2011 Southwestern Europe Regional Contest Problems

Hosted by

Facultad de Informática. Universidad Complutense de Madrid

Contents

A
Alphabet Soup

Peter is having lunch at home. Unfortunately for him, today's meal is soup. As Peter's mother is aware that he doesn't like it very much, she has cooked a special soup using pasta pieces shaped like letters from the alphabet, numbers and other characters. She has a special knife with which she can prepare an unlimited supply of pasta pieces that may come in S different forms. The soup always has P pasta pieces in it, and is so thick that the pieces never move.

Despite her efforts, Peter is still not happy with today's menu and asks how many days in his life he will have to eat soup. His mother promises him that she will prepare a different soup every day, and that on no day will the dish contain the same shapes in all positions as any soup dish previously served. However, the number P of pasta pieces, as well as the positions in which pieces float, will remain the same every day. Peter is not easily fooled (or so he thinks), and he cleverly realizes that this can still make him eat soup for ages. In an attempt to reduce the number of configurations, he tells his mother he will not accept any dish which can be obtained by rotating one of the configurations previously seen.

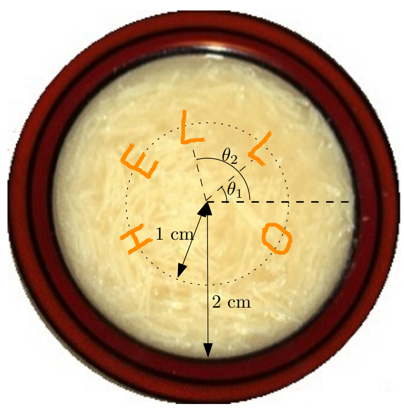


Figure 1: Top view of Peter's dish

Consider the dish as a circle of radius 2 centered at the origin. All the symbols will be floating in the soup at a given angle (in millidegrees) at distance 1 from the origin. Two plates are considered equal if you can perform a rotation of one of the dishes about its center so that the positions of the symbols, as well as the symbols themselves, are the same in both.

Your program will be given the number of possible symbols Peter's mother has available, and the angles determining the location of each of the pasta pieces (measured clockwise in millidegrees). Write a program that returns the number of possible plates Peter's mother can prepare. As this number can be very large, **output the number modulo** 100,000,007, which is prime.

Input

The first line of input in each test case contains two numbers: S ($2 \leq S \leq 1,000$), the number of symbols Peter's mother can use; and P ($P > 0$), the number of pasta pieces floating in the soup. Each of the next P lines contain the angle A ($0 \leq A < 360,000$) of one of the P pieces (measured clockwise in millidegrees). All angles will be different.

Different tests cases are separated by a blank line. After the last test case there is a line with $S = P = -1$.

Output

For each test case output a single integer in a line by itself, the number of different plates Peter's mother can cook modulo 100,000,007.

Sample Input

```
2 4
0
90000
180000
270000

100 5
0
45000
90000
180000
270000

-1 -1
```

Sample Output

```
6
99999307
```

B
Coin Collecting

As a member of the Association of Coin Minters (ACM), you are fascinated by all kinds of coins and one of your hobbies involves collecting national currency from different countries. Your friend, also an avid coin collector, has her heart set on some of your precious coins and has proposed to play a game that will allow the winner to acquire the loser's collection (or part thereof).

She begins by preparing two envelopes, each of them enclosing two coins that come from different countries. Then she asks you to choose one of the two envelopes. You can see their contents before making your choice, and also decline the offer and take neither. This process is repeated for a total of r times. As the game progresses, you are also allowed to change your mind about your previous picks if you think you can do better. Eventually, your friend examines the envelopes in your final selection, and from among them she picks a few envelopes herself. If her selection is non-empty and includes an even number of coins from every country (possibly zero), she wins and you must hand over your entire coin collection to her, which would make years of painstaking effort go to waste and force you to start afresh. But if you win, you get to keep the coins from all the envelopes you picked.

Despite the risks involved, the prospect of enlarging your collection is so appealing that you decide to take the challenge. You'd better make sure you win as many coins as possible.

Input

The first line of each test case is the number r of rounds ($1 \le r \le 300$); a line with $r = 0$ indicates the end of the input. The next r lines contain four non-negative integers $0 \le a, b, c, d < 10,000$, meaning that your friend puts coins from countries labelled a and b inside one of the envelopes, and c and d inside the other one. A blank line separates test cases.

Output

Print a line per test case containing the largest number of coins you are guaranteed.

