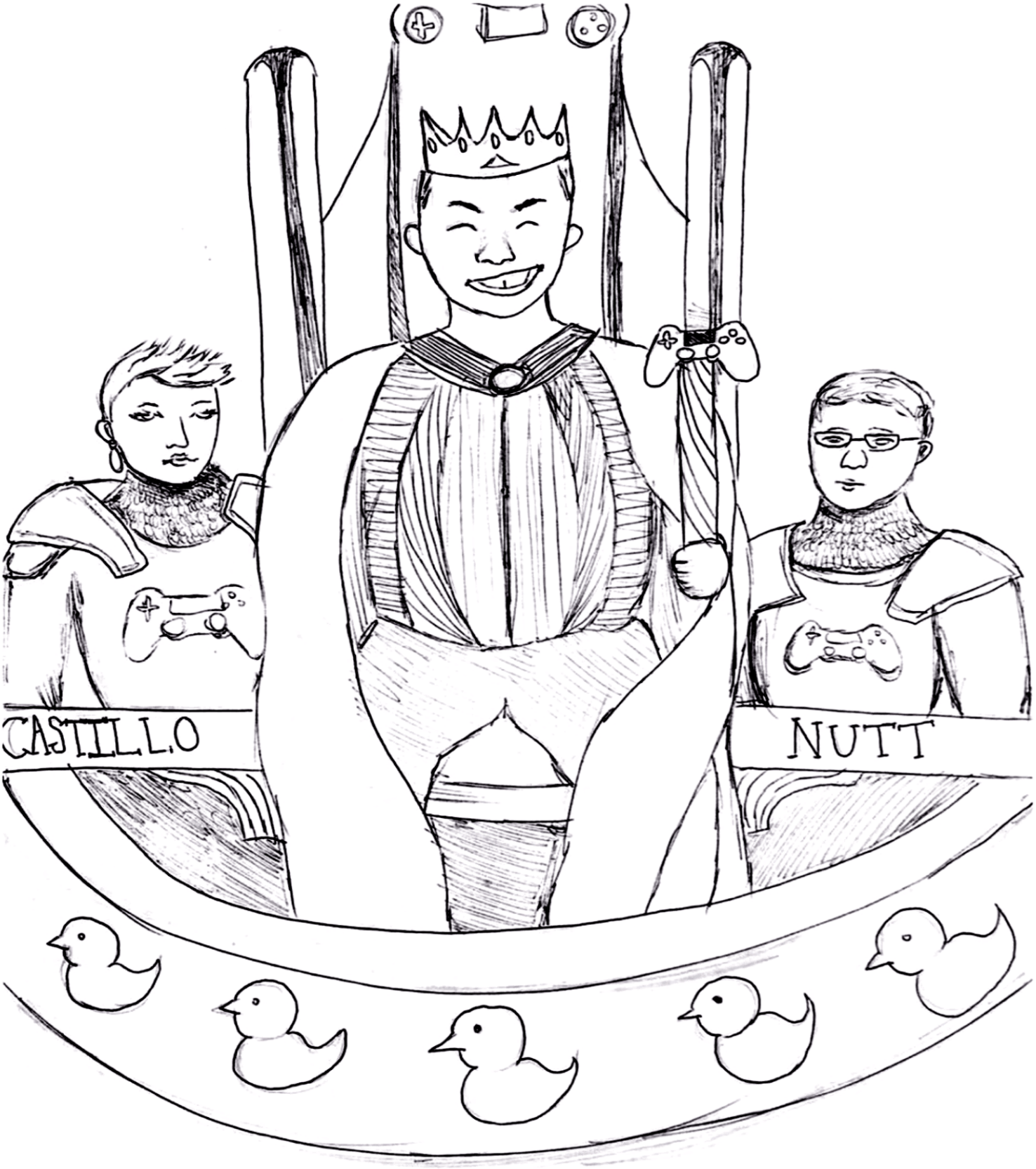
Ye Old Kingdom of Stroudonia



Seven Lakes HS Kickoff 2022

Stroud's Throne

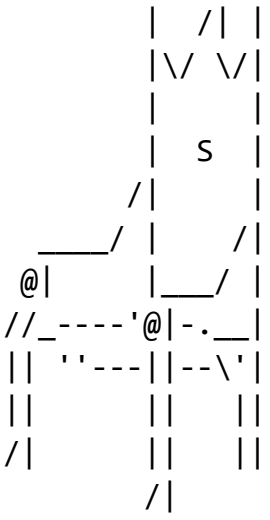
Problem Description:
Getting tired of his old, rickety lounger, King Stroud decides to order a new chair from Medieval Ikea. However, the chair came unbuilt! Not used to the hard labor, he forces one of his servants to construct it.

Construct Stroud's throne so he can ~~sleep~~ sit peacefully.

Input Description:
No input.

Output Description:
Output the following ASCII art:

Sample Output:



Counterfeit Catchers

Problem Description:
Stroud's expert coinmaster will determine from a stack of coins which one is different from the others. There is guaranteed to be one and only one different coin.

Input Description:
The first line will have **N**, the number of test cases. For each test case, the first line will have **C** the number of coins in the stack and **L**, the length of each coin design in lines. The next **C * L** lines will have all of the coin designs.

Output Description:
Output the counterfeit coin as shown below.

Sample Input:

```
1
3 5
ccccc
c   c
c e c
c   c
ccccc
ccccc
c   c
c e c
c   c
ccccc
cccoc
c   c
c e c
c   c
ccccc
```

Counterfeit Catchers (Cont.)

Sample Output:

```
Sir, coin 3 is a counterfeit!
```

Sample Input:

```
1
14 1
l
l
l
l
l
l
l
l
l
l
l
l
I
l
```

Sample Output:

```
Sir, coin 13 is a counterfeit!
```

The Wall

Problem Description:
To protect the North side of his kingdom from intruders, King Stroud has devised the construction of an enormous wall, which will take great quantities of resources. Being a strategic master, Stroud knows exactly what areas of his border need extra protection. To build the walls, Stroud will hire different teams of people to build each segment of the wall. For each team, the cost to increase the length of the wall by 1 is **A** Stroud Bucks, and starting a new segment of the wall, thus hiring a new team, costs **B** Stroud Bucks. Help Stroud minimize the amount of Stroud Bucks he needs to spend to protect all the needed regions.

