A - Blocks for kids

Time Limit: 5 seconds

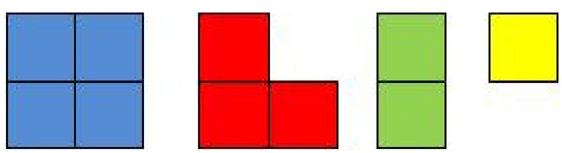
Wango is a brilliant maths teacher. He has two sons Kango and Dango. They are born two years apart on the same day! Kango is 9 and Dango is 7. Their birthdays are approaching again. Wango has to buy them a gift each. After long thought, this time Wango wanted to give each of his sons a piece of the Pango board and thus introduce them to higher mathematics.

A size n (n >= 0) Pango board is a 2 by n rectangle of unit squares. A pango board has to be tiled with Pango pieces. Any tiling with the Pango pieces is acceptable. A size 0 Pango board exists and is unique and serves as an example of the empty set.

Four types of Pango pieces are available.

```
1    2    3    4
==   ==   ==   ==
XX   X    X    X
XX   XX   X
```

Picture of the four kinds of pieces:



When Wango presents a board to Kango or Dango, he has to tile the board completely with these pieces (unlimited number of pieces of each type are available) and then give them out. Note that pieces cannot be rotated for tiling. To cut costs (recession mind you), Wango decides to buy a single board of size N, then choose a k (0 <= k <= N), and give a size k board to Kango and a size (N-k) board to Dango, (tiled of course). Help him find the number of ways he can give the presents. Two ways are distinct if and only if either Dango gets a different board or Kango gets a different board. Two Pango boards are considered the same if and only if they have the same tiling (same set of tiles at the same places) from left to right (rotation of board is not allowed in comparing).

Input

The input consists of a sequence of cases, one per line.
Each case consists of one integer N (0 <= N <= 1000,000,000) representing the size of the board which Wango is going to buy.
The input will end with a line containing -1. This case should not be processed.
There will be a maximum of 300 test cases.

Output

One line per case, outputting the number of ways Wango can distribute the presents to his sons modulo 10,007.

Sample Input

```
0
1
2
-1
```

Sample Output

```
1
4
16
```

Explanation

Number of different 0-sized Pango boards = 1
Number of different 1-sized Pango boards = 2
Number of different 2-sized Pango boards = 6
For N = 0, he has to give 0-sized boards to both his sons. He can do this in 1*1 = 1 way
For N = 1, he has to give 0-sized board to one of his sons, and 1-sized board to the other, for a total of 2*1 + 1*2 = 4 ways
For N = 2, he can give the presents in 6*1 + 2*2 + 1*6 = 16 ways

B - Paper Presentation

Time Limit: 4 seconds

2M scientists are supposed to present papers in a conference in a day. The day is divided into 2 slots, the morning slot and the evening slot. M scientists present their paper in the morning slot and the remaining in the evening slot. Both slots are separated by a lunch break.

Some scientists depend on a paper from some other scientists to be presented before theirs. So if Scientist A is presenting a paper on "Graph Theory" and Scientist B on "Max flow-Min cut", then A has to present before B. Lunch break is a time of merry making and partying, so attendees tend to forget the papers in the previous half. Due to this, the dependent scientist (B in this case) has to present the paper in the same slot as the scientist on whom he is dependent (A in this case). Given the dependencies, find the number of possible orderings of presenting the papers.

Input

The first line of input will contain an integer T <= 20 denoting the number of test cases.
Each test case will be formatted as follows:-
The first line will contain an integer denoting 1<= M <= 8.
The next 2M lines will contain 2M characters each. Each character will either be 'Y' or 'N'. If the i th line's j th character is 'Y' it means that scientist i is dependent on scientist j. 'N' signifies no dependence. A scientist will never be dependent on himself.

Output

Output one line per case that contains an integer denoting the number of possible ordering of scientists.

