Computer Science Competition
Hands-On Programming Set

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. All problems have a value of 60 points.

3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.

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

5. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.

II. Names of Problems

Number	Name
Problem 1	Claim the Hand
Problem 2	Math
Problem 3	Chess
Problem 4	Sort
Problem 5	War
Problem 6	Yes
Problem 7	Powers
Problem 8	Rook
Problem 9	N4096
Problem 10	Vault
Problem 11	Rat
Problem 12	Number

NOVICE ONLY PROBLEMS	
Problem 13	Cost Cutting
Problem 14	Magic Formula
Problem 15	Pushups
Problem 16	Skip

1. Claim the Hand

Program Name: hand.java **Input File: hand.dat**

A jousting tournament just started, to see which knight is worth enough for m'lady's hand. Knights from all over the world come together to fight for m'lady's hand.

The rules are as follow. First, we will use knockout rules, meaning if a knight loses he will be kicked from the tournament. Second, knights can only fight each other if the difference in victories between them is less than one. Third and lastly, all knights are to have fun.

Input

An integer N will denominate the number of test cases. Each test case represents a unique tournament containing a unique number of knights. The following N lines will contain a number M denominating the number of knights competing in that tournament. There will be at least 2 knights contesting m'lady's hand. Tournament configurations are not required to ALWAYS be valid. They must have at least 1 possible way of being valid.

Output

You are to output the maximum number of battles the winner of the tournament can win.

Example Input File

```
3
2
4
13
```

Example Output to Screen

```
1
2
5
```

2. Math

Program Name: math.java **Input File: math.dat**

You have recently been hired by a new start-up company, Calc. They intend to make a program that solves mathematical expressions. Because you are their only programmer it is your job to write said program. The expressions will only contain the following mathematical operators +,-,*, and / which represent add, subtract, multiply, and divide respectively. Take in mind that the expressions may contain multiple parenthesis. Follow the standard order of operations to solve each expression.

Input

An integer N will denominate the number of test cases. The following N line will contain an expression. The expression will only contain integers and the operators specified in the description. Negatives will be denoted as −#.

Output

You are to output the result of each expression on its own separate line. The answer will be rounded to the hundreds place.

Example Input File

```
3
2+2
2*(2+2)
1/2
```

Example Output to Screen

```
4.00
8.00
0.50
```

3. Chess

Program Name: chess.java **Input File: chess.dat**

Chess is an amazing game. A true game of wit and deception. Print out a chess board identical to that in the output. The uppercase letters represent the black pieces and the lowercase letters represent the white ones. The letters R,H,B,Q,K,P represent Rook, Horse, Bishop, Queen, King, and Pawn respectively.

Input
None.

Output
Print out a chess board identical to the one in the output.

Example Input File
```
None.
```

Example Output to Screen
```
RHBQKBHR
PPPPPPPP
........
........
........
........
pppppppp
rhbkqbhr
```

4. Imperial Sort

Program Name: sort.java **Input File: sort.dat**

The Caesar is angry at you for sleeping on your job. As punishment, he orders you to sort his chess pieces. Because the Caesar is so rich and powerful he has many chess pieces. He instructs you to sort them by importance. If a piece is white it will be represented by an uppercase letter if its black it will be represented by a lowercase letter. The letters R,H,B,Q,K,P represent Rook, Horse, Bishop, Queen, King, and Pawn respectively. The most important piece is the king followed by the queen, rook, horse, bishop, and finally the pawn. Take note that the Caesar prefers the white pieces over the black ones so even a white pawn is considered more important than a black king.

Input

An integer N will denominate the number of test cases. The following N lines will contain a series of space separated letter which represent chess pieces.

Output

Output the chess pieces in ascending order of importance. The output ends immediately after the last chess piece is printed (i.e. there should NOT be a space after the character representing the last chess piece).

Example Input File

```
3
P K Q R H
K k K R q q
P k q q r r k P
```

Example Output to Screen

```
P H R Q K
q q k R K K
r r q q k k P P
```

5. Honorable Wars

Program Name: war.java **Input File: war.dat**

The Caesar will only fight a war if said war is an honorable war. A war is considered honorable if the number of friendly soldiers equals the number of enemy soldiers. However, because the Caesar's regiments have a high level of comradery they refuse to split up into smaller sub units. It's is your job to find out if a war will be honorable. If a combination of regiments equals the enemy army then the war is honorable else it will be dishonorable.

Input

An integer N will denominate the number of test cases. The following N cases will contain an integer M, representing the number of soldiers in the enemy army. On the next line a series of space separated numbers will represent the number of soldiers in each regiment.

Output

If it's possible to attain a combination of regiments that equal to the enemy army then print out "HONOR" else print "DISHONOR".

Example Input File

```
2
3
2 4 4 6
30
16 6 23 84 276 1
```

Example Output to Screen

```
DISHONOR
HONOR
```

6. Yes or No

Program Name: yes.java **Input File: yes.dat**

The empire is ruled like a democracy and a dictatorship at the same time. For instance, if the Caesar is present then whatever he votes for will be the decision. If he's not present then the decision will fall to a democratic vote. Meaning whatever most of the people want will be followed. A person can change their vote by voting again, but only the last vote counts.