Input Description:
The first line will have N, the number of test cases. The first line of each test case will have **L, A**, and **B** the length of the border to protect, and the two costs mentioned previously. The next line will be a string of length L, with an "X" representing a position that needs protection and a "." representing a position that doesn't.

Output Description:
Output the minimum cost to protect all needed regions

Sample Input 1:
```
1
20 1 5
X.X.X.X.XX......XX.
```

Sample Input 2:
```
1
20 5 1
X.X.X.X.XX......XX.
```

Sample Output 1:
```
17
```

Sample Output 2:
```
46
```

Qualification Queries

Problem Description:
Stroud has lined up all of the applicants to his inner legion of knights in nice rows and columns. Each knight has an assigned skill level based on their battle experience and other expertise. To pick his inner legion, Stroud simply wants to pick a sub-rectangle, but it must be only made up of knights above a certain skill level. Having tried the random queries method, Stroud seems to want a more methodical technique. Now, he will pick the largest possible sub-rectangle.

Input Description:
The first line will have N, the number of test cases to follow. For each test case, the first line will have **R**, **C**, and **S**, which are the number of rows and columns that the knights are formed in, and the minimum skill to join the legion. The next **R** lines will have **C** integers denoting the skill level.

Output Description:
Output the largest number of knights Stroud can let in.

Sample Input:
```
1
4 4 10
5 6 10 6
4 10 10 20
7 15 12 67
78 80 4 7
```

Sample Output:
```
6
```

Seven Stroud Army

Problem Description:
Owen, Andy, and Aayush are planning to build an army from the villagers that live in the castletown. However, after receiving a list of all the villagers, the group realizes they cannot take everyone into the army, like old and young people, storeowners, and secret spies.