Sample Input

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

6
1 4 1 4
2 4 2 4
0 3 0 3
0 4 0 4
4 3 4 3
1 3 1 3

0
```

Sample Output

```
6
8
```

C
Cybercrime Donut Investigation

Year 2042. The Internet has evolved to a virtual reality dataspace where crimes are committed every day. The 2041 SWERC winner developed an agent that drops a donut every time a crime is committed in the Cyberspace. Each of the donuts has its own signature. The Madrid Police have a huge database with crimes and their donut signatures.

Today is your day. Your task is to implement a new agent that looks for the records in the database that bear a strong resemblance to the given signature of a dropped donut found at a new crime scene.

Figure 2: The major piece of evidence for today's unsolved crime streak

Experts in virtual criminology have obtained the best similarity measure between donuts: compute the difference in radius of the internal part of the toroids (holes), compute the difference in radius of the external part of the toroids (tubes), and then add up those differences.

Input

The first line of each test case contains n ($1 \leq n \leq 100,000$), the number of donuts in the database. The i^{th} of the following n lines contains the radius of the hole and radius of the tube of the i^{th} donut in the database, described by two integers l and w ($1 \leq l, w \leq 10^9$). After that there is a line containing q ($1 \leq q \leq 50,000$), the number of donuts that you are looking for in the database. Then q lines follow, the i^{th} of them describing the dimensions of the newly found i^{th} donut in the same way.

Different test cases are separated by a blank line. A line containing -1 marks the end of the input.

Output

The output of your program on each test case should have q lines, each of them containing an integer. The i^{th} of these lines should contain the similarity between the newly found i^{th} donut and the donut in the database that most closely resembles it. Outputs for different test cases should be separated by a blank line.

Sample Input

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

2
1 1
9 9
4
4 5
6 5
2 5
3 4

-1
```

Sample Output

```
3
0

7
7
5
5
```

D
Distributing Ballot Boxes

Today, besides SWERC'11, another important event is taking place in Spain which rivals it in importance: General Elections. Every single resident of the country aged 18 or over is asked to vote in order to choose representatives for the Congress of Deputies and the Senate. You do not need to worry that all judges will suddenly run away from their supervising duties, as voting is not compulsory.

The administration has a number of ballot boxes, those used in past elections. Unfortunately, the person in charge of the distribution of boxes among cities was dismissed a few months ago due to financial restraints. As a consequence, the assignment of boxes to cities and the lists of people that must vote in each of them is arguably not the best. Your task is to show how efficiently this task could have been done.

The only rule in the assignment of ballot boxes to cities is that every city must be assigned at least one box. Each person must vote in the box to which he/she has been previously assigned. Your goal is to obtain a distribution which minimizes the maximum number of people assigned to vote in one box.

In the first case of the sample input, two boxes go to the first city and the rest to the second, and exactly $100,000$ people are assigned to vote in each of the (huge!) boxes in the most efficient distribution. In the second case, $1, 2, 2$ and 1 ballot boxes are assigned to the cities and $1,700$ people from the third city will be called to vote in each of the two boxes of their village, making these boxes the most crowded of all in the optimal assignment.

Input

The first line of each test case contains the integers N ($1 \leq N \leq 500,000$), the number of cities, and B ($N \leq B \leq 2,000,000$), the number of ballot boxes. Each of the following N lines contains an integer a_i, ($1 \leq a_i \leq 5,000,000$), indicating the population of the i^{th} city.

A single blank line will be included after each case. The last line of the input will contain `-1 -1` and should not be processed.

Output

For each case, your program should output a single integer, the maximum number of people assigned to one box in the most efficient assignment.

Sample Input

```
2 7
200000
500000

4 6
120
2680
3400
200

-1 -1
```

Sample Output

```
100000
1700
```

E
Game, Set and Match

In this problem you need to assist in computing the probability of winning at tennis. Here is a brief explanation of how the scoring system works. In a tennis *match*, players play a certain number of consecutive *sets*. Each *set* is in turn made up of a series of *games* (and may include a *tie-break* if needed). Finally each game is made of *points*.

Points. Every point is started by one of the players serving (i.e. hitting the ball into the service box in the opposite court) and the other receiving serve. The server then attempts to return the ball into the server's court and players alternate hitting the ball across the net. When one of the players fails to make a legal return (e.g. if the ball is knocked out of the court), he or she loses the point. The specifics of how points are won are not important to us.

Games. The scoring system within a game is peculiar to say the least. As the player wins points in a game, his score goes from the initial value of 0 (read "love") to 15, 30, or 40 (yes, just when you think you're starting to spot a pattern in this mess it breaks down). There is no a-priori limit to the length of a game (meaning the number of points played), but a player's score is always indicated by one of these numbers according to the following rules. When a player has three points (score 40) and wins the following point as well, he wins the game unless the scoreline was $40 - 40$ (read "deuce") to start with. A player needs to win two consecutive points from deuce to win the game. Winning one gives him advantage; if followed by a second winning point the game is won by him, but if followed by a losing point the score reverts to deuce.