Input

An integer N will denominate the number of test cases. The following N cases will contain an integer M. This integer, M, denominates the number of people participating in the vote. The following M lines will contain a name following by the letter Y or N meaning yes or no, respectively. All the names will be in lowercase for simplicity's sake.

Output

If the Caesar is not present print out what most of the people choose, YES or No. Else if the Caesar is present print out what the Caesar choose. In the event of a tie print out Tie.

Example Input File

```
3
2
rem Y
ram Y
3
rem Y
rem N
ram N
4
emilia Y
rem Y
ram Y
caesar N
```

Example Output to Screen

```
YES
No
No
```

7. Powers of 7 and 8

Program Name: powers.java

There are two 4-digit numbers that if you raise each of their digits to the 7th power and add them together you get a number that in the octal number representation is equal to the original number.

Let's try this with the number 2123. The sum of all four digits of 2123 each raised to the 7^{th} power is

$$2^7 + 1^7 + 2^7 + 3^7 = 2444$$

Treating 2444 as an octal number we see that its decimal value is 1316. However, since 1316 is not equal to the original number 2123, this is NOT one of the two numbers we are looking for.

Input

None.

Output

Print each of the numbers on a separate line

8. Rook

Program Name: rook.java **Input File: rook.dat**

In chess, the rook or tower piece can move either vertically or horizontally as many steps as the player pleases. Your job is given a chess board you are to find out how many places are safe to put your piece such that it can't be taken by a rook on the next turn. There can be more than 2 rooks on the board at a time. You may assume that there will only be rooks on the board. The standard chess board is 8x8. Use the following board to figure out the location of the rook.

57	58	59	60	61	62	63	64
49	50	51	52	53	54	55	56
41	42	43	44	45	46	47	48
33	34	35	36	37	38	39	40
25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24
9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8

Input

An integer N will denominate the number of test cases. The following N cases will contain a number M which will denominate the number of rooks that will be present on the board. The following M lines will contain a number between 1 and 64 inclusive of 1 and 64 which will represent the location of the rook.

Output

Print out the number of places on the board in which it is safe to put your piece.

Example Input File

```
2
2
34
13
1
1
```

Example Output to Screen

```
36
49
```

9. 4096

Program Name: N4096.java **Input File: N4096.dat**

Your friend just came up with a great idea for a game. He explains the rules to you.

- You can only swipe left, right, up, and down
- The game is set on a 4x4 tile board
- Each tile will have the one of the following values: *, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096
- When you swipe in a direction all the tiles will shift in that direction if the tile in front of it is of the same value or it is a space denoted by an *.
- If when you swipe the tiles are of the same value they will combine into one tile of the value of one of the tiles times 2.
- Each row and column is independent of each other
- A tile can only be combined once in a turn. Ie. 2 2 2 2 yields * * 4 4 instead of * * * 8

You are to make a program that prints out a board after a board and a direction to shift is given to you.

Input

The first line will contain a single integer `n` that indicates the number of data sets that follow. Each data set will contain a 4x4 4096 board followed by the direction you are to shift the board. Each tile of the board is separated by a space.

Output

Output the board after it has been shifted in the requested direction.

Example Input File

```
2
* * * *
* * * *
* * * *
2 2 * *
left
2 2 2 *
4 8 4 *
16 32 8 4
* * * *
right
```

Example Output to Screen

```
* * * *
* * * *
* * * *
4 * * *
* * 2 4
* 4 8 4
16 32 8 4
* * * *
```

10. Caesar's Vault

Program Name: vault.java **Input File: vault.dat**

You are a robber and you seek to still as many riches from the Caesar's vault as you can. However, you can only carry so much weight. Find out what's the maximum value of riches you can steal from the vault.

Input

The first line will contain a single integer N that indicates the number of data sets that follow. Each data set will contain two integers, M and X. Integer M will designate the amount of riches and integer X will denote the maximum weight of riches that you can carry. The following line will contain the value of the riches and the next line will contain their respective weights.

Output

Print out an integer that will denote the maximum value of riches that you can steal form the vault.

Example Input File

```
2
4 10
3 7 27 4
3 5 26 4
1 10
20
11
```

Example Output to Screen

```
11
0
```

11. Vault Rat

Program Name: rat.java **Input File: rat.dat**

The guards found you stealing the riches now it's time to find a way out. You must find the fastest way out the vault. You can only move horizontally and vertically. Your only obstacle to escape this vault are the mountains of gold, denoted by the 'G' character, you cannot traverse this character. Each tile you traverse takes you exactly 1 second. You are to print out the minimum number of seconds that it will take u to escape the maze. The letter 'S' is you starting location and 'E' will be your exit.

Input

The first line will contain a single integer `N` that indicates the number of data sets that follow. Each data set will start with an integer R and C which will denote the dimensions of the maze followed by the maze.

Output

Output the minimum number of seconds that takes to escape the vault in the following format "X seconds" x being the minimum number of seconds. If you are not able to escape the vault print out "Oh Rem please help me".

