



Clements High School Computer Science Invitational

Novice Division

November 19, 2022



Tean Number:	School:		
<h1>Problem Checklist</h1>			
Each problem is worth 60 points. Every incorrect submission is -5 points if the problem is eventually answered correctly.			
Problem Number	Problem Name	Tally of Incorrect Submissions	Score
1	Turkey Ranger		
2	Potato Problems		
3	Weights		
4	Funky Editor		
5	Dinner Deal		
6	Amazing Neighborhood		
7	Manufacturing Plant		
8	Crafting		
9	Turkey Rebellion		
10	Turkey Escape		
11	Store		
12	Turkey Baking		
13	Turkey Race		
14	Turkey Beef		
15	Flying Turkeys		
		Total:	

Problem Number	Problem Name	Tally of Incorrect Submissions	Score
1	Turkey Ranger		
2	Potato Problems		
3	Weights		
4	Funky Editor		
5	Dinner Deal		
6	Amazing Neighborhood		
7	Manufacturing Plant		
8	Crafting		
9	Turkey Rebellion		
10	Turkey Escape		
11	Store		
12	Turkey Baking		
13	Turkey Race		
14	Turkey Beef		
15	Flying Turkeys		
		Total:	

1. Turkey Ranger

Program name: turkeyranger.java

Input file: N/A

The Clements Computer Science club wants to host a Thanksgiving party which includes a roasted turkey. Unfortunately, due to dietary restrictions, the club was unable to serve the dish, so being the geeks they are, the club wants everyone to create the food items using ASCII art. You have been assigned to create the turkey. Help liven the Thanksgiving spirit at the Thanksgiving party by drawing a turkey using ASCII art!

Input

(none)

Output

Print the turkey exactly as shown in the sample input

Sample Input

N/A

Sample Output

```
++  
<(o ) _/_ \ \/  
U\ \/\ \ \ )  
| > | /  
 \____/_/_/  
 | |
```

2. Potato Problems

Program name: **potatoes.java**

Input file:
potatoes.in

Rich is hosting a Thanksgiving party at his house this year. He promised everyone that he would make his world-famous mashed potatoes, but he procrastinated on his grocery shopping and there were only a few bags of potatoes left at the store. Each bag of potatoes can serve **up to 2 people**. Given the number of bags that the store has left, **n**, Rich needs your help in determining whether he will have enough servings for the number of guests, **g**.

Input

The first and only line of input contains a positive integer **n**, the number of bags of potatoes, and a positive integer **g**, the number of guests.

Output

Print **YES** if Rich can make enough servings of mashed potatoes and **NO** otherwise.

Sample Input 1

13 34

Sample Output

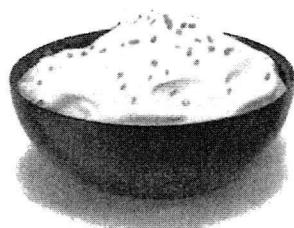
NO

Sample Input 2

25 40

Sample Output

YES



3. Weights

Program name: **weights.java**

Input file:
weights.in

On a nice fall day, two friends, Justin and Lai want to buy a turkey to celebrate Thanksgiving. Oddly enough, both of them wanted their portion of the turkey to weigh an even amount of pounds. Help the friends decipher if the following turkey weights allow for their demands to be met!

Input

The first line contains a single integer, **N**, the number of turkeys. The next **N** lines each contain one integer representing the turkey's weight.