Input Description:
The first line will contain number **N** dictating the number of data sets to follow.
The first number **V** of a data set will inform you how many villagers are to follow.
The following **V** lines will contain the first and last name of the villager, their occupation, and their age.

Conditions:
-Spies have "Stroud" within their first or last name (any case). Do not include them in the army.
-The best age for a soldier is `25`. The closer a person is to that age, the more fit they are to join the army. A younger soldier is more fitting that an older soldier with the same age difference to `25`.
-Do not let anybody below `21` or above `60` into the army.
-Only let peasants join the army.
-If there are two people of equal fitness, output the first person sorted alphabetically by their last name.

Output Description:
Output the best candidate from each group to add to the army.
If there are no fitting candidates in the group, output nothing.

Seven Stroud Army (Cont.)

Sample Input:

```
2
7
Kevin Staroud Peasant 32
Lauren Staruod Peasant 28
Bob Tulip Cobbler 35
Stroud Tulip Peasant 25
Nighting Gale Peasant 17
Potato Bob Peasant 22
Carrot Ale Peasant 22
2
Stroud Yellow Peasant 25
Yellow StrOud Peasant 25
```

Sample Output:

```
Carrot Ale
```

Jester's Jokes

Problem Description:
King Stroud likes to host elaborate feasts at his castle, which often feature stand-up comedy from the official court Jester. The Jester has a special position because he's the only one allowed to make jokes about the King! If any other of the feast attendees joke about the King, they get sent to the ~~Dungeon~~ Fungeon.

Naturally, in all the discussion, King Stroud doesn't hear every single joke at the feast, and His Highness doesn't have the time to review the official transcript by the royal reporter, so he's hired you to do it for him.

Given the transcript of jokes at the feast, figure out who needs to be sent to the Fungeon. Anybody who makes a joke, excluding the Jester, that includes the word "Stroud" gets added to the Fungeon list. "Stroud" can appear with any capitalization, in any part of a word (i.e., mention of "Stroudonia" counts as a mention of "Stroud").

Input Description:
You will receive a transcript of jokes. The first line contains an integer **N**, the number of entries. Each of the entries will begin with the speaker's name followed by a colon. The following unknown number of lines will be the speaker's jokes. An empty line will mark the end of an entry.

Output Description:
Output the "Fungeon List:" followed by the names of people to send to the Fungeon on each line. Print the names in chronological order with respect to when they said an illegal joke.

Jester's Jokes (Cont.)

Sample Input:

```
5
Sir McMurry:
We argued all day about what to call a medieval soldier.
But it was getting late so we decided to call it a knight.

Sir Nutt:
Why does Stroud have a roundtable in his court? So that none of his
other knights can corner him!

Jester:
What is the rain's favorite medieval reenactment? Storming a castle.
Look! Stroud has his shoes on backwards!

Sir McMurry:
What did Stroud say when he heard that the peasants were revolting?
He said he agrees because they never bathe and always stink.

Empress Castillo:
What do medieval postmen wear? Chainmail.
```

Sample Output:

```
Fungeon List:
Sir Nutt
Sir McMurry
```

Animal Farming

Problem Description:
Stroud's kingdom has amassed quite an impressive collection of farm animals, so he decides it's time to start building pens for them.

Each pen is a quadrilateral enclosure, but Stroud is quite picky with which animals go into which pens. Sheep must go into equilateral pens, cattle must go into non-equilateral parallelogram pens, and pigs must go in all other pens. Help Stroud figure out which animals to put in each of his pens!

Input Description:
The first line will contain a single integer **N**, representing the number of test cases to follow. Each test case will contain 4 positive integers, the length of each side of the pen given in clockwise order.

