Problem A
Count Me In
Time Limit: 2 seconds

Given a sentence in English, output the counts of consonants and vowels.

Vowels are letters in ['A','E','I','O','U','a','e','i','o','u'].

Input

The test file starts with an integer $S(1 \leq S \leq 100)$, the number of sentences.

Then follow S lines, each containing a sentence - words of length 1 to 20 separated by spaces. Every sentence will contain at least one word and be comprised only of characters [a-z][A-Z] and spaces. No sentence will be longer than 1000 characters.

Output

For each sentence, output the number of consonants and vowels on a line, separated by space.

Sample Input	Sample Output
3 You can win this thing May be too optimistic Just try to have fun	12 6 10 8 11 5

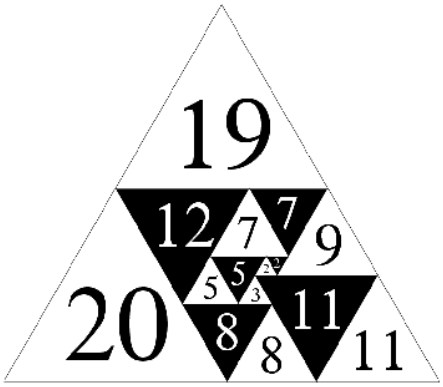
This page is intentionally left blank.

Problem B

Triangulate This!

Time Limit: 2 seconds

Given two positive integers A and B, where $B \leq A$ and B divides A, how many equilateral triangles of size B you need to completely cover the triangle of size A.



Input

The test file starts with an integer $T (T \leq 100)$, the number of test cases.

Each test case consists of two integers A and B on a line. $(1 \leq B \leq A \leq 1,000, B|A)$

Output

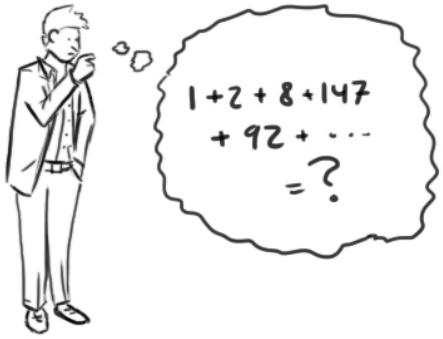
For each test case, output the minimum number of equilateral triangles with side B that can completely cover the equilateral triangle with side A.

Sample Input	Sample Output
2 2 1 3 3	4 1

This page is intentionally left blank.

Problem C
Addition Affliction
Time Limit: 3 seconds

Albert is a Computer Science student that loves math! Unfortunately, despite his love for math, he does not have a very good handle on the basics! He especially has difficulties with addition, an affliction that has plagued him since his youngest days.



Unfortunately for Albert, addition is a common part of life. He would like to finally master the mysterious magic of addition. Albert has noticed that numbers which add to a multiple of 10 are the easiest.

Albert would like you to help him achieve his addition aspirations; he doesn't want you to do all of his work for him (that wouldn't help!), but if you could rearrange some of his practice expressions into something a little easier, he would be very grateful!

Input

Each line of input will be a summation of nonnegative integers in the form

$$x_1 + x_2 + \cdots + x_N$$

for some $N (2 \leq N < 1000)$, where $0 \leq x_i \leq 100000$ for each x_i.

Lines will not contain any whitespace, only digits and the plus symbols.

The input will be terminated by a line containing only 0.

Output

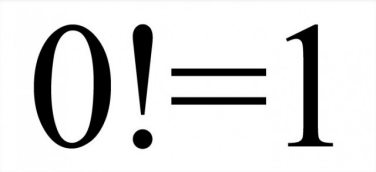
For each line, print the same expression but with the numbers in an easier order: put pairs of numbers that add to a multiple of 10 at the front. That is, if there is a number x_i in the sequence for which $x_i + x_j$ is a multiple of 10 for some other number x_j in the sequence, and x_j is not already paired with another number, then x_i, x_j should appear adjacent to each other before any unpaired numbers.

Note that there may be several correct answers for a given line, e.g. "1+2+4+6" could be rearranged as "4+6+1+2" or "6+4+2+1".

Sample Input	Sample Output
1+2+3+4+9 153+214+64+7+26 1+2+3+4+5 1951+1569+481+4823+142+4677 10+9+8+7+2+1 1+3+5+10+15+20+30 0	1+9+2+3+4 153+7+64+26+214 1+2+3+4+5 1951+1569+4823+4677+481+142 9+1+8+2+7+10 10+20+5+15+1+3+30

Problem D
Numbers are Easy
Time Limit: 2 seconds

Given an integer N, what is the smallest positive integer X, whose representation in base 10 consists only of digits '0' and '1', such that X is divisible by N?



Input

The first line of the input file starts with the integer T, the number of test cases $(1 \leq T \leq 100)$. Each test case consists of a number $N(1 \leq N \leq 300)$ on a line.

Output

For each test case, output the smallest positive number X such that X is divisible by N and it contains only digits '0' and '1'. Given the constraints, it is guaranteed that there always is a solution (for any positive N) and, in this problem, it will always fit into a 64-bit signed integer.

Sample Input	Sample Output
3 1 2 20	1 10 100

This page is intentionally left blank.