Sample Input

```
3
2
NNNN
NNNN
NNNN
NNNN
2
NYNN
NNNN
NNNY
NNNN
2
NYYY
YNYY
YYNY
YYYN
```

Sample Output

```
24
2
0
```

C - Gold Digging

Time Limit: 4 seconds

Mr. Goldust (1817-1890) was one of the first gold prospectors in the California Gold Rush. He literally struck gold there and became the owner of a few hundred gold pits. Mr. Goldust's great great great grandson Mr. Stardust currently owns the gold pits. Most of the gold has been dug up already, so Mr. Stardust wants to finish digging and get going to Las Vegas. A corporate giant offered to help him by supplying machines.

The machines are worth their weight in gold, so Mr. Stardust can only buy exactly one such machine. This machine was built using advanced science and thus does not work unless given appropriate working conditions.

Each day, Mr. Stardust will assign the machine to exactly one of the gold pits. If he assigns it to pit 'i', two things can happen:

- The machine will break down - with probability bi (0 < bi <= 1) . The machine cannot be used any more
- The machine will extract gold - with probability (1-bi) the machine will extract a proportion ri (0 <= ri <= 1) of the gold remaining in pit i

Theoretically the machine can last forever or break down very soon. So, Mr. Stardust's plan is to wait suitably long and then take off to Las Vegas. Of course, he will end up broke if the machine breaks down on day 1. He first needs to know how much gold he can expect to get using the machine optimally, that is, the best expected value of gold Mr. Stardust can achieve with an optimal strategy of allocating the machine to the pits.

Input

The input consists of a sequence of cases.
Each case starts with N (0 < N <= 100) on a line, representing the number of gold pits. Following this line are N lines, each one describing one pit. The ith line has three integers xi, yi and gi where bi = xi/100 , ri = yi/100 and gi is the amount of gold in pit i. (1 <= xi <= 100 , 0 <= yi <= 100 , 1 <= gi <= 100).
The last case will be followed by a -1. This case should not be processed.
There will be a maximum of 50 test cases.

Output

Output one case per line, the best expected value of gold that can be obtained, rounded to six decimal places.

Sample Input

```
1
50 100 100
1
50 50 100
2
50 100 100
50 50 100
-1
```

Sample Output

```
50.000000
33.333333
66.666667
```

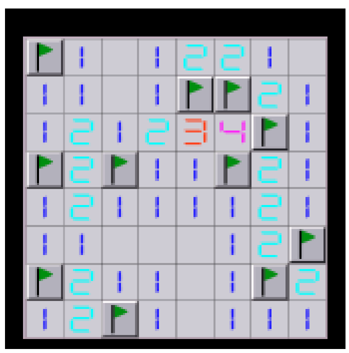
Explanation

In the first and second sample cases, keep assigning the machine to the only pit available. In the third sample case, clearly it is better to assign the machine to pit 1 on day 1 and if it survives, assign it to pit 2 from day 2 onwards.

D - Minesweeper

Time Limit: 4 seconds

Minesweeper is a single-player computer game. The objective of the game is to clear an abstract minefield without detonating a mine. When the game is started, the player is presented with a grid of **n** x **m** blank squares. If the player clicks on a square without a mine, a digit is revealed in that square, the digit indicating the number of adjacent squares that contains mines. Two squares are adjacent if they share an edge or a corner, i. e. a square can have at most 8 adjacent squares. By using logic, players can in many instances use this information to deduce that certain other squares are mine-free (or mine-filled), and proceed to click on additional squares to clear them or mark them with flag graphics to indicate the presence of a mine.



Clark Kent is a Minesweeper addict. And with help from his Kryptonian (a planet far far away from earth) powers he solves them at lightning speed and gives them to you. Your job is to tell him whether the solved version is correct or not. A board is correctly solved iff all flagged squares **should** contain a mine and every square containing a number **X** has exactly **X** adjacent squares flagged.

