VCU School of Engineering
2015 Computer Science Programming Contest

Welcome to the 2015 Programming Contest. Before you start the contest, please be aware of the following:

1. There are ten (10) problems in the packet, using letters A-J. These problems are NOT sorted by difficulty. When a team's solution is judged correct, the team will be awarded a balloon. The balloon colors are as follows:

Problem	Problem Name	Balloon Color
A	The Daycare	gold
B	Efficiency	lavender
C	Encryption at its Finest	lime green
D	Dolphins	orange
E	Mutual Funds	pink
F	The Computer Science Department's Holiday Party	powder blue
G	A Farmer's Effort	silver
H	The Movie Festival	teal green
I	Staircase Intrigue	white
J	Doublets	yellow

2. All solutions must read from standard input (System.in) and write to standard output (System.out).

3. Solutions for problems submitted for judging are called runs. Each run will be judged. Runs for each particular problem will be judged in the order in which they are received. However, it is possible that runs for different problems may be judged out of order. For example, you may submit a run for B followed by a run for C, but receive the response for C first. DO NOT request clarifications on when a response will be returned. If you have not received a response for a run within 10 minutes of submitting it, **you may ask the lab monitor to send a runner to the judge to determine the cause of the delay. Under no circumstances should you ever submit a clarification request about a submission for which you have not received a judgment.**

The judges will respond to your submission with one of the following responses. In the event that more than one response is applicable, the judges may respond with any of the applicable responses.

Response	**Explanation**
Yes	Your submission has been judged correct.
Wrong Answer	Your submission generated output that is not correct.
Output Format Error	Your submission's output is not in the correct format, is misspelled, or did not produce all of the required output.
Excessive Output	Your submission generated output in addition to what is required.
Compilation Error	Your submission failed to compile.
Run-Time Error	Your submission experienced a run-time error.
Time-Limit Exceeded	You submission did not solve the judges' test data within **2 minutes**.

4. A team's **score** is based on the **number of problems they solve and penalty points**. The penalty points reflect the time required to solve a problem correctly and the number of incorrect submissions made before the problem is solved. For each problem solved correctly, penalty points are charged equal to the time at which the problem was solved plus 20 minutes for each incorrect submission. No penalty points are added for problems that are never solved. Teams are ranked by the number of problems solved. Ties are resolved in favor of the team with the fewest penalty points.
5. This problem set contains sample input and output for each problem. However, you may be assured that the judges will test your submission against several other more complex datasets, which will not be revealed. Before submitting your run you should design other input sets to fully test your program. Should you receive an incorrect judgment, you are advised to consider what other datasets you could design to further evaluate your program.
6. In the event that you think a problem statement is ambiguous, you may request a clarification. Read the problem carefully before requesting a clarification. If the judges believe that the problem statement is sufficiently clear, you will receive the response, "The problem statement is sufficient, no clarification is necessary." If you receive this response, you should read the problem description more carefully. If you still think there is an ambiguity, you will have to be more specific or descriptive of the ambiguity you have found. If the problem statement is ambiguous in specifying the correct output for a particular input, please include that input data in the clarification request.
7. The submission of abusive programs or clarification requests to the judges will be considered grounds for immediate disqualification.
8. Good luck and HAVE FUN!!

Problem A
The Daycare

The daycare Nannies R Us has installed a new high-tech system for keeping track of which children are under its supervision at any moment in time. Each parent of a child has a card with a unique ID, where the ID is some integer x. Every morning, the parent swipes the card at a card-reader when dropping their child off at the daycare. The system then adds x to an array of integers. In the evening, when the parent comes to pick up the child, the same card is swiped, and a second copy of x is added to the list. Each night at midnight, the array is wiped clean and reset to have zero elements. In this way, the daycare can quickly check every afternoon which children are yet to be picked up by their parents by picking out the array elements that occur precisely once. Unfortunately, at 4:55 pm on Friday, March 13, 2015, the daycare realizes that somewhere in their daycare is precisely one child who is yet to be picked up – can you help them determine the integer ID of this child?

Input
The input consists of two rows. On the first row, an integer n satisfying 1<= n <= 10,000 is given. On the second row, a sequence of n integers is given.

Output
The ID of the child left in the daycare.