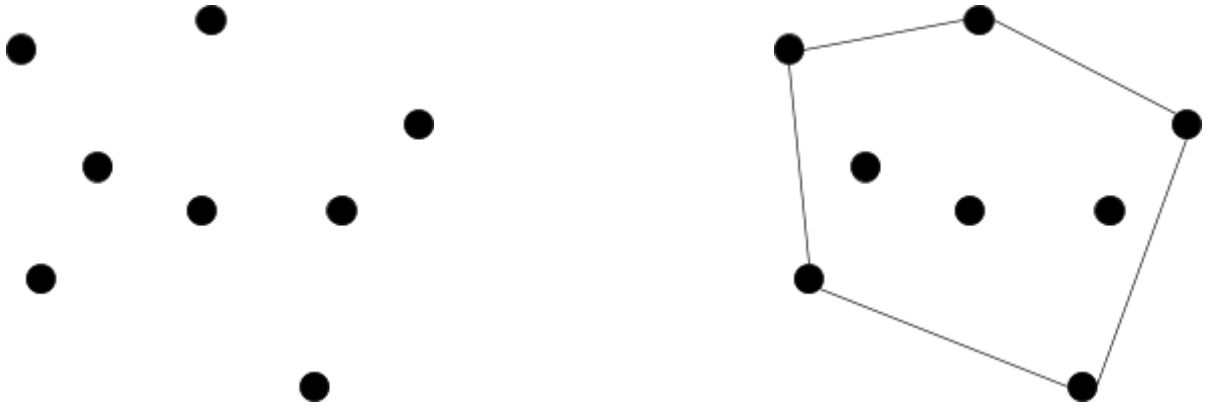
Output Description:
Output the animal that will go in each pen on each line.

Sample Input:
```
4
4 4 4 4
1 2 3 4
10 20 10 20
10 10 20 20
```

Sample Output:
```
sheep
pigs
cattle
pigs
```

Castillo's Castillo

Problem Description:
Empress Castillo was watching the success of Stroudonia and decided she wanted to try building a castle of her own: Castillo's Castillo! She already has some towers set up, but she needs help building the outer walls. However, before she can do so, she needs to know the minimum length of wall she needs to import from Stroudonia to enclose all the towers.



Input Description:
The first line of each test case will contain a single integer **N**, representing the number of test cases to follow. The first line of each test case will contain a single integer **M**. The next **M** lines will contain two integers, x and y, representing the location of a tower.

Output Description:
For each test case, output the length of the outer walls, rounded to 2 decimal places.

Castillo's Castillo (Cont.)

Sample Input:

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

Sample Output:

```
19.36
4.00
```

Knight Initiation

Problem Description
The knight initiation process is very selective. Only the best knights can serve for Stroud. Unfortunately, Stroud cannot select too many of his friends or else the public will think he is biased. The next quest requires the sum of all strength values of his knights to be at least **X**. Find the minimum number of friends Stroud can select such that the total strength value of his troops is at least **X**.

Input Description
The first line will contain T, the number of test cases.

Each test case will contain 3 lines:
First line: 2 integers **N**, the number of knights, and **X** the minimum strength value
Second line : **N** integers denoting the strength of each Knight
Third Line : **N** integers one or zero (one denoting that this knight is Stroud's friend)

Output Description
The minimum number of friends Stroud can select.

Sample Input
```
1
5 10
2 4 8 1 1
0 1 1 0 0
```

Sample Output
```
1
```

Explanation
Stroud can select knights 1 and 3 to select only 1 friend.

Royal Sacrifice

Problem Description:
Lately, the population of Ye Old Kingdom of Stroudonia has been growing, but the productivity hasn't (meaning there are more mouths to feed, but the same amount of food). To cut back on the debt accruing from food imports, King Stroud has decided to send some of the peasants to distant, far away land...

Please organize the peasants by their last name and help King Stroud find which consecutive group of peasants with the least productivity to send off. The group must be the exact size that King Stroud is looking for.

To determine the least productive group, each peasant is marked with **X**, their productivity. The group that has the smallest sum of **X** is the least productive. While organizing the peasants, if two peasants have the same last name, put the peasant with the lower **X** value first.

Input Description:
The first line will consist of **C**, the number of test cases to follow. The first line of each test case will consist of two numbers, **N**, representing the total number of peasants, and **M**, representing the number of peasants King Stroud wants to send off. Each of the next **N** lines for each test case contains the last name of a peasant, followed by their **X** value.