Input

The first line of input will contain an integer T <= 20 denoting the number of test cases.
Each test case will be formatted as follows:-

- The first line will contain two integers separated by a single space denoting 1<=**n**<=20 and 1 <= **m**<=20 respectively.
- The next **n** lines will contain **m** characters each. Each character will either be a digit (0 to 8 inclusive) or 'F'. The presence of 'F' indicates that Clark has flagged the square. The digits indicate the number of mines in the adjacent squares.

Output

Output one line per case:-

- 'Well done Clark!' if the board was solved successfully.
- 'Please sweep the mine again!' otherwise.

Note that quotes are for clarity only.

Sample Input

```
2
8 8
F1012210
1101FF21
121234F1
F2F11F21
12111121
1100012F
F21101F2
12F10111
8 8
F1012210
1101FF21
121234F1
F2FF1F21
12111121
1100012F
F21101F2
12F10111
```

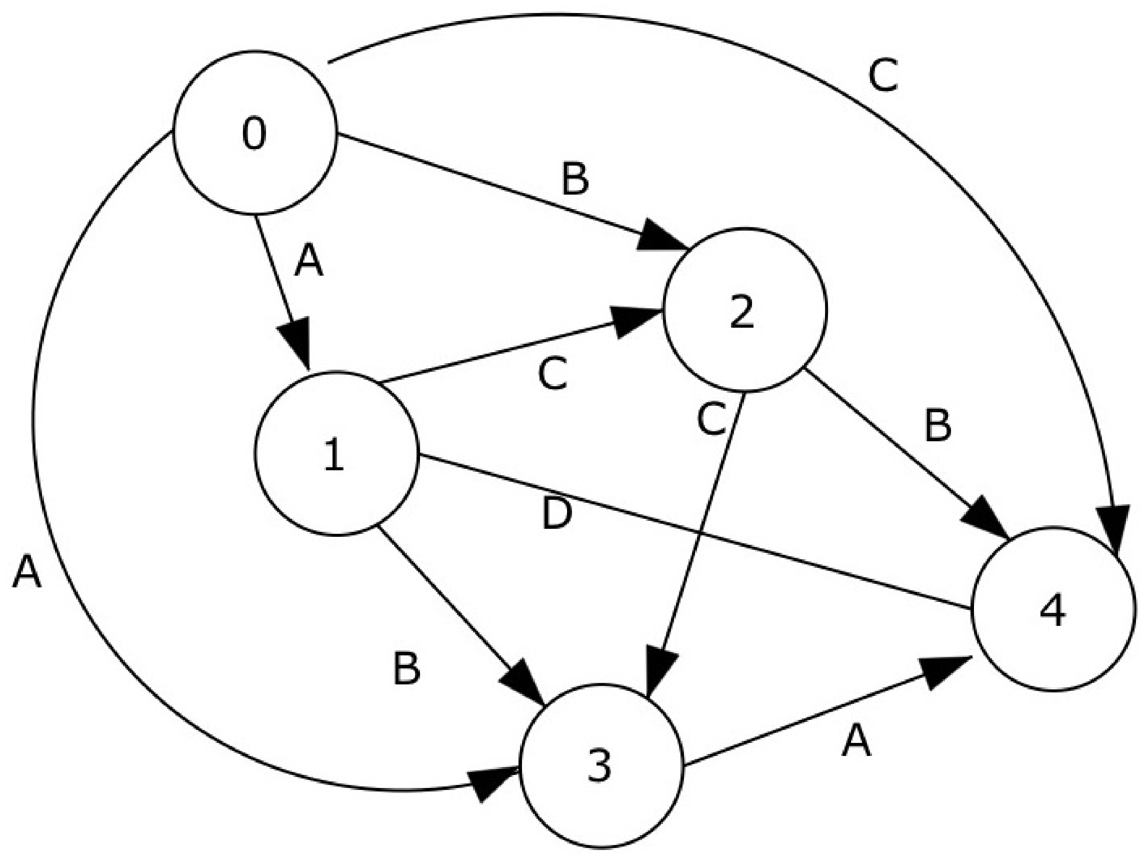
Sample Output

```
Well done Clark!
Please sweep the mine again!
```