Output

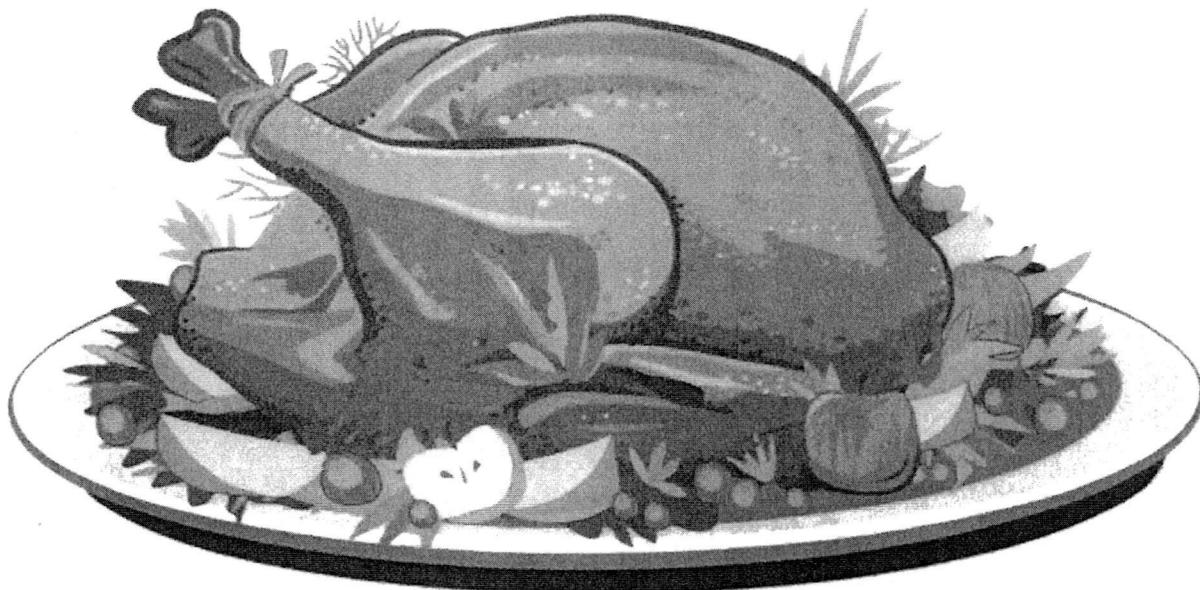
Print out "YES" if it's possible to split the turkey's weight so that both portions are represented by even integers and "NO" otherwise.

Sample Input

```
2
10
5
```

Sample Output

```
YES
NO
```



4. Funky Editor

Program name: **funkyeditor.java**

Input file: **funkyeditor.in**

Endlec is drafting invitations to his annual Thanksgiving dinner party! However, his text editor is acting up by adding extraneous spaces in between each word. Can you help Endlec reformat his invitations so that each word is only separated by one space?

Input

Each line contains space-separated string tokens, representing a single sentence. It is guaranteed that each word is separated by at least 1 space and that there are no leading or trailing whitespaces.

Output

For each sentence, print the formatted string in which each word is only separated by 1 space. Make sure that there is not an extraneous trailing space after the final word of each message.

Sample Input

I am buying lots of Turkey!
I will have too much to eat by myself.
There will also be mashed potatoes.
Hope to see you there!

Sample Output

I am buying lots of Turkey!
I will have too much to eat by myself.
There will also be mashed potatoes.
Hope to see you there!



5. Dinner Deal

Program name: **dinnerdeal.java**

Input file: **dinnerdeal.in**

Sarah is going grocery shopping for Thanksgiving dinner. Help Sarah find the total cost of her shopping trip.

Because it is Thanksgiving, there is a 10% discount on every item with the term "turkey" in its name. For every item in Sarah's grocery list, apply a 10% discount to every item with the term "turkey" (case sensitive) in its name. Find the total price of the items in Sarah's grocery list with the applied discounts. Round your answer to two decimal places.

Input

The first line will contain integer **n**, the number of items on her list.

The following **n** lines will contain double **A** and string **B**, which represents the price of the item and the name of the item Sarah is trying to buy. It is guaranteed that string **B** doesn't contain spaces.

Output

Print out the price of Sarah's grocery list as a double rounded to two decimal places. Do not include the dollar sign in your answer.

Sample Input

2
15.98 Roastturkey
18.08 Turkeymeatloaf

Sample Output

32.46

Explanation:

"Roastturkey" contains the term "turkey", so it receives the 10% discount. However, "Turkeymeatloaf" does not receive the discount because it does not contain the term "turkey".



6. The Amazing Neighborhood

Program name: neighborhood.java

Input file: neighborhood.in

The Amazing Neighborhood is having a neighborhood wide Thanksgiving party. However, no one wants to host the party, because being a party host is a lot of work. To resolve this conundrum, the Amazing Neighborhood decides to randomly select one name from each household and decide the party host based on which name is the longest. If multiple names are the same length, the party host is decided by whoever's name comes first alphabetically.. Given a list of names selected by the Amazing Neighborhood, determine which person's family will be hosting the Thanksgiving party.

Input

The first line of input contains a positive integer **N** ($1 \leq N \leq 20$), the number of names. The next **N** lines each contain a name from the Amazing Neighborhood.

Output

Print a single string containing the name of the person whose family will be hosting the Thanksgiving party.