Output Description:
Output a single integer, the sum of the **X** values of the consecutive peasants with the least productivity.

Sample Input:
```
1
6 3
Potter 2
Skywalker 7
Fett 8
Rodgers -5
Parker -1
Stark 9
```

Sample Output:
```
-4
```

Royal Coffers

Problem Description:

King Stroud needs his minions to transfer coins from his endless royal coffers to his throne so he can pay for his upcoming board game spending spree. His minions are very precise: each can run from the throne room to the coffers, fill its barrel, and deliver the coins in a specific amount of time. Each minion carries a coin barrel of a specific capacity, which it always fills to the max capacity on every trip. King Stroud does not want to overwork his minions, so he would like your help to find the least amount of time he needs to meet his coin target.

Input Description:

The first line will consist of integer **N**, the number of minions he has to work with, and long **C**, the number of coins that King Stroud requires for his board game spending spree. The next **N** lines each consist of integers **S**, **B**, and **T**, the start time, barrel size, and coin collection time of each minion respectively.

Output Description:

Output the minimum amount of time required for the number of coins in the throne room to meet or exceed the target amount of coins.

Sample Input:

```
3 6
0 2 3
1 1 1
2 2 20
```

Sample Output:

```
5
```

Explanation:

	Total:
At **t=0**, the **1st** minion has started collecting and delivering coins.	**0 Coins**
At **t=1**, the **2nd** minion has started collecting and delivering coins.	**0 Coins**
At **t=2**, the **3rd** minion has started collecting and delivering coins.	**0 Coins**
At **t=2**, the **2nd** minion delivered 1 coin to the throne room.	**1 Coin**
At **t=3**, the **1st** minion delivered 2 and the **2nd** minion delivered 1.	**4 Coins**
At **t=4**, the **2nd** minion delivered 1.	**5 Coins**
At **t=5**, the **2nd** minion delivered 1.	**6 Coins**
(target reached, print out 5).	

Uniform Milk Production

Problem Description

King Stroud recently opened a goat cheese factory. His N goats each can produce milk at rates $A_1, A_2, \ldots, A_n$ respectively with $1 \leq A_i \leq 10^2$ (in milliliters per hour.) Because Stroud boasts about having the most consistent cheese in the world, he wants the goats to each have a similar milk output, so the composition of milk is as similar as possible for each goat (if a goat produces too little milk compared with its peers, its milk will have a richer consistency.) However, King Stroud is suspicious that one of his goats is either not keeping up with or outpacing the rest of the trip (group of goats).

In order to find the goat that does not fit, he lines up all of his goats, and looks at each group of K ($2 \leq K \leq N$) consecutive goats. He wants to find a goat such that it maximizes the absolute difference between the average milk outputs of the K consecutive goats and the average milk outputs of the same $K - 1$ goats without it.

Input Description

The first line describes the number of test cases, T. The first line of each test case describes N ($1 \leq N \leq 50$) and K ($1 \leq K \leq N$). The next line describes N integers, each separated by a spaced, representing A.

Output Description

Find the goat that doesn't fit and output its index (if multiple goats tie, return the earlier index). Note that this problem is 1-indexed (index of first goat starts at one.)

Sample Input

```
1
7 3
6 1 4 2 10 10 3
```

Sample Output

```
4
```

Explanation

It can be shown that $|avg(\{2, 10, 10\}) - avg(\{10, 10\})| = |\frac{22}{3} - \frac{20}{2}| = 8/3$ is the largest absolute difference possible. Thus, the 4th goat (which has a rate of 2 ml/hr) should be removed.

Territorial Disputes

Problem Description

After emerging from the Dark Ages, Stroudonia and its neighbors have tried to claim territory and colonize Aruba. Each piece of territory is continuous and can be reached by any other part of that territory by moving north, south, east, or west one unit T times without crossing into unclaimed land. A ". " symbol (without quotes) represents claimed territory while "#" (without quotes) represents unclaimed land. Each kingdom rules its own piece of territory, which is separated from the others by unclaimed land.

