Sia's Box

Sia has a mysterious box and she wants to play a game with you. The box supports two types of operations:

- `1 x`: Put the number x into the box.
- `2`: Take out a number from the box.

Sia will give you a sequence of operations and the results of the above `2` operations. Your task is to determine what is really hidden in the box: a stack, a queue, a max priority queue or something else.

Input

The first line of the input contains a single integer N (1 <= N <= 1000), indicating the number of operations. Each of the next N lines contains a single operation described above. For operation `2`, there is an additional number x indicating the result of that operation. The value of x in both operations satisfies 1 <= x <= 100.

Output

You should print one of the following answers on a line by itself:

- `stack` if it is certain that the box is a stack.
- `queue` if it is certain that the box is a queue.
- `priority queue` if it is certain that the box is a max priority queue.
- `impossible` if it is certain that the box cannot be any of those three data structures.
- `not sure` if the box could possibly be more than one of those three data structures.

Example 1

```
Input:
6
1 1
1 2
1 3
2 1
2 2
2 3

Output:
queue
```

Example 2

```
Input:
6
1 1
1 2
1 3
2 3
2 2
2 1

Output:
not sure
```

Example 3

```
Input:
2
1 1
2 2

Output:
impossible
```

Example 4

```
Input:
7
1 84
1 36
1 61
1 4
2 4
1 61
2 61

Output:
stack
```

Example 5

```
Input:
7
1 2
1 5
1 1
1 3
2 5
1 4
2 4

Output:
priority queue
```

Labor Strike

There are `n` labor unions in New York City. You have learned that the labor union `i` will undertake a strike every `s_i` days.

For example, assume the 1st day is Sunday, and labor union 1 undertakes a strike ever 4 days. Then there will be a strike on the 4th day (the Wednesday), the 8th day (following Sunday) and so forth. (See the table below.)

Because the MTA employees support all the unions, the subways do not work if at least one of the unions is on strike. The only exception to this are Fridays and Saturdays: even if a union has a planned strike for that day, the subways will still operate as usual.

You need to figure out the number of days that you can not take the subway in the following `m` days.

You can assume the 1st day is always Sunday.

Days	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
Union 1				x				x				x		
Union 2			x			x			x			x		
Union 3								x						
No subway			x	x				x	x			x		

Input

The 1st line contains an integer `m` (`7 <= m <= 3650`) equal to the number of days for which you wish the make the calculation.

The next line contains another integer `n` (`1 <= n <= 100`) representing the number of labor unions.

The i-th of the next `n` lines contain a positive integer `s_i` (which will never be a multiple of 7) giving the striking period of labor union i.

Output

Output one integer followed by a newline: the total number of days that the subway won't operate.

Example 1 (shown in the table above)

```
Input:
14
3
4
3
8
```

```
Output:
5
```

You can find there will be strikes on the 3rd, 4th, 6th, 8th, 9th, and 12th days. So the subway will not operate on the 3rd, 4th, 8th, 9th, and 12th days (since the 6th day is a Friday).

Example 2

```
Input:
100
4
12
15
25
40
```

```
Output:
15
```

Scorewords

Alice developed a new word game called *Scorewords*. The game comes with a special dictionary that matches words to their point values. In the game, each player comes up with a sentence. The sentences are scored based on the sum of the scores for each word. If a word does not appear in the dictionary, its score is zero.

Alice contracted you to write a program that given all the sentences provided by different players, calculates their scores.

Input

The first line of input contains 2 positive integers, `m <= 1000`, the number of words in the dictionary, and `n <= 100`, the number of sentences.

`m` lines follow; each containing a word (a string of up to 16 lower-case letters) and its point value (`0 <= value <= 1,000,000`).

Following the dictionary are the `n` sentences. Each sentence consists of one or more lines of text such that each space separated word contains only lowercase letters. A period on a line by itself marks the end of the sentence.

Output For each sentence, output its score computed as the sum of values for all words that appear in the sentence. Words that do not appear in the dictionary have scores equal to zero. The output ends with a newline.

Example

```
Input:

7 2
administer 100000
spending 200000
manage 50000
responsibility 25000
expertise 100
skill 50
money 75000
the incumbent will administer the spending of kindergarden milk money
and exercise responsibility for making change he or she will share
responsibility for the task of managing the money with the assistant
whose skill and expertise shall ensure the successful spending exercise
.
this individual must have the skill to perform a heart transplant and
expertise in rocket science
.

Output:

700150
150
```

Grade Sort

You are given the midterm exam scores for all students in all of NYU campuses, 0 <= score <= 100.

To report the midterm grades, you need to display all the scores in ascending order.

Input

The first line contains one integer N. N is the total number of grades, 1 <= N <= 10,000,000.

In the next line, there are N integers indicating the grades.

Output

Print a line with N space separated integers. These integers are the grades of the midterm sorted in ascending order. The output ends with a newline.

Warning: This problem may have potentially very large inputs/outputs (see the value for N above). Use fast IO operations.

Example 1

```
Input:
4
50 60 100 90

Output:
50 60 90 100
```

Example 2

```
Input:
6
99 92 93 92 93 91

Output:
91 92 92 93 93 99
```

Transitive Relation

In mathematics, a relation R over a set X is *transitive* if for all elements a, b, c in X, whenever R relates a to b and b to c, then R also relates a to c. Transitivity is a key property of both partial orders and equivalence relations.

For example, whenever $x = y$ and $y = z$, then we know that $x = z$, thus equality is a transitive relation over real numbers.

One may naturally split the set S into equivalence classes with transitive relations.

Formally, given a set S and an transitive relation over S, the equivalence class of an element $a \in S$ is the set $EC(a) = x \in S|x\ a$.

Given a finite set S and a transitive relation R over S. Find the largest equivalence class.

Input

The first line contains tho numbers N and M. N is the number of elements in S and M is the number of pairs of elements, which are known to be in the relation R. You can assume that $1 <= N <= 30,000$ and $0 <= M <= 500,000$.

Elements in S are indexed with integers from 1 to N. Each of the following M lines contain a pair of integers A and B $(1 <= A, B <= N, A \neq B)$ which states that (A, B) belongs to the transitive relation R. There could be repetitions among the given pairs.

Output

Output 1 number followed by a newline. The size of the largest equivalence class.

Example 1

```
Input:
3 2
1 3
2 3

Output:
3
```

Example 2

```
Input:
5 7
1 2
2 1
4 5
3 5
5 3
3 4
1 2

Output:
3
```

Example 3

```
Input:
10 3
1 2
4 5
6 7

Output:
2
```