COCI 2016/2017

Round #2, November 5th, 2016

Tasks

Task	Time limit	Memory limit	Score
Go	1 s	32 MB	50
Tavan	1 s	32 MB	80
Nizin	1 s	64 MB	100
Prosječni	1 s	64 MB	120
Zamjene	6 s	512 MB	140
Burza	1 s	512 MB	160
Total			650

Mirko quickly got tired of Jetpack Joyride and started playing Pokémon GO! on his phone. One of the curiosities of this game is the so-called *evolution* of Pokémon.

In order to evolve Pokémon of species P_i, Mirko must provide K_i candy intended for a Pokémon of that species. After the evolution of that Pokémon, he gets 2 candies back. Pokémon can evolve **only** with the help of candy intended for their species.

Mirko has N species of Pokémon and M_i candy for Pokémon of species P_i and wants to know how many total Pokémon he can evolve.

He also wants to know which Pokémon can evolve the most number of times. If there are multiple such Pokémon, output the one with the smallest Pokédex number. In other words, the one that appears earliest in the input data.

INPUT

The first line of input contains the integer N ($1 \le N \le 70$), the number of Pokémon species.
The following $2N$ lines contains N sets of data, wherein it holds:

- line $2i$ contains string P_i, 20 characters long at most, the name of the i^{th} Pokémon species;
- line $2i$+1 contains integers K_i ($12 \le K_i \le 400$) and M_i ($1 \le M_i \le 10^4$), the number of candy necessary for the evolution of one Pokémon of the i^{th} species and the total number of candy Mirko has for Pokémon of the i^{th} species

OUTPUT

The first line of output must contain the total number of Pokémon that Mirko can evolve.
The second line of output must contain the name of the Pokémon that can evolve the most number of times.

SCORING

In test cases worth 16 points total, it will hold $N = 3$.
The first line of output will count towards 50% of points for that test case.
The second line of output will count towards 50% of points for that test case.

SAMPLE TESTS

input

```
4
Caterpie
12 33
Weedle
12 42
Pidgey
12 47
Rattata
25 71
```

output

```
14
Weedle
```

input

```
7
Bulbasaur
25 74
Ivysaur
100 83
Charmander
25 116
Charmeleon
100 32
Squirtle
25 1
Wartortle
100 173
Pikachu
50 154
```

output

```
11
Charmander
```

Clarification of the first test case:
Let's describe how Mirko evolved Weedles. For Weedles' first evolution, Mirko spent 12 candy, but got back 2, so he has 32 candy left (42-12+2). After the second evolution, he is left with 22 candy. After the third evolution, he had 12 candy, which was enough for just one more evolution. This way, Mirko evolved 4 Weedles.

Similarly, we see that Mirko can evolve at most 3 Caterpies, 4 Pidgeys and 3 Rattatas.

Out of all Pokémon, Weedle and Pidgey evolved the most number of times, but Weedle's Pokédex number is smaller (it appears earlier in the input data), so it is the solution of the second part of the task.

Little Željko has been reading his grandma's old letters in the attic and has come across a word of length *N*. Unfortunately, he couldn't make out what it said because of the spilled ink. He rewrote the word on a piece of paper by replacing each of the *M* illegible letters with the character '#'. He handed the piece of paper to his grandma and she gave him *K* different candidates for each of the illegible letters. After that, Željko wrote all the possible words in a notebook and decided to closely look at their properties in order to determine what the original words was. After seeing the words written down in the notebook, his grandma realized that the word they're looking for is the X^{th} one in alphabetical order. Željko had the sniffles the day they learned the alphabet in school, so he's asking you for your help with determining the original word.

INPUT

The first line of input contains the integers *N*, *M*, *K* and *X* ($1 \leq N \leq 500$, $1 \leq M \leq N$, $1 \leq K \leq 26$, $1 \leq X \leq 10^9$).
The second line of input contains the word of length *N* that Željko wrote on a piece of paper, consisting of lowercase letters of the English alphabet and the character '#'.
Each of the following *M* lines contains one word of length *K*, the i^{th} of these words consisting of letters that could replace the i^{th} illegible letter.

Number *X* will always be less than or equal to the total number of words that can be constructed.

OUTPUT

The first line of output must contain the X^{th} word in alphabetical order.

SCORING

In test cases worth 30% of total points, it will hold $M = 1$ and $K = 3$.
In test cases worth an additional 30% of points, it will hold $M = 1$.