E - Palindromic paths

Time Limit: 3 seconds

In Ragannagar (a small town in India), people are obsessed with palindromes . There are **N** road junctions (also called points) labeled **0** to **N-1** and roads exist between every pair of points. Roads are onewayed and for the road connecting point **i** to point **j** (**i** < **j**) the direction to travel is **i** to **j.** Each road is labeled with a letter between 'A' to 'Z' . Rajar ,the traveler, wants to travel from point **0** to point **N-1**. However he wants to cover the longest palindromic path.



In the above arrangement the possible paths to take are:-

- ACCA
- ABA
- ACB
- BCA
- AD
- BB
- AA
- C

The largest palindrome amongst these is ACCA, so Rajar will take this path. Given the above configuration, help him decide which path to take.

Input

The first line of input will contain an integer denoting the number of test cases **T** <=25. Each test case will be formatted as follows:-

- The first line of each test case contains an integer denoting 2<= **N** <=50.
- The next **N** lines contain **N** characters each. Each character is a letter between 'A' to 'Z'. The jth character in the ith line denotes the label for the road between i to j and this will be equal to the ith character in the jth line. The ith character of the ith line will be * denoting no road exists.

Output

Output one line per case -
The longest palindromic path available or "NO PALINDROMIC PATH" if none exists.
Note that quotes are for clarity only.
In case more than one longest path exists output the lexicographically smallest one.

Sample Input

```
2
5
*ABAC
A*CBD
BC*CB
ABC*A
CDBA*
5
*AXYZ
A*BQR
XB*BT
YQB*A
ZRTA*
```

Sample Output

```
ACCA
ABBA
```

F - Pile it down

Time Limit: 6 seconds

Sita and Gita are smart kids. Generally after completing their homework they go out for some outdoor games. But it's raining today and they have to stay home. They decide to break their piggy-bank and count their savings. They find out that they have accumulated many coins over the last few months and decide to play a game with these coins. First they divide the coins into 2 piles containing **X** and **Y** coins respectively.

Now they start the game by alternating turns. In each turn a player can do one of the following:

- Remove any number of coins from a single pile.
- Remove equal number of coins from both piles.
- Pass on the turn to the next player. Note that this still counts as a turn.

The game ends when no move is possible and the player who cannot make a move loses. Both players play optimally. Being smart, both players calculate the outcome of the game before the game begins. The player who loses tries to maximize the number of turns in the game and player who wins tries to minimize the turns. No player can pass more than **P** times. Sita starts the game.

Input

The first line of input will contain an integer T <= 200 denoting the number of test cases.
Each test case will contain a single line formatted as follows:-

```
X Y P
```

0<=X<= 1000
0<=Y<= 1000
0<=P<= 100

Output

Output one line per case.
The name of the winner and the number of moves in the game separated by a single space.

Sample Input

```
2
3 4 0
4 5 1
```

Sample Output

```
Sita 3
Sita 5
```

G - Dice Poker

Time limit: 6 seconds

Dice Poker is a version of Poker that is played using Dice. Two or more people can play this game. The detailed rules of the 2 player version of the game are mentioned in the section: Rules of the Game.

Two players A and B are playing the game of Dice Poker. Being experts in this game, both always play optimally. A always plays to maximize his chance of winning, and B tries to minimize the chance of A winning (draw is fine for B). In the first round, Player A rolls the Dice, followed by player B. You as an onlooker, look closely at the rolls and wonder what will A and B do next, and what is the probability of A winning. Haven't played this game much before, you decide to write a program to predict with what probability will A win this Game after the 2nd (final) round.