Problem E
Best Buddies
Time Limit: 6 seconds

You have signed up for the annual Calgary Collegiate Programming Contest. Unlike previous years, participants are not choosing their own team members, but are assigned to teams as follows. Consider we list all participants in alphabetic order. Then Team 1 consists of the first three participants in this order. Team 2 consists of the next three participants in this order. Assume that the number N of participants is divisible by 3 so that every team has exactly 3 team members.

Assume that every participant has exactly one first name and exactly one last name, the name is spelled over the alphabet $\{a, \ldots, z\}$, and that no two participants carry the exact same name. The alphabetic order we choose, is to first sort in lexicographically ascending order with respect to last names, and then, when necessary, with respect to first names. So john smith is alphabetized after carla smith, but before bob taylor.

Given the name of a participant, output the names of the participant's two team mates.

Input

The first line of the input contains an integer, N, the number of participants ($3 \leq N \leq 99,999$). Then follows N lines, each containing the first name and last name of the participant. Both first and last name consist of at least 2 and at most 16 lower case characters of English alphabet. The next line contains an integer, Q, the number of queries ($1 \leq Q \leq N$). Then follows Q lines, each containing the first name and last name of the participant. The queried names are guaranteed to exist in the original list.

Output

For each of the Q test cases, output the names of the two team mates in the described order.

Sample Input	Sample Output
12 john smith carla smith bob taylor alice watson sarah taylor ryan singh jessica li kevin wright charles lee megan davies ryan davies julia leung 3 ryan davies sarah taylor kevin wright	megan davies charles lee alice watson kevin wright sarah taylor alice watson

Problem F
Solitaire
Time Limit: 2 seconds

You are playing a game of solitaire on an array of $N+1$ cells, where each I^{th} cell contains value I. You start with the sum $S = 0$ at the cell 0 and keep throwing a regular 6-sided die and making the moves to the "right" (towards the N^{th} cell), adding the value of the cell to S. The game is over when you land on the last (N^{th}) cell. If you are to land past the last cell, you have to disregard that die throw. For example, if you are on second-to-the-last cell, you have to wait until the die shows 1 to make your last move.

Instead of throwing the die, we provide you with our Almost Random Die Throw Generator ™, which is a permutation of the sequence 1,2,3,4,5,6. Concatenate this sequence as many times as you need in order to find out what the next throw is. For example, if the Almost Randomly Die Throw Generated Sequence ™ is 2,4,6,1,3,5, after the 6^{th} throw yields 5, the 7^{th} will yield 2 again, then 4, then 6 and so on ...

Your score for the game is the sum at the end of the game. Given N and the Almost Randomly Die Throw Generated Sequence ™, what is your score for this game?

Input

The first line of the input file starts with the integer T, the number of games to process ($1 \leq T \leq 100$).

Each case consists of two lines. First one contains the integer $N (1 \leq N \leq 1000)$, as described in the statement. The second line contains the Almost Randomly Die Throw Generated Sequence ™

Output

For each game, output its final score on a line. Again, you must end the game at the last cell.

Sample Input	Sample Output
2 10 1 2 3 4 5 6 100 2 4 6 1 3 5	20 1505

This page is intentionally left blank.

Problem G
Serves Me Right
Time Limit: 2 seconds

You are in charge of maintaing a web server. There are up to N users that can connect to it and your company is paying for N concurrent licenses (meaning, all N of users can be logged in at the same time).

The economy is taking a hit and your company is looking to cut costs in all areas. Your supervisor would like to know if paying for N licenses is really necessary.

Server logs events in the following way:

- log time when a user connects to the server (user session starts)
- log time when a user disconnects from the server (user session is removed)
- log time when the server is restarted

Note that not all users necessarily log out (they may just close their browser instead) - that is why there is a session timeout value in minutes, at the end of which a user is disconnected regardless of his/her activity at the time. This automatic disconnect is not logged.

When web server is restarted, all users are disconnected and their sessions are removed from the server.

Given a web server log for a day and a user session timeout value, what is the maximum number of users logged in at any given point in time?

Input

The first line of the input file starts with the integer T, the number of web logs to process $(1 \leq T \leq 100)$.

Each case starts with two integers on the first line - E and TO $(1 \leq E, TO \leq 1440)$ Then follow E log entries on each line. They can be in either of these two formats:
TIME SERVER RESTART
or
TIME USER ⟨USER_NAME⟩ ⟨USER_ACTION⟩

TIME is in the format HH:MM (24-hour clock)
USER_NAME is a non-empty string of up to 16 lower case english characters. There are no duplicate user names (i.e. if you see the same user name, it refers to the same user)
USER_ACTION is one of LOG_IN or LOG_OUT

The entries are ordered by time, all times are distinct. There will be no incosistencies in the server log. Not all N users have to be present in the log.

Output

For each test case, output the number of users that are using the web server and the maximum number of users logged in at the same time over the period covered by the log. Output both numbers on the single line, separated by a single space character.

Sample Input	Sample Output
2 4 30 08:30 USER alice LOG_IN 09:00 USER bob LOG_IN 09:30 USER alice LOG_IN 10:00 USER bob LOG_IN 4 45 08:30 USER alice LOG_IN 09:00 USER bob LOG_IN 09:30 USER alice LOG_IN 10:00 USER bob LOG_IN	2 1 2 2