SAMPLE TESTS

input

```
9 2 3 7
po#olje#i
sol
znu
```

output

```
posoljeni
```

input

```
4 1 2 2
#rak
zm
```

output

```
zrak
```

Clarification of the first test case:

The possible words, in aphabetical order, are: “pololjeni”, “pololjeui”, “pololjezi”, “poooljeni”, “poooljeui”, “poooljezi”, “posoljeni”, “posoljeui”, “posoljezi”.

Do Geese See God? Or, Was It A Rat I Saw? Nevermind the geese or rats, this is just an unnecessary introduction to showcase Mislav's love of palindromes. Help him solve the following task!

Let *A* be an array of *N* integers. We say that *A* is palindromic if for each *i* it holds *A[i]* = *A[N-i+1]*, where *A[i]* represents the *i*th element of array *A*, and the index of the first element in the array is 1.

Mislav can modify the array in the following way: in a single move, he chooses two **adjacent** elements of that array and replaces them with their sum. Notice that the number of elements in the array is going to decrease by 1 after each move. Mislav wants to know what is the least number of moves he must make in order for the original array to become palindromic.

INPUT

The first line of input contains the integer *N* ($1 \leq N \leq 10^6$) that represents the number of elements in the array.
The following line contains *N* space-separated positive integers that represent the elements in Mislav's array. The numbers in the input will be at most 10^9.

OUTPUT

Output the minimal number of moves it takes to transform the original array to a palindromic one, given the rules from the task.

SCORING

In test cases worth 30% of total points, it will hold $N \leq 10$.
In test cases worth 60% of total points, it will hold $N \leq 1\ 000$.

SAMPLE TESTS

input	input	input
3 1 2 3	5 1 2 4 6 1	4 1 4 3 2
output	**output**	**output**
1	1	2

Clarification of sample test cases:

1. **1 2** 3 -> 3 3
2. 1 **2 4** 6 1 -> 1 6 6 1
3. **1 4** 3 2 -> 5 **3 2** -> 5 5

Slavko is bored, so he's filling out an N × N table with positive integers.

He's particularly happy if he manages to fill out the table so that the following conditions are met:

- The average of the numbers in each row is an integer that is located in the same row.
- The average of the numbers in each column is an integer that is located in the same column.
- All numbers in the table are different.

Help Slavko find a table that will make him happy.

INPUT

The first line of input contains the integer N (1 ≤ N ≤ 100).

OUTPUT

Output N lines, in each line output N integers separated by space.
Let the j^{th} number in the i^{th} line correspond to the value that Slavko will write down in the i^{th} row and j^{th} column of the table.

All numbers must be greater than 0 and smaller than 1 000 000 000.
If there are multiple solutions, output any.
If there is no solution, output -1.

SAMPLE TESTS

input

```
3
```

output

```
1 2 3
4 5 6
7 8 9
```

input

```
2
```

output

```
-1
```

Clarification of the first test case:
The averages of individual rows are, respectively: 2, 5, 8.
The averages of individual columns are, respectively: 4, 5, 6.
Since the average of each row is located in the correspoding row and the average of each column is located in the corresponding column, the output table will make Slavko happy.

Dominik has envisioned an array of positive integers $p_1, \ldots, p_n$. Let's denote the sorted version of that array as $q_1, \ldots, q_n$.

Also, he envisioned a set of allowed substitutions. If a pair (X, Y) is a member of the set of allowed substitutions, Dominik can swap the numbers at positions X and Y in array p.

Marin is giving Dominik Q queries, and each of them is of one of the following types:

1. Swap numbers at positions A and B.
2. Add pair (A, B) to the set of allowed substitutions.
   Marin can give pair (A, B) that is already in the set of allowed substitutions.
3. See if it's possible to sort the array using only the allowed substitutions?
   The substitutions can be used in an arbitrary order, and each substitution can be made an arbitrary number of times.
4. Let's call a pair of positions (A, B) **linked** if the number from position A is possible to transfer to position B using only allowed subtitutions.
   Let's call the set of all positions linked to position A **the cloud** of A.
   A cloud is **good** if it's possible for each position j from the cloud to achieve $p_j = q_j$ using a series of allowed substitutions.