Sample Inputs

```
7
1 1 2 9 -11 -11 9

9
1 2 1 3 2 4 3 4 5
```

Sample Outputs

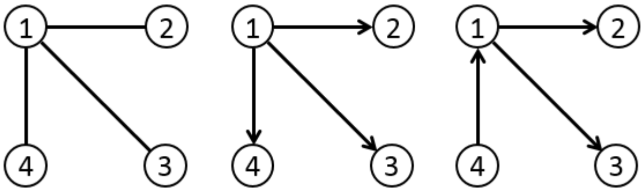
```
2

5
```

Problem B
Efficiency

Byteland is a very efficient country. The cities are connected by two-way roads, in a way that is very… efficient: for every pair of cities, there is only one route between them, either direct or going through other cities. Still, any city can be reached from any other city.

The youth of Byteland decided to rebel against all that efficiency. They devised a prank: on each road connecting a pair of cities, they will post a sign, making it a one-way road. If they go with their plan, some cities will not be reachable from some other cities.

The government learned about the plan, and is assessing the potential damage. Each road will be one-way, but which way? The government agents haven't learned that yet, and there are so many possible configurations… To assess the best case scenario, they ask: What configuration of one-way roads leads to largest number of ordered pairs of cities (A, B) such that one can still go from A to B? As for the worst case scenario, they ask: Which configuration leads to the fewest number of available city-to-city routes? For example, for 4 cities connected as shown below, the worst case is only 3 pairs of cities (1→2, 1→3, 1→4), while the best case is 5 pairs of cities (1→2, 1→3, 4→1, 4→2, 4→3).



Road network Worst case (3) Best case (5)

Your task is to assess the situation and, given the map with all the roads, calculate the highest and lowest possible number of ordered pairs of cities A, B such that one can still go from A to B once all roads are converted to one-way roads.

Input

The first line specifies the number of cities, 2<=n<=1000.
Each of the next n-1 lines contains two integers in the range 1…n, indicating two cities that are linked by a direct bi-directional (before the prank) road.

Output

On a single line, print two numbers: The smallest and the largest possible number of ordered pairs of cities such that one can still go from one to the other (after the prank).

Sample Input

```
4
1 2
1 3
1 4
```

Sample Output

```
3 5
```

Problem C
Encryption at its Finest

Alice recently read the news about Edward Snowden, and learned that encryption could be used to protect personal communication effectively. She decided to construct a simple encryption scheme to protect her own communications as follows. First, she generates a set of randomly chosen “secret keys” K1, K2, K3. Next, using these keys, she encrypts a set of messages M1, M2, M3, obtaining ciphertexts C1= K1 + M1, C2= K2+ M2, C3 =K3+M3.

In implementing her scheme, she decides to choose the secret keys from the set A={1, 2, …, 100}, and the messages from the sequence B=(a, b, c, …, z). She is adamant that her ciphertexts should *also* be elements of the sequence B; to this end, she defines the ``+’’ operator in expressions such as C1=K1+M1 to *shift* element M1 in B to the right by K1 positions. For example, if K1=1 and M1= b, then the corresponding ciphertext is C1= c. Similarly, if K1=2 and M1=b, then the ciphertext is C1=d, if K1=24 and M1=b, the ciphertext is C1=z, and if K1=25 and M1=b, then the ciphertext is C1=a.

Input

The input consists of four rows.

- The first row is an integer n satisfying 1 <= n <= 100 specifying the number of secret keys.
- The second row is a sequence of n integers, each from the set A={1,2,…,100}, corresponding to the secret keys.
- The third row is a sequence of symbols from the set B={a,b,…,z} corresponding to the messaged to be encrypted.
- The fourth row is a special value -1 to indicate the end of the input phase.

Output

Output a single row of symbols from set B corresponding to each of the ciphertexts.

Sample Input

```
3
1 2 3
bzy
-1
```

Sample Output

```
cbb
```

Problem D
Dolphins

The Royal Bureau of Cyphers came up with the following cypher. For each day of the year, there is a secret integer code, known to the diplomats and the King. When a diplomat wants to send a message M to the King, e.g. M: "*riot expected*", the embassy will prepare and send two separate letters, A and B, for example A: "*the king might visit soon fall is a riot of color here expected to last till november*" and B: "*the king might postpone visit after fall carnival is an absolute riot here crowds expected*".

Always, the texts A and B both contain at least one copy of all the words that occur in the message M, in the same order as in M. That is, M is an ordered-preserving subset of words in A, and an order-preserving subset of words in B. M should not be the only message that meets the above criterion – the King affectionately calls all messages that meet it 'Dolphins'. For example, for the texts A and B above, "*riot expected*" is a Dolphin, but there are other Dolphins: "*king might fall*", *"fall is expected"* and "*king might visit*", among others. Upon receiving a pair of letters A and B from an embassy, the King will find all Dolphins, sort them alphabetically, and use the secret integer code to pick from the order the proper Dolphin. Using this protocol faces a danger, though. If the list of Dolphins for texts A and B is short, the eavesdroppers may guess the original message even without knowing the secret code.

