¿Fácil De Decir?

Un password seguro es algo delicado. Los usuarios prefieren passwords que sean fáciles de recordar (como `amigo`), pero este password puede ser inseguro. Algunos lugares usan un generador randómico de passwords (como `xvtpzyo`), pero los usuarios toman demasiado tiempo recordándolos y algunas veces lo escriben en una nota pegada en su computador. Una solución potencial es generar password "pronunciables" que sean relativamente seguros pero fáciles de recordar.

FnordCom está desarrollando un generador de passwords. Su trabajo en el departamento de control de calidad es probar el generador y asegurarse de que los passwords sean aceptables. Para ser aceptable, el password debe satisfacer estas tres reglas:

1. Debe contener al menos una vocal.
2. No debe tener tres vocales consecutivas o tres consonantes consecutivas.
3. No debe tener dos ocurrencias consecutivas de la misma letra, excepto por 'ee' o 'oo'.

(Para el propósito de este problema, las vocales son 'a', 'e', 'i', 'o', y 'u'; todas las demás letras son consonantes.) Note que Estas reglas no son perfectas; habrán muchas palabras comunes/pronunciables que no son aceptables.

●

La entrada consiste en una o más potenciales passwords, uno por línea, seguidas por una línea conteniendo una palabra 'end' que señala el fin de la entrada. Cada password tiene como mínimo una y como máximo veinte letras de largo y esta formado por solo letras en minúscula. Por cada password, despliegue si es o no aceptable, usando el formato mostrado en el ejemplo de salida.

Ejemplo de entrada

```
a
tv
ptoui
```

```
bontres
zoggax
wiinq
eep
houctuh
end
```

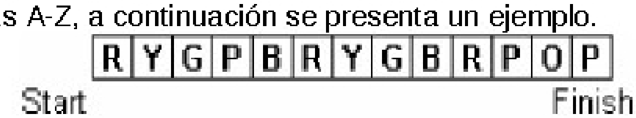
Ejemplo de salida

```
<a> is acceptable.
<tv> is not acceptable.
<ptoui> is not acceptable.
<bontres> is not acceptable.
<zoggax> is not acceptable.
<wiinq> is not acceptable.
<eep> is acceptable.
<houctuh> is acceptable.
```

Colorville

Un simple juego de niños usa un tablero que es una secuencia de cuadrados coloreados. Cada jugador tiene una pieza de juego. Los jugadores alternan turnos, sacando cartas que tienen cada una uno o dos cuadrados coloreados del mismo color. Los jugadores mueven su pieza hacia adelante en el tablero hacia el siguiente cuadrado que haga pareja con el color de la carta, o hacia adelante hasta el segundo cuadrado que haga pareja con el color de la carta que contiene dos cuadrados coloreados, o hacia adelante hasta el último cuadrado en el tablero si no hay un cuadrado con el que emparejar siguiendo la descripción anterior. Un jugador gana si su pieza está en el último cuadrado del tablero. Es posible que no exista ganador después de sacar todas las cartas.

En este problema los colores se representan las letras mayúsculas A-Z, a continuación se presenta un ejemplo.



Considere el siguiente deck de cartas: R, B, GG, Y, P, B, P, RR

Para 3 jugadores, el juego procede como sigue:

```
Jugador 1 saca R, se mueve al 1er cuadrado
Jugador 2 saca B, se mueve al 5to cuadrado
Jugador 3 saca GG, se mueve al 8vo cuadrado
Jugador 1 saca Y, se mueve al 2do cuadrado
Jugador 2 saca P, se mueve al 11vo cuadrado
Jugador 3 saca B, se mueve al 9no cuadrado
Jugador 1 saca P, se mueve al 4to cuadrado
Jugador 2 saca RR, Gano! (no hay R s al
frente de esta pieza así que va hasta el
último cuadrado).
```

Usando la misma tabla y el mismo deck de cartas, pero con 2 jugadores, el jugador 1 gana después de 7 cartas. Con 4 jugadores, no hay ganador después de utilizar todas las 8 cartas.

La entrada consiste en información de uno o más juegos. Cada juego comienza con una línea conteniendo el número de jugadores (1-4), el número de cuadrados en el tablero (1-79), y el número de cartas en el deck (1-200). Seguido por una línea de caracteres que representan los cuadrados coloreados del tablero. Seguidos por las cartas en el deck, uno el cada línea. Las Cartas pueden tener una letra o dos de las mismas letras. El final de la entrada esta señalado con una línea que tiene 0 para el número de jugadores – los otros valores son indiferentes.