Rules of the Game:-

In the first round, two players start of by rolling 5 dice each, one after the other. For this problem we shall assume that all Dice are 6 sided and fair containing digits 1 to 6. Then a round of betting happens, which is irrelevant to this problem statement. In the second round the first player picks between 0 to 5 of his currently rolled dice and rolls them again. All the 5 dice are then merged (merge here simply means that all the 5 dice are taken together to form a set, including the ones rolled in the first round and are not re-rolled in the second) to form the final roll (or hand). Then the second player looks at the final hand of the first player and then similarly re-rolls between 0 to 5 dice to get her final hand. The hands are then compared and one with the higher weight wins.

In the real game, in case of a tie both the players share the pot, but since this problem specifically asks for the probability of A winning, we shall assume that a tie is considered as A not winning (hence B wins).

To determine which of the hands has higher weight, following arrangements are considered and are in descending order of weight. The first arrangement beats the second and so on.

Arrangements:

- 5 of a kind: All the 5 dice show the same number. Example: 3, 3, 3, 3, 3
- 4 of a kind: 4 of the dice show the same number. Example: 2, 2, 2, 2, 5
- Full house (3 of a kind + 2 of a kind): 3 of the 5 show the same number and the rest 2 show a different number. Example: 4, 4, 4, 6, 6
- Straight: All the 5 dice are in a sequence. Example: 1, 2, 3, 4, 5 (or the other possibility is 2, 3, 4, 5, 6)
- 3 of a kind: 3 of the dice show the same number. Example: 6, 6, 6, 2, 4
- 2 Pair: There are 2 pairs. Example: 4, 4, 1, 1, 3
- Pair: 2 of the dice show the same number. Example: 3, 3, 1, 2, 4
- Rest: Example: 1, 2, 3, 4, 6

The highest weight arrangement that suits the hand is considered. For example a 5 of a kind is also a 4 of a kind, but the player would naturally call it 5 of a kind. Also in a particular arrangement, the higher weight arrangement is the one with the highest value of the most significant part of the arrangement. If a the tie still remains then it is broken by the next significant part and so on. For Example, In 4 of a kind the most significant part are the 4 dice with the same number. So the arrangement 4, 4, 4, 4, 1 is better than 3, 3, 3, 3, 6. In 3 of a kind the 3 with the same number is the most significant part. However in a 2 pair, there are 2 equally significant parts, and the one with the highest number will be considered first for breaking the tie. In the 'Rest' arrangement, first compare the highest numbers of the 2 hands, then the 2nd highest and so on.

Input

First line contains T (T <= 10) the number of test cases to follow For each case, the first line contains 5 space separated numbers, the arrangement of the dice after the first round for player A. The second line contains 5 space separated numbers, the arrangement of the dice after the first round for player B. Each of the numbers will be between 1 and 6 inclusive.

Output

Per test case output a single number in a line, giving the probability of A winning. You will need to round it off to 6 decimal places. (0.0000005 -> 0.000001 and 0.0000004 -> 0.000000). Always output exactly 6 decimal places.

Sample Input

```
5
6 6 6 6 6
1 2 3 4 5
1 2 3 4 5
2 3 4 5 6
1 2 3 1 1
1 2 3 4 5
1 3 1 2 2
2 2 3 4 6
1 1 1 1 1
5 4 5 5 5
```

Sample Output

```
0.999871
0.056503
0.430834
0.477623
0.833333
```

Explanation of the first case

A already has the best possible set (all 6's) so he won't roll again. B has no chance of winning. However she can probably force a draw by throwing all 6's. So the only option for her is to re-roll all the 5 dice. The numbers of possible outcomes are 6*6*6*6*6 out of which only 1 will force a draw. So the answer is 1 - (1/6^5).

H - Advise National Security!

Time limit: 10 seconds

It's 2015. Terrorists are still largely on the prowl. Governments however have decided to be smarter. Now, they monitor each higway, airway and seaway. Across each road, they built several cameras which can capture images ahead of them.