Sample Input

```
3
Sally
Bobbina
Rhea
```

Sample Output

```
Bobbina
```

Explanation:

The three names Sally, Bobbina, and Rhea contain 5, 7, and 4 characters respectively, so Bobbina is the longest name.

7. Manufacturing Plant

Program name: manufacturing.java
manufacturing.in

Input file:

Farmer John is extremely stressed this Thanksgiving because of the unfathomable amount of corn he needs to produce for this year's harvest. Luckily, he has managed to lease a local, genetically-supercharged corn manufacturing plant that produces ears of corn in batches of size b ($1 < b < 10^7$). The plant takes t ($1 < t < 10^7$) minutes to crank out each batch. Given the number of ears of corn needed, c ($1 < n < 10^7$), please calculate the time required to produce all the corn.

Input

The first line contains integer n , ($0 < c < 10^7$), the number of cases
All following lines contain $b\ t\ c$ in that order.

Output

Please print the number of minutes needed to fulfill each test case on a separate line. Note: you may need to use *long* to store your final answer.

Sample Input

3
3 5 9
2 7 1
1 1 9

Sample Output

15
7
9

Explanation:

For the first test case, the production of 9 ears of corn in batches of 3 takes 5 minutes per batch, so it will need 15 minutes total

For the second test case, the production of one ear of corn in batches of two takes less time than that needed for a full batch; but because the production of each batch takes 7 minutes, at least one batch must be produced, resulting in a total time of 7 minutes to produce.

Finally, the last case requires 9 batches, each needing one minute to produce

8. Crafting

Program name: **crafting.java**

crafting.in

Input file:

Instead of spending time with his family, a man by the name of Jonathan is playing a future version of Minecraft. In this game, you can craft a knife with 2 sticks and 4 stones. Additionally, you can craft a pickaxe with 4 sticks and 2 stones.

Each of these crafted items can later be traded into the shop for 5 credits. If Jonathan has x sticks and y stones, what is the maximum number of credits he can end up with?

Input

The first line contains a single integer, N , the number of cases. The next N lines each contain the integers x and y denoting the number of sticks and stones, respectively, that Jonathan has.

Output

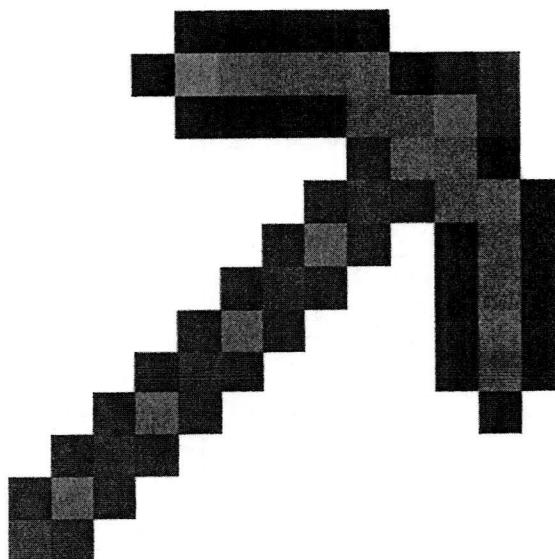
Print out the maximum number of credits that Jonathan can obtain in each case if each knife or pickaxe is sold for 5 credits.

Sample Input

```
2
40 0
10 8
```

Sample Output

```
0
15
```



9. Turkey Rebellion

Program name: rebellion.java

Input file:
rebellion.in

The turkeys are trying to take a stand this year and rebel against the cruelty of Thanksgiving. However, their communications were discovered by the humans, who then encrypted all of the turkeys' messages. Every letter has been changed into a number corresponding to its alphabetical position. (1=a, 2=b, 3=c, etc.) The turkeys need your help deciphering the entries to coordinate their attack.

Input

The first line contains a positive integer **T**, the number of entries sent between the turkeys.

Each entry contains two lines. The first line contains integer **n**, the number of characters in the message. (**Hint:** The message can contain 0 characters) The second line contains the encrypted message composed of a series of **n** positive integers (0-26).

Note: Spaces in the message will be denoted by a 0.

Output

For each of the **T** messages, output the deciphered messages, one message per line , all in lowercase.

Sample Input

```
3
12
23 5 0 1 18 5 0 18 5 1 4 25
9
15 11 0 21 19 0 20 15 15
10
1 20 20 1 3 11 0 14 15 23
```

Sample Output

```
we are ready
ok us too
attack now
```

10. Turkey Escape

Program name: **turkeyescape.java**

Input file: **turkeyescape.in**

Tyler the Turkey, wanting to escape becoming Thanksgiving dinner, ran away from his farm. Help him find a safe place to live.