Now, the colonizers have come to terms with a new agreement in hopes of gaining even more land. For each unit of unclaimed land, that unit will belong to the kingdom with the closest ORIGINAL territory with the distance measured by Manhattan distance. If two (or more) kingdoms tie, that piece of land will be counted as disputed territory for all kingdoms involved, with each kingdom still filing a claim for that land. Otherwise, it will count as undisputed territory for the closest kingdom.

Note: the Manhattan distance between two pieces of land with locations at (r_1, c_1) and (r_2, c_2) is described as $|r1 - r2| + |c1 - c2|$ where $|x|$ denotes the absolute value of x.

Input Description

The first line contains n and m, which denote the height and the width of Aruba. The next n lines will each contain m characters, with each character being either a "#" or a ". ".

Output Description

On the first line, output the total amount of undisputed territory.

Sample Input

```
4 6
#..#.#
...#.#
.###..
##....
```

Sample Output

```
21
```

Territorial Disputes (Cont.)

Explanation

Consider the territories as belonging to the kingdom numbered 1 and 2. Then, the map shall be equivalent to:

```
#11#2#
111#2#
1###22
##2222
```

After claiming new territory, the map becomes like this with "D" representing disputed territory between kingdoms 1 and 2.

```
111D22
111D22
11D222
122222
```

Here, there are only 3 units of disputed territory, so there will be 21 units of undisputed territory.

Census Statistics Outliers

Problem Description

King Stroud has decided that it is time to conduct his first census of Stroudonia. Because Stroudonia has grown to an unbelievable size, he has divided it into districts. Now that he has the population of each district, he wants to know which ones are outliers. An outlier is defined as anything less than $Q1 - 1.5 \cdot (Q3 - Q1)$ or anything greater than $Q3 + 1.5 \cdot (Q3 - Q3)$. Given the populations of each district in non-descending order and $Q1$ and $Q3$, find all the outliers in non-descending order.

Input Description

The first line contains integer T, the number of test cases to follow. The first line of each test case contains integers N, the number of districts, and $Q1$ and $Q3$. The next line contains N spaced integers indicating the populations of each district in non-descending order. It is guaranteed that $Q1$ and $Q3$ are the actual first quartile and third quartile of the populations.

Output Description

Output the outliers in non-descending order (descending order, allowing repeats), each separated by a space.

Sample Input

```
1
10 14 17
2 2 14 15 15 16 17 17 30 31
```

Sample Output

```
2 2 30 31
```

Explanation

The interquartile range or $Q3 - Q1$ is $17 - 14 = 3$. The twos are less than $14 - 1.5 \cdot 3$ and 30 and 31 are greater than $17 + 1.5 \cdot 3$.

Flaming Flamingo Fighting Force

Problem description

In a recent invasion, King Stroud's army got stuck in the swamps they were trying to pass through, resulting in a defeat. To pass the swamps, King Stroud has decided to create a new army—an army of Flamingos. Flamingos have long legs used to tread in deep water without getting wet. With that being said, flamingo legs are only so long, and if the water is too deep, they will get wet.

The Flamingo army needs to cross N swamps. Each lake has different depths for each meter going across it, and the depths for each meter can be represented as a sequence of integers. For instance the sequence `[1, 2, 3]` shows depth `1` for the first meter, depth `2` for the second meter, and depth `3` for the third meter. The Flamingos in the army have legs length of L feet, and they are only comfortable being wet for K meters per swamp. Being wet means that at an index in the swamp, the water's depth is greater than the leg length of the Flamingos.

Input description:

The first line contains *N*, the number of test cases.
The first line of each test case contains the length M ($1 \leq M \leq 1000$) of the current swamp in meters, the length of the flamingo's legs L, in meters, and the number of meters the flamingos are willing to be wet for, K.
The second line of each test case contains M space-separated integers representing the depth of the swamp at each meter. The depth will be positive and will not exceed `500`.

Output description

For each test case, output "Yes" if the army can cross the swamp or "No" if they cannot.