Example: at $40 - 30$, if the first player wins the next point he wins the game. However, if the second player wins the next three points the game is his.

Sets. A player wins a set if he wins at least four games (in the current set) and he is two games ahead of his opponent but, as you may be starting to suspect, there is yet another exception. In case the scoreline for the number of games won reaches six-all ($6 - 6$), a tie-break is played instead to decide the set.

Example: at $5 - 4$, if the first player wins the next game he takes the set $6 - 4$. But if he loses, the set is still undecided and can eventually go to either $7 - 5$, $5 - 7$ or a tie-break.

Tie-break. A tie-break (and the set to which it belongs) is won when a player wins at least seven points by a margin of two points or more.

Match. The winner of a match is the first player to win 2 sets (the wins do not need to be consecutive). Hence a match may go to 2 or 3 sets depending on how the game develops.

Rafa has been carefully studying his past performances against his next opponent and he knows he wins each point with probability precisely p, irrespective of whether he is serving or receiving and regardless of all other points played. Can you help him assess his chances of winning the match?

Input

Each test case is described by a single floating point number p, $0 \leq p \leq 1$ in its own line. A value of -1 for p marks the end of the input.

Output

For each test case, print a single line with the probabilities of Rafa winning a given game, set and match, respectively. These three numbers must be separated by a space character. Your answers should be accurate to within an absolute error of 10^{-6}.

Sample Input

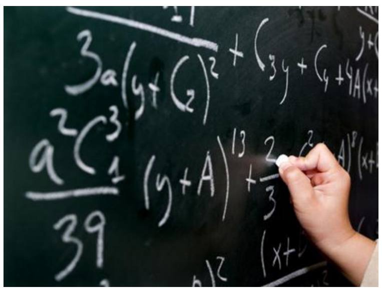
```
0.5
0.3
0.7
-1
```

Sample Output

```
0.50000000000 0.50000000000 0.50000000000
0.09921103448 0.00016770463 0.00000008437
0.90078896552 0.99983229537 0.99999991563
```

F
Guess the Numbers

John has never been very good at maths. Due to his bad grades, his parents have sent him to the Academic Coalition of Mathematics (ACM). Despite the large amount of money his parents are spending on the ACM, John does not pay much attention during classes. However, today he has begun to think about all the effort his parents are putting into his education, and he has started to feel somewhat... guilty. So he has made a decision: he is going to improve his maths grades!



However, no sooner had he resolved to pay attention than the lesson ended. So the only thing he has been able to do is to hurriedly copy the content of the blackboard in his notebook. Today, the teacher was explaining basic arithmetic expressions with unknowns. He vaguely remembers that his classmates have been substituting values into the unknowns to obtain the expressions' results. However, in all the hurry, John has only written down expressions, values and results in a messy fashion. So he does not know which value comes with each unknown, or which result goes with each expression.

That is the reason he needs your help: he wants to know, given an expression, some values and a result, whether it is possible or not to assign those values to the unknowns in order for the expression to evaluate to the given result. The particular assignment of values does not matter to John, as he wants to do it by himself. He only wants to know whether it is possible or not.

Input

Each test case in the input file consists of two lines:

- The first line contains a sequence of natural numbers. The first one ($1 \leq n \leq 5$) is the number of unknowns that will occur in the expression. It is followed by a sequence of n integers $v_1 \dots v_n$ ($0 \leq v_i \leq 50$), which are the values to be assigned to the unknowns. Finally, there is an integer m ($0 \leq m \leq 1000$) representing the desired result of the evaluation of the expression.

- The second line contains an arithmetic expression composed of lowercase letters (`a-z`), brackets (`(` and `)`) and binary operators (`+`, `-`, `*`). This expression will contain n unknowns, represented by n different lowercase letters, without repetitions. The expression will not contain any blanks and will always be syntactically correct, i.e. it is just an unknown or has the form `(`e_1 op e_2 `)`, where e_1 and e_2 are expressions and op is one of the three possible binary operators.

The input will finish with a dummy test case of just one line containing `0 0`, which must not be processed.

Output

For each test case, print a single line with `YES` if there exists an assignment of the values $v_1 \dots v_n$ to the unknowns such that the expression evaluates to m, and `NO` otherwise. Notice that each value v_i must be assigned to exactly one unknown.