Por cada juego, la salida es el jugador ganador y el número total de cartas usadas, o el número de cartas en el deck, como se muestra en el ejemplo de salida. Siempre use el plural "cards".

Ejemplo de entrada

```
2 13 8
RYGPBRYGBRPOP
R
B
GG
Y
P
B
P
RR
2 6 5
RYGRYB
R
YY
G
G
B
3 9 6
QQQQQQQQQ
Q
QQ
Q
Q
QQ
Q
0 6 0
```

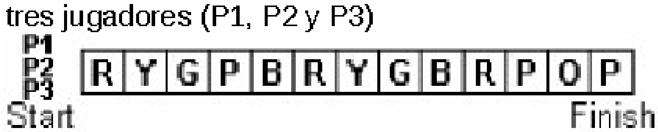
Ejemplo de salida

```
Player 1 won after 7 cards.
Player 2 won after 4 cards.
No player won after 6 cards.
```

El primer párrafo del planteamiento del problema describe claramente en que consiste el mismo.

Para quien no entendió en qué consiste este problema, bastará con ver mejor el ejemplo propuesto en el problema.

Este es nuestro tablero, y al inicio se encuentran nuestros tres jugadores (P1, P2 y P3)

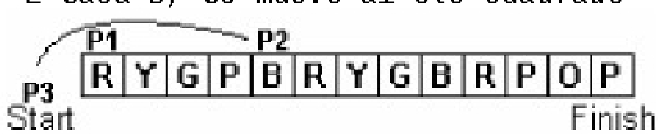


También sabemos que las cartas de nuestro deck saldrán en el siguiente orden: R, B, GG, Y, P, B, P, RR

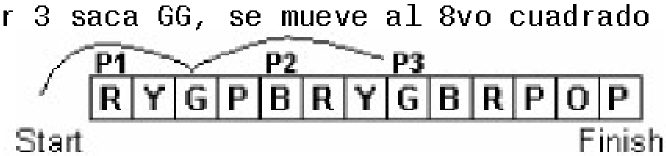
Simulemos el juego de manera gráfica:

Jugador 1 saca R, se mueve al 1er cuadrado

Jugador 2 saca B, se mueve al 5to cuadrado



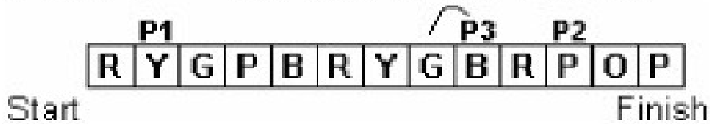
Jugador 3 saca GG, se mueve al 8vo cuadrado



Jugador 1 saca Y, se mueve al 2do cuadrado

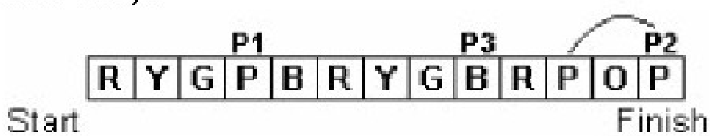
Jugador 2 saca P, se mueve al 11vo cuadrado

Jugador 3 saca B,   se mueve al 9no cuadrado



Jugador 1 saca P,   se mueve al 4to cuadrado

Jugador 2 saca RR, ¡Gano! (no mas hay R s al frente de esta pieza así que va hasta el último cuadrado).



Falling Leaves

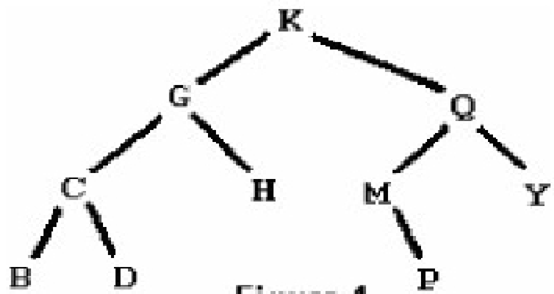


Figura 1

La Figura 1 muestra la representación gráfica de un árbol binario de letras. Lo familiarizados con los árboles binarios pueden saltarse la definición de árbol binario de letras, hojas de un árbol binario, y búsqueda en un árbol binario de letras, e ir directo al problema.

Definición.
Un **árbol binario de letras** puede ser una de dos cosas:
1. Puede estar vacía.
2. Puede tener un nodo raíz. Un nodo tiene una letra como dato y hace referencia a subárboles izquierdo y derecho. Los subárboles izquierdo y derecho son también árboles binarios de letras.