Your task, as the Court Programmer, is to come up with an algorithm that will count how many Dolphins are there for two given letters A and B. More specifically, for two input texts, you should count possible lengths of all Dolphins, and count how many Dolphins of each length are there. Length of a Dolphin is just the number of words in it. For example, if the two texts are: "*the king went fishing yesterday afternoon*" and *"yesterday afternoon the king saw a man fishing off his pier*", there is one Dolphin of length three: "*the king fishing*", four Dolphins of length two: "*the king*" and "*the fishing*", "*king fishing*" and "*yesterday afternoon*", and five Dolphins of length one: "*yesterday", "afternoon", "the*", "*king*" and "*fishing*".

Input
The input consists of two lines, the first line is the string A and the second line is the string B. You may assume that the strings consists of only lowercase characters and spaces, the length of each string is at most 1000 characters, the length of each word is between 1 and 10 characters, and no word will occur twice in a single string.

Output
On a single line, print two numbers: Dolphin length (number of words), and the number of different Dolphins of that length. The lines should be sorted in descending order by Dolphin length.

Sample Input
```
the king went fishing yesterday afternoon
yesterday afternoon the king saw a man fishing off his pier
```

Sample Output
```
3 1
2 4
1 5
```

Problem E
Mutual Funds

Your pet elephant has just passed away, and in its will, has left you $500. In your jubilation, you have decided to invest your money into mutual funds with the hope of slowly growing your windfall over the years. Specifically, you plan to choose one mutual fund, and on January 1 of each year, you plan to invest an additional $100 of your money into that fund. After five years, you plan to cash in all your funds. The problem is, there are many mutual funds to choose from – which one will give you the best return for your money?

Of course, you cannot see the future, so instead, you will make your prediction based on the last 5 years of each fund's performance. Specifically, the way the funds work is as follows. For any given fund, let y_i denote the i-th year of the past 5 years of the fund, where 1<=i<=5. Then, in each year y_i, the fund returned some (possibly negative) percentage p_i. In addition, each fund charges a "manager's fee", which is stated as an annual percentage and is charged on the balance of your current investment on December 31 of each year. For any given fund, you would hence like to know your total return after all these factors are considered, stated as a percentage, after 5 years.

Input

The input is specified as follows.

- The first row of input contains the number of mutual funds, n, which you will consider. You may assume 0<= n <= 100.
- For each mutual fund, you will then read in two lines.
  - The first of these lines is a list of five real numbers, where the i-th real number corresponds to the annual return of the fund in year i, stated as a percentage.
  - The second line contains the annual manager's fee for that fund, also stated as a percentage.

  You may assume that all percentages are given with at most two decimal places.

Output

The output consists of two rows:

- The first row is a list of n real numbers, each rounded to one decimal place, with the *j*-th number corresponding to the total return of your investment at the end of the 5 years under mutual fund *j*, stated as a percentage.
- The second row contains the index of the mutual fund which returned the largest amount after five years.

Sample Input

```
2
3 5.1 -7.6 2 4
1.2
2 -1 5 4 -2
0.89
```

Sample Output

```
-0.2 1.4
2
```

Problem F
The Computer Science Department’s Holiday Party

The Computer Science department at VCU plans to hold a holiday party for its students. The department will provide a menu of candidate dishes to the students, and each students will vote as follows: Each student picks two dishes from the menu, and for each pick, states whether he or she *likes* the dish or *dislikes* the dish. Note that the two picks need not be distinct, and in fact, there is the possibility that a student might pick a single dish twice and vote *like* the first time and *dislike* the second time.

Your task is to write a program which behaves as follows: First, you must read in the students' wishes. Then, you must determine whether there exists a list of dishes to be served which satisfies at least one vote of each student. Note that some students might have two *like* votes or two *dislike* votes – thus, choosing all the dishes or none of the dishes may not satisfy at least one vote per student.

Input

The input contains multiple test cases. Each test case contains the following lines

- The first line contains two positive integers n and m, satisfying $0 \leq \ n \leq \ 30\,000, 0 \leq m \leq 10\,000$. Here, n is the number of students, and m is the number of dishes on the menu. The students are numbered from 1 to n, and the dishes are numbered from 1 to m.
- Each of the next n lines contains two non-zero integers, separated by single space. The $(i+1)$-st line contains numbers w_i and v_i representing the votes of the i-th student, $-m \leq \ w_i \, , v_i \ <= m; w_i \, , v_i \neq 0$. A positive number means that the student likes the dish specified by the absolute value of the number, and a negative number means the student dislikes the dish.