Example Input File

```
2
2 2
S.
.E
3 3
S..
.GG
.GE
```

Example Output to Screen

```
2 seconds
Oh Rem please help me
```

12. Number

Program Name: number.java **Input File: number.dat**

You have discovered a new piece of technology. However, the numbers on it do not please the Caesar. Your job it to translate those number into the roman numerals. The roman numeral follows the following rules.

- We read them from left to right.
- If the number to the left is smaller than the one to the right, we subtract the number on the right from the number on the left. Example: IX = 9
- We add all the values after the subtractions if any. Example: XXX = 30, XIX = 19
- Numbers can only be repeated 3 times.
- Only I, X, C, M can be repeated.
- V, L, and D can never be on the left of a larger number. Ie. LC is invalid

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

Input

The first line will contain a single integer `N` that indicates the number of data sets that follow. The following lines will contain a integer M which will be the number that you will be turning into a roman numeral.

Output

Print out the roman numeral equivalent of the number that you were to translate.

Example Input File

```
3
9
19
532
```

Example Output to Screen

```
IX
XIX
DXXXII
```

PROBLEMS 13-16 are for NOVICE division ONLY

13. NOVICE ONLY: Cost Cutting

Program Name: cost.java **Input File: cost.dat**

Company XYZ has to cut some costs. They have many departments with only three (3) employees and they decide they are are going to lay-off two (2) of them. After a series of meetings, they have decided to lay off the person who gets the most salary and the one who gets the least. You will be given the salaries of each of the 3 employees in a department and have to determine the salary of the person who survives.

Input

The first line of input is an integer T (T < 20) that indicates the number of test cases. Each case consists of a line with 3 distinct positive integers. These 3 integers represent the salaries of the three employees. All these integers will be in the range [1000, 10000].

Output

For each case, output the case number followed by the salary of the person who survives.

Example Input File

```
3
1000 2000 3000
3000 2500 1500
1500 1200 1800
```

Example Output to Screen

```
Case 1: 2000
Case 2: 2500
Case 3: 1500
```

Credit: UVa Online Judge Problem #11727

PROBLEMS 13-16 are for NOVICE division ONLY

14. NOVICE ONLY: Magic Formula

Program Name: magic.java **Input File: magic.dat**

You are given a quadratic function, f(x) = ax2 + bx + c

You are also given a divisor d and a limit L.

How many of the function values f(0), f(1), . . . , f(L) are evenly divisible by d?

Input
Input consists of a number of test cases. Each test case consists of a single line containing the numbers a b c d L (−1000 ≤ a, b, c ≤ 1000, 1 < d < 1000000, 0 ≤ L < 1000). Input is terminated by a line containing '0 0 0 0 0' which should not be processed.

Output
Print the answer for each test case (the number of function values f(0), f(1), . . . , f(L) divisible by d) on a separate line.

Example Input File
0 0 10 5 100
0 0 10 6 100
1 2 3 4 5
1 2 3 3 5
0 0 0 0 0

Example Output to Screen
101
0
0
4

Credit: UVa Online Judge Problem #11934

PROBLEMS 13-16 are for NOVICE division ONLY

15. NOVICE ONLY: Pushups

Program Name: pushups.java

You are trying to get to the point where you can do 40 pushups.

Your personal trainer devises a training program for you that has you doing a specific number of pushups every day.

She has you doing **N** pushups every day for **N** days and periodically adding 1 pushup according to these rules.

- if it is an odd numbered day, again do **N** pushups every day for **N** days.
- if it is an even numbered day, change to doing **N+1** pushups every day for **N+1** day.

She starts you out at 1 push per day on the first day.

Input

```
None.
```

Output

Print out the days when the number of pushups increases in the format shown below. The last line you should print is when you start doing 40 pushups a day. NOTE: There is no "s" at the end of "pushup" for the first day. Make sure your capitalization and punctuation matches the below.

Example Output to Screen

```
On day 1 do 1 pushup every day for 1 day.
On day 2 do 2 pushups every day for 2 days.
On day 4 do 3 pushups every day for 3 days.
On day 10 do 4 pushups every day for 4 days.
On day 14 do 5 pushups every day for 5 days.
On day 24 do 6 pushups every day for 6 days.
On day 30 do 7 pushups every day for 7 days.
  .
  .
  .
```

PROBLEMS 13-16 are for NOVICE division ONLY

16. NOVICE ONLY: Skip

Program Name: skip.java

Starting at 1 print 10 numbers in a row where each number is 1 more than the previous number.
Starting at 1 print 10 numbers in a row where each number is 2 more than the previous number.
Starting at 1 print 10 numbers in a row where each number is 3 more than the previous number.

Do this for 40 rows.

Input
None.

Output
Print each set of numbers on a separate line. There is a single space after each number, including the last on the line.

Example Output to Screen
```
1 2 3 4 5 6 7 8 9 10
1 3 5 7 9 11 13 15 17 19
1 4 7 10 13 16 19 22 25 28
1 5 9 13 17 . . .
```



```
1 41 81 121 161 201 241 281 321 361
```