En la representación gráfica de un árbol binario de letras:
1. Un árbol vacío es omitido completamente.
2. Cada nodo esta indicado por

    Su dato letra,

    Un segmento de línea abajo a la izquierda hacia su subárbol izquierdo, si el subárbol izquierdo no está vacío,

    Un segmento de línea abajo a la derecha hacia su subárbol derecho, si el subárbol derecho no esta vacío.

Una **hoja** en un árbol binario es un nodo donde ambos subárboles están vacíos. En el ejemplo en la Figura 1, tiene cinco nodos con datos B, D, H, P, y Y.

El recorrido preorder de un árbol de letras satisface las propiedades:

1. Si el árbol esta vacío, entonces el recorrido preorder está vacío.
2. Si el árbol no esta vacío, entonces el recorrido preorder consiste en lo siguiente, en orden:
   El dato del nodo raíz,
   El recorrido preorder del subárbol izquierdo del nodo raíz,
   El recorrido preorder del subárbol derecho del nodo raíz.

El recorrido preorder del árbol de la Figura 1 es KGCBDHQMPY.

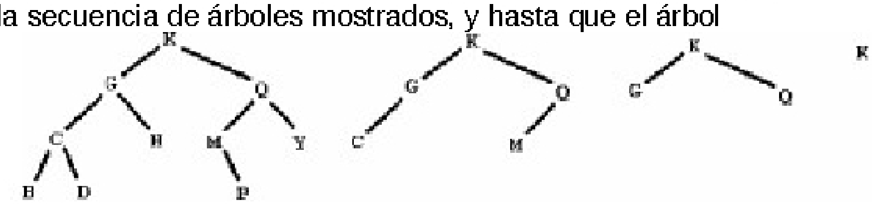
Un árbol como el de la Figura 1 es también un árbol binario de búsqueda de letras. Un árbol binario de búsqueda de letras es un árbol de letras en el cual cada nodo satisface:

1. Los datos raíz vienen después en el alfabeto que todos los datos en los nodos en el subárbol izquierdo.
2. Los datos raíz vienen antes en el alfabeto que todos los datos en los nodos en el subárbol derecho.

El problema:
Considere la siguiente secuencia de operaciones en un árbol binario de búsqueda de letras:

Borrar las hojas y listar los datos removidos
Repetir este proceso hasta que el árbol este vacío.

Empezando por el árbol de abajo a la izquierda, producimos la secuencia de árboles mostrados, y hasta que el árbol



este vacío removiendo las hojas de datos

BDHPY
CM
GQ
K

Tu problema es empezar con tales secuencias de líneas de hojas de un árbol binario de búsqueda de letras y desplegar el recorrido preorder del árbol.
La entrada contiene uno o más sets de datos. Cada set de datos es una secuencia de uno o más líneas con letras mayúsculas. Las líneas contienen las hojas removidas del árbol binario de búsqueda de la forma descrita anteriormente. Las letras en una línea están listados en orden alfabético. Los sets de datos están separados por una línea que contiene un asterisco ('*'). El último set de datos está seguido por un signo de dólar ('$'). No hay espacios en blanco ni líneas vacías en la entrada.

Por cada set de datos de entrada, hay un único árbol binario de búsqueda que puede ser producido con la secuencia de hojas. La salida es una línea que contiene solo el recorrido preorder del árbol, sin blancos.

Ejemplo de entrada

```
BDHPY
CM
GQ
K
*
AC
B
$
```

Ejemplo de salida

```
KGCBDHQMPY
BAC
```

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, $0 \le p \le 1$ 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

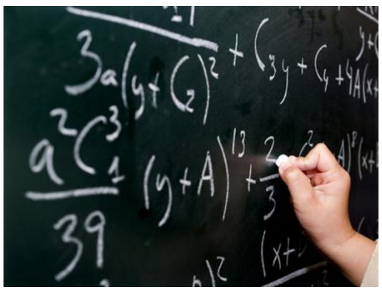
```
0.5
0.3
0.7
-1
```

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 ($0 \leq k \leq n-1$) 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 $1 \leq i \leq n$?

Input

Each test case consists of two lines. The first contains the number n ($1 \leq n \leq 10^6$), the number of integers in the sequence. The second contains n integers $a_0, \ldots, a_{n-1}$ ($-1000 \leq a_i \leq 1000$) 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
```