

As and Bs
Input File: AsAndBsIn.txt

When Nadia was a child, she was frightened by the letters A and B. Finally, after years of therapy, she can tolerate an occasional A or B, but cannot deal with two consecutive occurrences of either A or B. For example “BOBBY” would make her say “AAK” except for the fact that “AAK” has a double occurrence of the letter A.

Since she receives a lot of email, she has asked you to write a program that will remove all double occurrences of the letter A and replace them with the letter B, and remove all double occurrences of the letter B and replace them with the letter A. Thus, BOBBY would become BOAY, and AAK would become BK. Sometime the replacement spawns another replacement. For example, TABBIES is initially changed to TAAIES, and then changed to TBIES. Your task is to read in a string of capital letters and output the string with no double occurrences of the letters A or B.

Inputs:

The first input will be the number of strings to process. This will be followed by one string per line.

Output:

There will one line of output per line of input, which will be the input strings with no double occurrences of A or B.

Sample Input

```
9
TBBBCABBA
AABCABBD
ABABABB
BABABAA
BBBCABBA
ABBAAB
AA
XBBAX
TBBBCABBAAABCABBDABABABBBABABAABBBBCABBAABBAABAAXBBAX
```

Sample output

```
TABCBA
ACBD
B
A
ABCBA
AB
B
XBX
TABCACBDABCXBX
```

Dice Game

Input File: DiceGameIn.txt

A group of n ladies are competing in a dice game in which each player rolls one standard die 10 times in each of r rounds. Player 1 completes her rounds first, then player 2, then player 3, etc., with player n rolling last. A player's score is the total of all of her die rolls. Once a player begins her turn, she continues to roll the die until she completes all r rounds *or* at the end of a round it is deemed she cannot win. Then the die is passed to the next sequential player.

Most games are completed in fewer than the maximum $r \times n$ rounds. For example, if two players were playing a game in which there were four rounds ($r == 4$) and the first player scored a perfect total of 240 points, then the game would be over as soon as the second player completed a round in which she scored less than 60 points. Your task is to determine the minimum number of rolls required to complete a game, the winning score, and winning player(s).

Inputs:

The first input will be the number of games to consider. This will be followed by two lines per game. The first of these lines will contain the number of players followed by the number of rounds for this game. The second line will contain a series of $n \times r$ integers that represent the total of the ten die rolls for each round of the game, had they not been interrupted. The first group of r numbers will be player one's rolls, the second group of r numbers will be player two's rolls, etc. All numbers on a line will be separated by a single space.

Output:

There will be two lines of output per game. The first line will contain the minimum number of rounds required to complete the game followed by the winning score. The second line will contain the player number(s) that won the game. All outputs on a line will be separated by a space.

Sample Input

```
4
2 4
60 60 60 40 60 40 50 34
2 4
60 60 60 40 60 41 60 60
4 4
60 45 60 35 60 19 60 30 50 60 20 30 50 60 20 50
2 5
10 10 10 10 10 60 10 20 30 40
```

Sample output

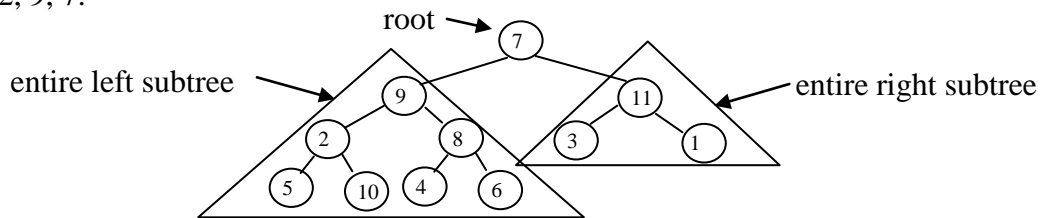
```
7 220
1
8 221
2
12 200
1 3
6 60
2
```

Eating Frenzy

Input File: EatingFrenzyIn.txt

Nora conducts an annual high school programming competition, which is followed by an eating frenzy and an awards ceremony. The competition results – a list of team numbers - are not divulged until the awards ceremony is completed, and so Nora has developed a scheme to encrypt the results. The *unencrypted* results list is simply a list of team numbers sorted in ascending order in the order in which the teams finished.

To generate the encrypted list, Nora “visualizes” the unencrypted results list as a left balanced binary tree (see below) and then outputs the tree’s entire right subtree, followed by its entire left subtree and finally the root of the tree, recursively. For example, if the team 7 won the competition, team 9 finished second, team 11 was third,... and team 6 came in last, then the *unencrypted* results list could be: **7, 9, 11, 2, 8, 3, 1, 5, 10, 4, 6**. In this case the encrypted list would be 1, 3, 11, 6, 4, 8, 10, 5, 2, 9, 7.