More specifically, there are N uniformly spaced cameras per kilometer along a highway of length M kms, making in total M*N cameras (there is no camera at the end of the M kms). The highway is oneway and goes from North to South. The cameras have a special property:

- They use direct vision to watch the next N (or less if the highway ends) cameras to their South thus covering 1 km.
- They communicate via satellite with all the cameras beyond the next N southern cameras thus knowing their video feed.
- They do not communicate with the next N southern cameras and only depend on their vision.

A camera being destroyed by a terrorist will be caught by cameras to its north. Corrupt politicians have sold this secret to the terrorists and they know that they can't destroy a camera unless they are sure that this camera is no longer communicating or being watched by a camera to its North. There is one further complication: Corrupt technicians have not properly installed the direct vision equipment. Thus, some of the cameras are substandard and deficient. These cameras have perfectly good communication with far off cameras (i.e. after the first N cameras), but cannot see some of the next N cameras. However, the saving grace was that there were no more than 10 deficient cameras in any 1 km stretch (among any contiguous N cameras).

As Anti Corruption Task force, you, a non corrupt patriot have to submit a security report. For that, you need to solve the following problem: If **exactly two** terrorists decide to destroy all the cameras on the highway without being caught, how long would it take them? Each terrorist can destroy one camera in one FULL minute. They can work simultaneously. Of course, they cannot destroy two cameras A and B at the same time if A can watch B or B can watch A.

The cameras are numbered 1 to M*N North to South.

Note that Ci cannot watch or get the video feed of Cj if i > j where Ci is the camera numbered i.

Input

Input will be a sequence of cases. Each case starts with M and N on a single line (1 <= M <= 15 ,1 <= N <= 20). M*N - 1 lines follow. The Cth line describes camera number C. It starts with a

number k. If k = -1 camera C is not deficient and there are no more numbers on this line. Otherwise, k numbers, aj (1<=j<=k) (C+1 <= aj <= C+N and aj <=M*N) follow on the same line meaning that camera C can watch camera aj. Note that camera C can always watch cameras C + N + 1 and later using satellite and will not be mentioned here.

The last case will be followed by a line containing two zeroes.

Output

Output one line per case, the minimum number of minutes required by two terrorists working in tandem to destroy all the cameras without being caught.

Sample Input

```
2 3
1 4
2 4 5
2 4 5
1 6
-1
0 0
```

Sample Output

```
4
```

Explanation

Camera i is referred to as Ci.
There are a total of 6 cameras, 3 per km.
C1 can watch C4 and later but not C2 or C3
C2 can watch C4 and later, but not C3
C3 can watch C4 and C5 but not C6
C4 can watch C6 but not C5
C5 can watch C6
C6 cannot watching anything.
T = 1: Destroy C1 and C2
T = 2: Destroy C3
T = 3: Destroy C4 and C5
T = 4: Destroy C6

I - Find Terrorists

Time limit: 5 seconds

The Prime Minister and his Accumulated Council of Ministers(ACM) are trying hard to find all possible terrorist locations. In his dream, the Prime Minister gets a message from God suggesting that the answer to all terrorist problems are numbers (say one such number is X) such that the number of factors of X(including 1 and X) is prime. These numbers supposedly contain the encrypted locations of terrorists. Since the ACM has no programmer, the Prime Minister needs your help in finding out such numbers.

Note: 1 is not considered a prime number.

Input

The first line of input will contain an integer T <= 20 denoting the number of test cases.
T lines follow, one per test case.
Each test case will be a line formatted as "L H" where L and H are integers and 0<=H<=10000

Output

Output one line per case a space separated list of all integers(sorted ascending) lying between L and H (both inclusive) such that the number of factors of each integer is prime. In case no such integer exist output -1.

Sample Input

```
3
1 1
1 2
2 5
```

Sample Output

```
-1
2
2 3 4 5
```