Sample Input:

```
3
3 4 2
4 4 4
5 3 2
1 7 2 5 3
6 5 3
7 9 3 8 5 6
```

Sample Output:

```
Yes
Yes
No
```

Knight Photo

Problem Description

King Stroud wants to take a photo with **N** knights. The ith knight is described as (h_i, w_i) where h_i is the height and w_i is the width of knight i. King Stroud wants to know for each knight i, if it is possible for another knight to stand in front of it. Knight j can only stand in front of knight i if and only if $h_i > h_j$ and $w_i > w_j$. Additionally, knights have the ability to swap their height and width if it makes it possible for another knight to stand in front of it. For example:

```
Knight i: (4, 5)
Knight j: (4, 3)
```

Swap height and width in knight `j:(4, 3) -> (3, 4).`
Now, knight j can stand in front of knight i.

Input Description

The first line contains **N**, the number of knights. The next **N** lines contain h_i and w_i, the height and width of the ith knight.

Output Description

Output **N** lines where the ith line contains "YES" or "NO" depending on if a knight can stand in front of knight i.

Sample Input

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

Sample Output

```
YES
NO
YES
```

Explanation

Knight 1 can have knight 2 stand in front of it given that we make `(1, 3)-> (3, 1)`. Knight 2 cannot not have any knights stand in front of it. Knight 3 can have knights 1 and 2 stand in front of it.

Water Collection

Problem Description:
After being hit with a large drought, the kingdom of Stroudonia is in dire need of a water source. Fortunately for them, they have recently discovered an aquifer right underneath a large, dried lake. The aquifer is able to push water up to the surface from deep underground, but only with enough pressure to be level with the lake's highest edge. The kingdom, however, wants to calculate the amount of water that can be trapped by the lake using its topography map. Having been given an array of heights, calculate the units of water the lake can collect when water rises from the bottom.

Input Description:
The first line will contain an integer **N** for the number of heights provided by the topography map. The next **N** lines will contain an integer indicating the height at that point. There will only be one `0` in the **N** lines displayed representing the point from where the water springs up from into the lake.

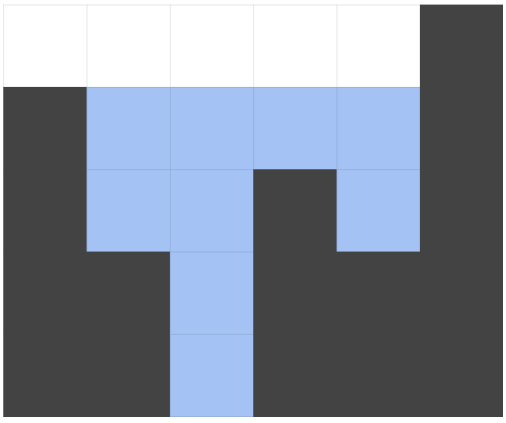
Output Description:
The output is a single integer indicating the total amount of water that is able to be trapped.

Sample Input:
```
6
4
2
0
3
2
5
```

Sample Output:
```
9
```

Clarification/Diagram:
The black indicates the values of the topographical heights.

The blue indicates the places where water can be stored.



Money... Money... Money!

Problem Description:
Stroudonia has a huge problem! The treasury, due to their lack of funds (surprising!), has shut down. However, the various money loans that will be given to other countries still have to go to them in order to help them recover. Without these loans, Stroudonia will not stand against the Huns! As the country's top advisor, you must help Stroudonia!

Input Description:
Given a list of **N** integers, find the amount of bills of each denomination.

Output Description:
A list of bills.

Bill Denominations:
`1000, 100, 50, 20, 10, 5, 1`

Sample Input:
```
3
10927
2342
12
```

Sample Output:
```
10:9:0:1:0:1:2
2:3:0:2:0:0:2
0:0:0:0:1:0:2
```

Explanation:
Take the third case. Since there is only one ten dollar bill and two one dollar bills, only a `1` and `2` will be printed in their respective places, with `0`'s filling in the rest.