Your task is to automate the translation of the results list into the encrypted list, so that Nora has time to participate in the eating frenzy.

Inputs:

The first line of input contains the number of cases to consider. This will be followed by two lines of input per case. The first of these lines will contain the number of teams in the competition. The second line will contain a sequence of integers, each separated by a space, which represents the *unencrypted* competition results list.

Output:

There will be one line of output per case. Each line will contain a sequence of integers, each separated by a space, which represents the *encrypted* results list.

Sample Inputs

```

2
11
7 9 11 2 8 3 1 5 10 4 6
32
6 4 23 10 14 2 15 5 19 30 1 18 32 9 7 21 29 11 3 27 31 13 16 20 17 25 24 8 28 22 26 12
  
```

Sample Outputs

```

1 3 11 6 4 8 10 5 2 9 7
26 22 7 28 8 9 15 24 25 32 17 20 18 2 23 16 13 1 31 27 30 14 3 11 19 29 12 21 5 10 4 6
  
```

Fishing

Input File: FishingIn.txt

Skyler inherited a lucky fishing pole from an old man who was her neighbor. He stipulated in his will that half of the fish caught with the pole would be given to his wife. After a long fishing drought, Skyler decided to give the old man's fishing pole a try and filled the boat with fish. The strange thing was that each fish weighed the same.

Write a program to calculate the number of pounds of fish his wife will receive.

Inputs

There will be one line of input that contains two numbers. The first number will be the number of fish Skyler caught, and the second number will be the weight of each fish.

Outputs

There will be one line of output containing the two numbers: the total weight of the fish caught followed by the number of pounds of fish delivered to the wife. The output line should be annotated *exactly* as shown below. If a numeric output is an integer, you may output it with or without a trailing zero (e.g., 570 or 570.0).

Sample inputs

28 15.5

Sample output

Total weight of the catch = 434.0 pounds. Pounds of fish delivered = 217.0

Raise
Input File: RaiseIn.txt

Your Uncle Thaddeus, who happens to be your employer, likes to constantly test your intelligence. Each time you and your coworkers are due for a raise, he offers two choices: a raise of 5% or \$1.00 per hour. Write a program that will choose the option that results in the highest increase in pay.

Inputs:

There will be one line of input. The first number on the line will be the number of employees who are negotiating their raises. That will be followed by the hourly pay rates of the employees. All the inputs will be separated by a single space.

Output:

There will be one line of output per employee that will contain two outputs separated by a space. The first output will be the choice of the raise option (5% or \$1) and the second output will be the new hourly pay rate after the raise. The monetary outputs will be formatted as US currency. If the pay rate would be the same with either choice, choose the 5% option.

Sample Input

4 10.00 22.00 25.00 8.50

Sample output

\$1.00 \$11.00

5% \$23.10

5% \$26.25

\$1.00 \$9.50

Sample Input

3
100 8
10 5 0 1
7 5
2 2 1 1
7 5
2 2 1 2

Sample output

14
24
12

Vases
Input File: VasesIn.txt

Ryan's mom gave him a vase with a cherished inscription on it, which he often places on his window ledge. He is moving into a high rise apartment complex and wants to make sure that if the vase falls off the ledge, it won't break. So he plans to purchase several identical vases and drop them out of the window to determine the highest floor he can live on without the vase breaking should it fall off the window ledge. Time is short, so Ryan would like to minimize the number of drops he has to make.

Suppose that one vase was purchased and the building was 100 stories. If it was dropped out of the first floor window and broke, Ryan would live on the first floor. However if it did not break, it would have to be dropped from all subsequent floors until it broke. So the *worst* case minimum number of drops for *one* vase ($V=1$) and an N story building is always N drops. It turns out that if Ryan's drop scheme is used, the *worst* case minimum number of drops falls off rapidly as the number of vases increases. For example, only 14 drops are required (worst case) for a 100 story building if two ($V = 2$) vases are used, and the worst case number of drops decreases to 9 if three vases are used.

Your task is to discover Ryan's (the optimum) drop algorithm, and then code it into a program to determine the minimum number of drops (worst case) given the number of vases Ryan purchased, V , and the number of floors in the building, N . Assume that if a vase is dropped and it does not break, it can be reused.

Inputs:

The first line of input contains the number of cases to consider. This will be followed by one line of input per case. Each line will contain two integers: the number of vases Ryan purchased, V , followed by the number of stories in the building, N .

Output:

There will be one line of output per case, which will contain the minimum number of drops, worst case.

Sample Input

5
2 10
2 100
3 100
25 900
5 100

Sample Output

(1 for each input)
4
14
9
10
7