The final test case contains only one line “0 0”. No output is expected for this test case.

Output

Your program should output “YES” if it is possible to form a list of dishes satisfying at least one vote per student, and “NO” otherwise.

Sample Input

```
2 3
-1 -2
1 2
3 3
-1 2
-1 -3
1 1
2 5
-1 -1
1 1
0 0
```

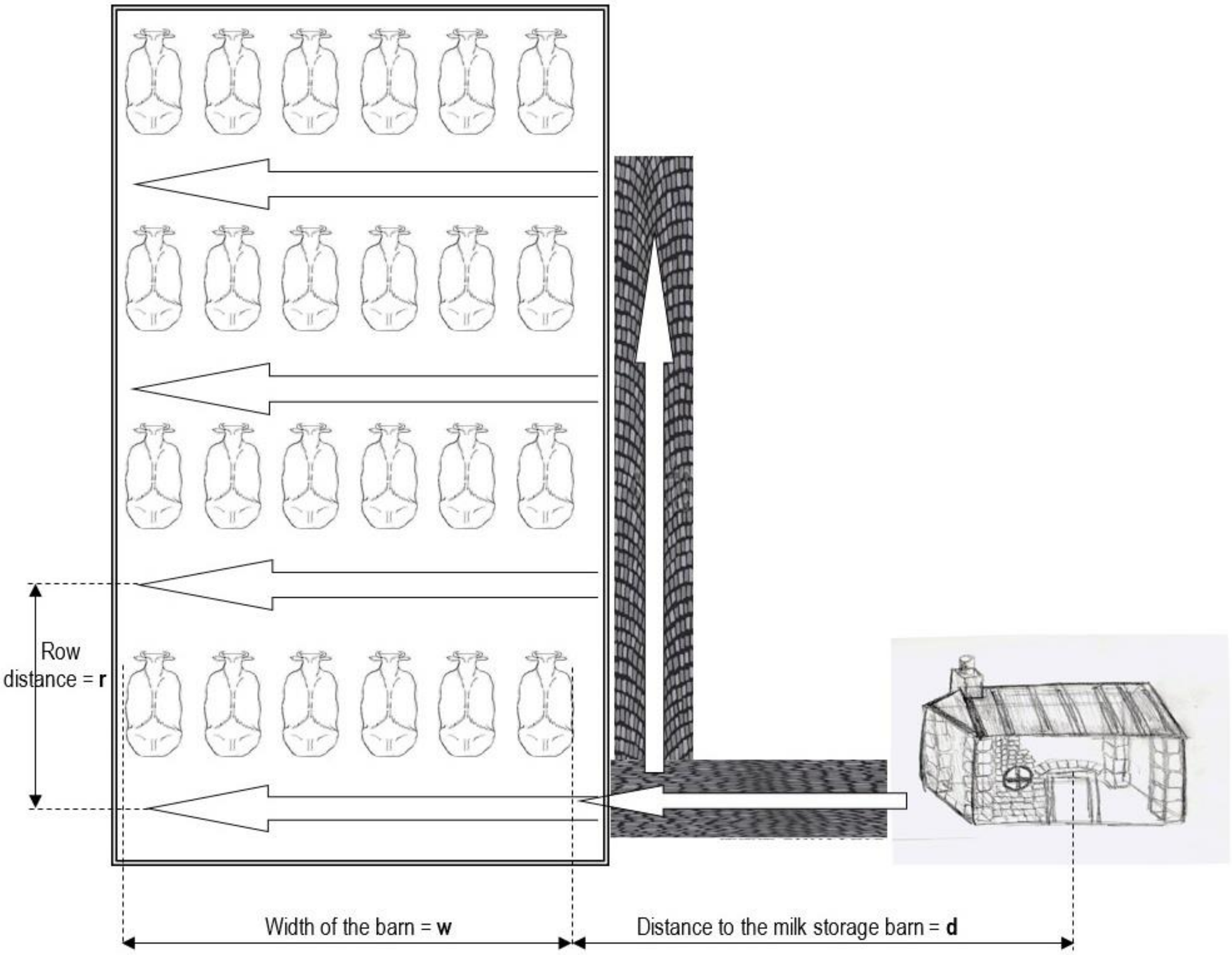
Sample Output

```
YES
YES
NO
```

Problem G
A Farmer's Effort

On the outskirts of Richmond, Virginia, lies a dairy consisting of a cow's barn and a milk storage facility. Each morning, the farmer rises at 5 am to milk his cows. Unfortunately, his milk container is only large enough to hold the quantity of milk produced by a single row of cows in the barn. As a result, he has a lot of walking to do, as illustrated in the figure below. Your task is to determine just how *much* walking the farmer needs to do each morning to milk all his cows.