Tyler's farmer has set up traps in places that runaway turkeys could go to. If the x-value or the y-value of a position is divisible by 13, the spot is not safe. In addition, if the sum of the x-value and the y-value of a position is 13, the spot is not safe.

In order to avoid being recaptured, Tyler wishes to be as far from his farm as possible. Assuming Tyler starts at the position (0,0), find the distance Tyler needs to travel to reach the furthest safe spot from the farm if he travels in a straight line.

Input

The first line will contain integer **n**, the number of positions Tyler could go to.

The following **n** lines will contain two integers, **x** and **y**, which represent the position of every possible place Tyler could run to.

Output

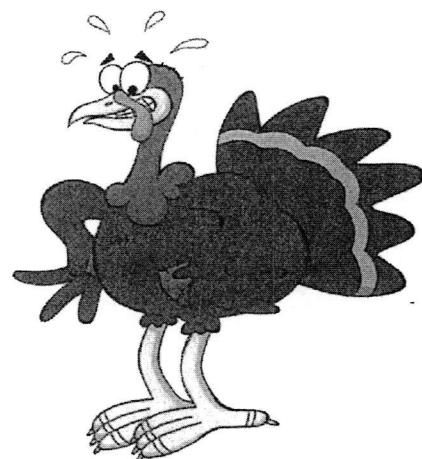
Output a single positive integer representing the distance Tyler needs to travel to reach the safe spot furthest from the farm rounded to the nearest integer.

Sample Input

```
4
3 4
13 7
1 12
5 12
```

Sample Output

```
13
```



Explanation:

The only given spots that are safe are (3, 4) and (5, 12): the x-value of (13, 7) is 13 and the sum of the x and y-value of (1, 12) is 13, making these positions unsafe.

The distance between (0, 0) and (5, 12) is 13, which is greater than 5, the distance between (0, 0) and (3, 4).

11. Store

Program name: **store.java**

Input file: **store.in**

Endlec is at the local store, buying turkeys. But for some odd reason, the prices are listed in Roman Numerals! Can you help him convert the turkey prices into integer prices?

Roman numerals are represented by 7 different symbols:

I - 1
V - 5
X - 10
L - 50
C - 100
D - 500
M - 1000

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

- I can be placed before V (5) and X (10) to make 4 and 9.
- X can be placed before L (50) and C (100) to make 40 and 90.
- C can be placed before D (500) and M (1000) to make 400 and 900.

Input

The first line contains a single integer, **N**, the number of turkeys. The next **N** lines contain one string representing the price of a single turkey in Roman Numerals. It is guaranteed that each string represents a valid Roman Numeral.

Output

For each turkey, output the integer conversion. Note that even though the integer represents a price, you will not have to include the \$ sign.

Sample Input

3
III
LVIII
MCMXCIV

Sample Output

3
58
1994

12. Turkey Baking

Program name: turkeybaking.java

Input file: turkeybaking.in

Joe is in charge of baking turkeys for Thanksgiving dinner. However, Joe's super cool infinitely large oven takes some time to power up, so it can only bake 1 turkey a time the first round, then 2 turkeys at a time the second round, and n turkeys the n th round. Given the number of turkeys Joe needs to bake, write a program to calculate how many rounds of baking Joe will need to bake all of his turkeys.

Input

The one and only line of input contains an integer N ($1 \leq N \leq 500$), the number of turkeys that Joe needs to bake.

Output

Print a single integer containing the number of rounds of baking Joe will need to bake all his turkeys.

Sample Input

23

Sample Output

7

Explanation:

6 rounds of baking can bake $1+2+3+4+5+6 = 21$ turkeys, which does not fully cover the 23 turkeys Joe needs to bake, so Joe will need 7 rounds of baking, which can bake $1+2+3+4+5+6+7 = 28$ turkeys.

13. Turkey Race

Program name: turkeyrace.java

Input file: turkeyrace.in

Farmer John wants only the best of his Turkeys to 'help him out' with his Thanksgiving meal. He decides to hold a race to see which Turkey runs the fastest, with the reasoning that more athletic turkeys are generally tastier. Surprisingly, Farmer John's Turkeys have figured out his plan, and decide to run as slow as possible. Farmer's John numbers his turkeys 1, 2, 3, 4.... Each turkey has speed v and each second, each turkey i looks at all its adjacent turkey friends ($i - 1$ and $i + 1$), checking to see if either of them are running slower than itself. If so, then they decrease their speed by two. Otherwise, the turkey's speed will increase by one (why miss an opportunity to show off how fast you can run?). Given a race of n ($1 < n < 10^7$) turkeys lasting s seconds, please find the unlucky turkey that will be eaten this year. Note that turkeys will not run at a negative speed.

