PLU February 2019 Programming Contest

Novice Division

I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.

2. Problems will have either no input or will read input from a specified file. All output should be to standard output (the monitor).

3. All input is as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros.

4. Your program should not print extraneous output. Follow the form exactly as given in the problem.

II. Names of Problems

Number	**Name**
Problem 1	Clothes
Problem 2	Friends
Problem 3	Periods
Problem 4	Population
Problem 5	Presidents
Problem 6	Star
Problem 7	Exam
Problem 8	Poker Hand
Problem 9	Time Limits
Problem 10	Pizza
Problem 11	Greedy Scheduler
Problem 12	PLU Logo

1. Clothes

Input File: clothes.dat

Billy's mom recently washed his clothes. When his mom finishes washing his clothes, she folds them and stacks them in one large pile. It is then Billy's responsibility to sort the clothes into three drawers for shirts, pants, and socks. When he places these items in the drawers he takes the top one off of the stack his mom gave to him and places it at the top of the stack in the respective drawer.

Each day Billy must wear a shirt, pants, and socks. He simply picks the top one off each stack in each drawer to wear for that day. Given the original stack of washed clothes, determine what Billy will wear up until he needs more clothes washed. Billy needs more clothes when he runs out of any necessary article of clothing for the day. Billy will only wear a piece of clothing once before it is declared dirty and needs washing. He does not ever wear dirty clothes.

Input

The first line of input will contain a single integer `n` that indicates the number datasets to follow. Each dataset begins with an integer `c` which is between 1 and 50. The next `c` lines are the stack of clean clothing in the format "*`article (type)`*" where *`article`* is the name of the article of clothing and *`type`* is the type of clothing. The value of *`type`* will be `shirt`, `pants`, or `socks`.

Output

The output will be a series of lines. Each line will contain the three articles of clothing for the specific day separated by a comma and one space. The shirt will be printed first followed by the pants and socks. The type should not be printed. There should be one blank line after each dataset.

Example Input File

```
2
6
mario logo (shirt)
red (pants)
hawaiian (shirt)
baggy (pants)
white (socks)
fancy (socks)
10
khaki (pants)
blueish green (shirt)
apple (shirt)
bright (socks)
blue (pants)
red (pants)
purple (socks)
ugly (shirt)
stupid (socks)
big red (pants)
```

Example Output to Screen

```
hawaiian, baggy, fancy
mario logo, red, white

ugly, big red, stupid
apple, red, purple
blueish green, blue, bright
```

2. Friends

Input File: friends.dat

Jane has many friends, but she likes some less than others. She has a number associated with each friend which represents how much she likes each friend. The higher the number, the more she likes that friend. Given her list of friends, sort them so that the best friends are at the top and the least best friends are at the bottom.

Input

The first line of input will contain a single integer `n` that indicates the number datasets to follow. Each dataset will consist an integer `m`, (0 < `m` < 500), which is the number of friends that will be in the dataset. The next `m` lines will have the format "*`name num`*", where *`name`* is the name of the friend and *`num`* (0 < *`num`* < 500) is the number that represents how much she likes her friend. The name will only be one word. No two friends will not have the same *`num`* value.

Output

Output a list of names, separated by a comma and one space, of the friends in order of how much Jane likes them.

Example Input File

```
2
2
bill 2
greg 5
4
jim 5
phil 8
paul 7
rob 2
```

Example Output to Screen

```
greg, bill
phil, paul, jim, rob
```

3. Periods

Input File: periods.dat

Eric gets distracted so sometimes he forgets to put periods at the end of his sentences. To help him out, you are to put a period at the end of his sentences if the period is not already present.

Input

The first line of input will contain a single integer `n` that indicates the number of lines to follow. Each line will consist of a sentence which may or may not have a period at the end.

Output

Output the sentence, making sure there is one period at the end.

Example Input File

```
3
You kicked my dog
No I did not.
It was the kid that did
```

Example Output to Screen

```
You kicked my dog.
No I did not.
It was the kid that did.
```

4. Population

Input File: population.dat

In Emily's country it is estimated that a person dies every 7 seconds, and a person is born every 4 seconds. Given a beginning population, estimate the population after a certain period of time.

Input

The first line of input will contain a single integer `n` that indicates the number of lines to follow. Each line will consist of two integers, `p` and `t`, where `p` is the beginning population, and `t` is the amount of time that will pass. Both `p` and `t` will be between the number 1 and 2,000,000,000.

Output

Output the estimated population after the period of time based on the above statistics.

Example Input File

```
3
12 14
530 200
4786 3543
```

Example Output to Screen

```
13
552
5165
```

5 Presidents

Input File: presidents.dat

Mark keeps a log of all the bills in his wallet, but the log does not indicate the actual value of the bills - it simply states the last name of the president on the front of each bill. You are to determine the actual value of the bills in Mark's wallet.

Input

The first line of input will contain a single integer `n` that indicates the number of lines to follow. Each line will consist of a series of last names of presidents. The presidents that could appear are: `Franklin`, `Grant`, `Jackson`, `Hamilton`, `Lincoln`, and `Washington`. The values of these presidents are \$100, \$50, \$20, \$10, \$5, and \$1, respectively.

Output

Output the value of the money in the wallet with a dollar sign in front.

Example Input File

```
3
Franklin Grant Jackson
Hamilton Lincoln Washington
Washington Washington Washington Franklin Jackson
```

Example Output to Screen

```
$170
$16
$123
```

6. Star

Input File: none

Print the star.