Sample Input

```
3 2 3 4 14
((a+b)*c)
2 4 3 11
(a-b)
1 2 2
a
0 0
```

Sample Output

```
YES
NO
YES
```

G
Non-negative Partial Sums

You are given a sequence of n numbers $a_0, \ldots, a_{n-1}$. A cyclic shift by k positions ($0 \le k \le n-1$) results in the following sequence: $a_k, a_{k+1}, \ldots, a_{n-1}, a_0, a_1, \ldots, a_{k-1}$. How many of the n cyclic shifts satisfy the condition that the sum of the first i numbers is greater than or equal to zero for all i with $1 \le i \le n$?

Input

Each test case consists of two lines. The first contains the number n ($1 \le n \le 10^6$), the number of integers in the sequence. The second contains n integers $a_0, \ldots, a_{n-1}$ ($-1000 \le a_i \le 1000$) representing the sequence of numbers. The input will finish with a line containing 0.

Output

For each test case, print one line with the number of cyclic shifts of the given sequence which satisfy the condition stated above.

Sample Input

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

Sample Output

```
3
2
0
```

H
Peer Review

For scientific conferences, scientists submit papers presenting their ideas, and then review each other's papers to make sure only good papers are presented at the conference. Each paper must be reviewed by several scientists, and scientists must not review papers written by people they collaborate with (including themselves), or review the same paper more than once.

You have been asked to write a program to check if your favorite conference is doing things right. Whether a paper is being reviewed too much, too little, or by the wrong people - the organizers must know before it is too late!

Input

The first line in each test case has two integers, K ($1 \leq K \leq 5$) and N ($1 \leq N \leq 1000$). K is the number of reviews that each paper will receive, while N is the number of papers to be reviewed. The conference only accepts papers with a single author, and authors can only present a single paper at the conference.

Each of the next N lines describes an author and includes the name of the institution to which the author belongs, followed by the list of the K papers he or she has been requested to review. It is assumed that researchers from the same institution collaborate with each other, whereas researchers from different institutions don't. All institution names are shorter than 10 characters, and contain only upper or lowercase letters and no whitespace. Since we have as many papers as authors, papers are identified by their author's index; paper 1 was written by the first author in the list, and paper N was written by the last author.

The end of the test cases is marked with a line containing $K = 0$ and $N = 0$. You should generate no output for this line.

Output

For each test case, your program should output `NO PROBLEMS FOUND` (if all rules are being followed) or P `PROBLEMS FOUND`, where P is the number of rule violations found (counting at most 1 violation per paper). If there is exactly one rule violation overall, your program should output `1 PROBLEM FOUND`.

Sample Input

```
2 3
UCM 2 3
UAM 1 3
UPM 1 2
2 3
UCM 2 3
UAM 1 2
UPM 2 2
0 0
```

Sample Output

```
NO PROBLEMS FOUND
3 PROBLEMS FOUND
```

I
Regular Convex Polygon

A regular convex polygon is a polygon where each side has the same length, and all interior angles are equal and less than 180 degrees. A square, for example, is a regular convex polygon. You are given three points which are vertices of a regular convex polygon R; can you determine the minimum number of vertices that R must have?

Input

Each test case consists of three lines. Line i consists of two floating point values x_i and y_i ($-10^4 \leq x_1, y_1 \leq 10^4$) where (x_i, y_i) are the coordinates of a vertex of R. The coordinates are given with a precision of 10^{-6}, i.e., they differ from the exact coordinates by at most 10^{-6}. You may assume that for each test case the Euclidean distance between any two given points is at least 1, and R has at most 1000 vertices. The input will finish with a line containing the word `END`.

Output

For each test case, print one line with the minimum number of vertices that R must have.

Sample Input

```
-1385.736326 -146.954822
430.000292 -2041.361203
1162.736034 478.316025
0.000000 4147.000000
-4147.000000 0.000000
0.000000 -4147.000000
END
```

Sample Output

```
3
4
```

J
Remoteland

In the Republic of Remoteland, the people celebrate their independence day every year. However, as it was a long long time ago, nobody can remember when it was exactly. The only thing people can remember is that today, the number of days elapsed since their independence (D) is a perfect square, and moreover it is the largest possible such number one can form as a product of distinct numbers less than or equal to n.

As the years in Remoteland have $1,000,000,007$ days, their citizens just need D modulo $1,000,000,007$. Note that they are interested in the largest D, not in the largest D modulo $1,000,000,007$.

Input

Every test case is described by a single line with an integer n, ($1 \leq n \leq 10,000,000$). The input ends with a line containing 0.

Output

For each test case, output the number of days ago the Republic became independent, modulo $1,000,000,007$, one per line.

Sample Input

```
4
9348095
6297540
0
```

Sample Output

```
4
177582252
644064736
```