Input

The first line contains t , ($1 < t < 10^7$) the number of test cases

The first line in each test case contains n and s ($1 < s < 10^7$)

Each of the following n lines contains v for a single turkey. The first line after is turkey 1, the second is turkey 2, etc.

Output

Print the number of the turkey that has the fastest speed when time runs out. If there is a tie, print the index of all the turkeys that have the highest speed in increasing numerical order.

Sample Input

```
1
3 3
5
6
5
```

Sample Output

```
1 3
```

Explanation:

For the first turn, only turkey 2 will become slower, giving 6, 4, 6 as their speeds.

For the second, both the first and third will decrease their speeds, with 4, 5, 4 as their speeds.

Finally, the last turn will have the middle turkey decrease their speed by 2 and the turkeys in positions 1 and 3 increase their speed by one, making their speeds 5, 3, 5.

14. Turkey Beef

Program name: **turkeybeef.java**

Input file:
turkeybeef.in

In a battle between two turkey tribes, each have their own armies to send to battle. Within their armies, each turkey has a stat and battles against a corresponding enemy turkey they are assigned to. If a turkey is unassigned, they automatically win their matchup. A turkey wins their battle if their power level is strictly greater than another turkey's.

Justin the Turkey had beef against another tribe of turkeys and is planning to go to battle. Justin's army consists of **N** ally turkeys. However, one of his informants told him the stats of the opposition tribe's army that they are sending. They find that the opposition army has **M** enemy turkeys. He's paranoid and wonders how many battles he can win with his current army.

Input

The first line of the input contains **T**, the number of test cases. The number of test cases will not exceed 1000.

The first line of every test case contains 2 values, **N** and **M**, both of which are nonnegative integers less than or equal to 1000.

The next line contains **N** values (which represents the ally turkeys), with the **a_i**-th value being the power level of the **i**-th ally turkey. The power level of each turkey lies between 0 and 10^5 , inclusive. The next line contains **M** values (which represents the enemy turkeys), with the **a_i**-th value being the power level of the **i**-th enemy turkey. The power level of each turkey lies between 0 and 10^5 , inclusive.

Output

Output the maximum battles that Justin the Turkey can win if he assigns his turkeys to the enemies as strategically as possible.

Sample Input

```
1
5 6
1 2 3 4 5
5 3 6 4 9 10
```

Sample Output

```
2
```

Explanation:

Justin can only win two of his battles if he assigns his power 5 turkey to fight the enemy's power 4 turkey, and his power 4 turkey to fight the enemy's power 3 turkey..

15. Flying Turkeys

Program name: flyingturkeys.java

Input file: flyingturkeys.in

Terry the turkey has been testing how far he could fly so that he could be better prepared for escaping hunters in the Thanksgiving season. Terry wants to brag to all his other turkeys in his flock as to how long he can fly, so he wants to maximize the distance he can fly. Unfortunately for Terry, turkeys are not great at flying, so for every meter he flies horizontally, his vertical height also decreases by one.

Terry's ranch has a row of trees of various heights, each spaced precisely 1 meter apart. Assuming Terry begins his flight from the top of one of the trees and ends his flight on top of a tree as well, although not necessarily a distinct tree, find the maximum horizontal distance Terry can fly. (Assume that if Terry passes a tree that is exactly the same as his elevation, he is able to fly over it)

Input

The first line contains one integer **N** ($1 \leq N \leq 10,000$).

The next **N** lines contain h_1 to h_n , where h_i denotes the height of the i th tree

Output

Output one integer, the maximum horizontal distance Terry can fly if he starts on some tree and flies as far as possible—he must end either on or above a tree.

Sample Input

6
3
5
4
2
2
3

Sample Output

3

Explanation

In this case the best Terry can do is start on tree number 2 (so he starts with height 5) and can fly over the next 3 trees, with heights 4, 2, and 2 respectively. When he reaches tree 6, his height above the ground is 1, which is not high enough to fly over tree 6, which has height 3. This means the maximum distance Terry can fly between two trees is **3 meters**, from the 2nd tree with height 5 to the 5th tree with height 2.