Input
None

Output
Print the star-like shape exactly as shown below. Print the . (period) and * (star) symbols.

Example Output to Screen
```
.........*.........
.........**........
.........*.*.......
...******...*......
....*........******
.....*...........*.
.....*..........*..
....*..........*...
...*....*.....*....
..*...*..*...*.....
.*..*......*..*....
.**..........*.*...
...............**..
```

7. Exam

Input File: exam.dat

Your friend and you took a true/false exam of n questions. You know your answers, your friend's answers, and that your friend got k questions correct.

Compute the maximum number of questions you could have gotten correct.

Input

The first line of input will contain a single integer *m* that indicates the number datasets to follow. Each dataset begins with a single integer k. The second line in the dataset contains a string of n (1 ≤ n ≤ 1000) characters, the answers you wrote down. Each letter is either a 'T' or an 'F'. The third line in the dataset contains a string of n characters, the answers your friend wrote down. Each letter is either a 'T' or an 'F'. The input will satisfy 0 ≤ k ≤ n.

Output

For each dataset print, on one line, the maximum number of questions you could have gotten correct.

Example Input File

```
2
3
FTFFF
TFTTT
6
TTFTFFTFTF
TTTTFFTTTT
```

Example Output to Screen

```
2
9
```

8. Poker Hand

Input File: poker.dat

You're given a five-card hand drawn from a standard 52-card deck. Each card has a rank (one of A, 2, 3, . . . , 9, T, J, Q, K), and a suit (one of C, D, H, S).The strength of your hand is defined as the maximum value k such that there are k cards in your hand that have the same rank.

Find the strength of your hand.

Input

The first line of input will contain a single integer `n` that indicates the number datasets to follow. Each dataset will consist a single line, with five two-character strings separated by spaces. The first character in each string will be the rank of the card, and will be one of A23456789TJQK. The second character in the string will be the suit of the card, and will be one of CDHS. It is guaranteed that all five strings are distinct.

Output

For each dataset output, on a single line, the strength of your hand.

Example Input File

```
2
AC AD AH AS KD
2C 4D 4H 2D 2H
```

Example Output to Screen

```
4
3
```

9. Time Limits

Input File: time.dat

A contest setter wants to determine the time limits for a given problem. There are n model solutions, and solution k takes t_k milliseconds to run on the test data. The contest setter wants the time limit to be an integer number of seconds, and wants the time limit to be at least s times larger than the slowest model solution. Compute the minimum time limit the contest setter can set.

Input

The first line of input will contain a single integer `m` that indicates the number datasets to follow. Each dataset is comprised of two lines. The first line of input contains two space-separated integers n and s ($1 \leq n \leq 100$ and $1 \leq s \leq 20$). The second line of input contains n space-separated integers $t_1, \ldots, t_n$ ($1 \leq t_k \leq 2000$ for all $k = 1, \ldots, n$).

Output

For each dataset print, on one line, the minimum time limit (in seconds) as a single integer.

Example Input File

```
2
2 5
200 250
3 4
47 1032 1107
```

Example Output to Screen

```
2
5
```

10. Pizza

Input File: pizza.dat

There’s a pizza store which serves pizza in two sizes: either a pizza slice, with area A1 and price P1, or a circular pizza, with radius R1 and price P2. You want to maximize the amount of pizza you get per dollar. Should you pick the pizza slice or the whole pizza?

Input

The first line of input will contain a single integer `n` that indicates the number datasets to follow. Each dataset is comprised of two lines. The first line contains two space-separated integers A1 and P1. Similarly, the second line contains two space-separated integers R1 and P2.
It is guaranteed that all values are positive integers at most 10^3. We furthermore guarantee that the two will not be worth the same amount of pizza per dollar.

Output

For each data set, print on a single line, either “Whole pizza” or “Slice of pizza” depending on which is the better deal.

Example Input File

```
3
8 4
7 9
9 2
4 7
841 108
8 606
```

Example Output to Screen

```
Whole pizza
Whole pizza
Slice of pizza
```

11. Greedy Scheduler

Input File: greedy.dat

A store has n cashiers numbered sequentially from 1 to n, with c customers in a queue. A customer at the front of the queue is constantly assigned to the first unoccupied cashier, (i.e., if more than one cashier is unoccupied select the cashier with the smallest number). The i^{th} customer's shopping cart takes t_i seconds to process.

Find which cashier will process each customer's shopping cart.

Input

The first line of input will contain a single integer `m` that indicates the number datasets to follow. Each dataset is comprised of two lines. The first line of input contains two space-separated integers n and c ($1 \leq n \leq c \leq 1000$). The second line of input contains c space-separated integers $t_1, \ldots, t_c$, representing the length of time required to handle that customer.

Output

For each dataset print, on a single line c integers separated by one space, where each integer is the cashier number that handles that customer.

Example Input File

```
1
3 10
406 424 87 888 871 915 516 81 275 578
```

Example Output to Screen

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

12. PLU Logo

Input File: none

Print the PLU logo as shown below.

Input
None

Output
Print the PLU logo exactly as shown below.

Example Output to Screen

```
PPPPPP  L       U    U
P    P  L       U    U
PPPPPP  L       U    U
P       L       U    U
P       L       U    U
P       LLLLLL  UUUUUU
++++++++++++++++++++++
///////////\\\\\\\\\\\
```