Specifically, you will be given as input the following parameters: (1) The number of rows of cows n, (2) the width of the barn w, (3) the distance from the barn to the storage facility d, and (4) the distance r between the rows of cows in the barn. Your task is to compute how many meters the farmer has to walk each morning to milk all the cows. Assume that he begins his walking each day at the storage facility with an empty milk container, and that he finishes back at the storage facility to empty the container full of milk from the n-th row. You should assume that n is a long integer, 1<=n<=1,000,000,000. The other three parameters are also long integers; these latter three having maximal value 100.


Row
distance = **r**
Width of the barn = **w**
Distance to the milk storage barn = **d**

Input

The input is specified as follows:

- The first row specifies the number of milk farms to consider.
- The second row contains the number of rows of cows, n, the width of the barn, w, the distance from the barn to the storage facility, d, and the distance between the rows of cows, r. All distances are specified in meters.

Output

For each farm, output the distance in meters that the farm's farmer must walk each morning to milk the cows.

Sample Input

```
3
100 20 15 4
25000 18 20 3
12500 10 10 2
```

Sample Output

```
46600 m
1876825000 m
312975000 m
```

Problem H
The Movie Festival

You're anxiously awaiting Richmond's upcoming Independent Movie Festival - there are so many great films to see! Unfortunately, the movies are screened at several stages, and the films often have overlapping running times. We say that two movies do not overlap only if one of them ends before the other starts. Your task is to write a program that takes a list of movies L you would like to see as input, and outputs the largest number of movies from L which can be seen without causing any time overlaps.

<u>Input</u>

The input is specified as follows:

- The first row contains integer n (2<=n<=100) that represents the number of movies on your list.
- The next n rows each contain two integers separated by a space, where the integers represent the start and finish time of one movie from the list as follows:
  - The first integer (>=0; <=1440) identifies the starting time for the movie, in minutes after midnight.
  - The second integer (>=0; <=1440) identifies the finish time for the movie, in minutes after midnight. The second integer is greater than the first.

<u>Output</u>

The output is a single positive integer representing the largest number of non-overlapping movies that can be selected from the input list.

<u>Sample Input</u>

```
4
650 840
610 700
710 835
835 900
```

<u>Sample Output</u>

```
2
```

Problem I
Staircase Intrigue

Your sister has Compulsive Climbing Disorder, and becomes fixated with the following game whenever she encounters a set of stairs. Each time she reaches a step in the staircase, she decides whether she wants to climb one step or two steps next. Whenever she reaches the top of the stairs, she returns to the bottom and repeats the game, this time choosing a new sequence of step sizes. She concludes the game when she has tried all possible sequences of 1- and 2-steps for climbing the staircase. Given the number of steps n in the staircase, your job is to determine the number of such distinct sequences.

Input
The input is an integer n satisfying 1 <= n <= 10,000.

Output
An integer representing the number of possible sequences of 1- and 2-steps for climbing an n-step staircase.

Sample Inputs

```
1

3

6

9
```

Sample Outputs

```
1

3

13

55
```

Problem J
Doublets

“Doublets” is a word game invented by Lewis Carroll in 1877. Given a start word X and an end word Y, the goal is to transform X into Y by changing one letter at a time. A dictionary of usable intermediate words is provided. In each iteration i of the game with current string X_i, precisely one letter can be changed to transform X_i into one of the given dictionary words. Here, X_0 indicates the start word, X, and each subsequent string is a word from the dictionary. This iterative process continues until a single letter transformation from the current intermediate string X_i to the end word is possible.

Input

The input is specified as follows:

- The first row specifies the number of instances of Doublets you will solve, given as an integer 1<=n<=100.
- The following lines specify each of the instances to be solved as follows:
    - The first row specifying an instance contains the start word, end word, and an integer 1<=m<=10,000, where m is the number of words in the dictionary
    - On each of the m subsequent rows, a dictionary word for the puzzle is given. All words for a puzzle will be the same length, and consist of lower-case alphabetic Latin characters (a,b,…,y,z).

Output

A single integer for each instance on its own row, which represents the minimum number of intermediate dictionary words that must be used to transform the start word into the end word for the respective instance, in order.

Sample Inputs

```
1
man ape 6
mat
oat
apt
opt
ore
are
```

```
2
bat hit 3
hot
hat
cat
hot dog 3
don
hit
dot
```

Sample Outputs

```
4
```

```
1
1
```