You must answer how many pairs of different positions (A, B) exist such that it holds:

a. Positions A and B are not linked
b. Cloud of A is not good and cloud of B is not good
c. If we add pair (A, B) to the set of allowed substitutions, the cloud of A (created by linking the cloud of A and cloud of B) becomes good.

Please note: Pairs (A, B) and (B, A) are considered an identical pair.

INPUT

The first line of input contains integers N and Q ($1 \le N, Q \le 1\,000\,000$).
The second line of input contains N integers $p_1, \ldots, p_n$ ($1 \le p_1, \ldots, p_n \le 1\,000\,000$).
Each of the following Q lines contains a query in the following format:

- The first number in the line is the type of query T from the set {1, 2, 3, 4}.
- If the type of query T is 1 or 2, two different integers A and B follow ($1 \le A, B \le N$) that represent the substitution (A, B).

OUTPUT

For each query of type 3 or 4, output the answer in its separate line.
The answer to query of type 3 is "DA" (Croatian for yes) or "NE" (Croatian for no), without the quotation marks, and the answer to query of type 4 is a non-negative integer from the task.

SCORING

In test cases worth 50% of total points, it will hold $N, Q \le 1\,000$.

PRIMJERI TEST PODATAKA

input	input	input
3 5 1 3 2 4 3 2 2 3 4 3	5 5 4 2 1 4 4 3 4 1 1 3 3 4	4 10 2 1 4 3 3 4 1 1 2 3 4 2 2 3 2 1 2 4 2 3 4 3
output	**output**	**output**
1 NE 0 DA	NE 1 DA 0	NE 2 NE 1 3 DA

Clarification of the first test case:

The answer to the first query is 1 because the pair of positions (2, 3) meets all given requirements.
The answer to the second query is NE (no) because it is impossible to transfer numbers 2 and 3 to the corresponding positions, because the set of allowed substitutions is empty.
After the third query, we add pair (2, 3) to the set of allowed substitutions.
The answer to the fourth query is now 0 because 2 and 3 are already linked, and the answer to the fifth query is DA (yes), because it is possible to sort the array by applying the allowed substitution (2, 3).

Daniel is tired of looking for a job, so he decided to visit his friend Stjepan. Surprisingly, when he entered Stjepan's home, he came across a tree with *N* nodes denoted with numbers from 1 to *N*. The node **number 1** contains a coin.

Stjepan told him: "*Put this blindfold on and we'll play!*" Daniel gave him a strange look, but decided to do it. Stjepan then told him the rules of the game:

- Daniel picks a node and marks it
- Stjepan moves the coin to an **unmarked** node adjacent to the one where the coin is currently in
- Stjepan marks the node which he moved the coin from

These three steps repeat until Stjepan can't make a move anymore. Given the fact that he is blindfolded, Daniel doesn't exactly know what node contains the coin in any given moment of the game. However, he does know the outline of the tree and where the coin was at the beginning of the game.

Daniel just realized that he hasn't applied to a single job for the past two hours, and wants to quickly finish playing the game. Now he wants to know if he can play in a way that, **no matter what moves Stjepan makes**, the game ends in **at most k moves**. In other words, so that **Stjepan moves the coin less than k times**.

Help Daniel and determine whether he can finish the game on time and go back to sending his resume to companies he's never heard of.

INPUT

The first line of input contains two integers, *N* and *K*. ($1 \le K \le N \le 400$)
Each of the following *N - 1* lines contains two integers *A* and *B* ($1 \le A, B \le N$) that denote that an undirected link exists between nodes labeled with *A* and *B*.
It is guaranteed that the given graph will be a tree.

OUTPUT

The first and only line of output must contain the word "DA" (Croatian for yes), without quotation marks, if Daniel can ensure that the game ends in at most *K* moves, and "NE" (Croatian for no) if he can't.

SAMPLE TESTS

input	input	input
6 2 1 2 2 3 3 4 1 5 5 6	3 1 1 2 1 3	8 2 1 2 2 3 2 4 5 6 6 8 1 5 7 1
output	**output**	**output**
DA	NE	DA

Clarification of the second test case:
Daniel can mark any node. If he marks node 1 or 2, Stjepan can move the coin to node 3, and if he marks node 3, Stjepan can move the coin to node 2.

Clarification of the third test case:
In his first move, Daniel can mark node 2, and in the second move mark node 6. Wherever Stjepan moves the coin in his first move, he won't be able to make a second move.