Problem K
A Complex Problem
Time limit: 3 seconds

There are many problems in the field of computer science, and some are harder than others. Computer scientists have accordingly categorized problems using complexity classes, and like to analyze these classes to see how they interact with each other.

A complexity class is a simply a set of problems; for example, the complexity class P is the set of decision problems which are solvable with an algorithm whose runtime scales as a polynomial function of the input size. We know some results about the relations between complexity classes: for example, every problem in the complexity class P also belongs to the complexity class NP of decision problems for which "yes"-instances can be verified in polynomial time. However, we don't know if every problem in NP is also in P (and we won't ask you to figure this out for today). Therefore, P and NP could be either one or two distinct complexity classes. On the other hand, we know that the Halting Problem is in ALL (the set of all decision problems) but not in R (the set of decision problems solvable by a Turing machine). Therefore, ALL and R are two distinct complexity classes.

Given a series of relations between complexity classes, can you find the minimum and maximum number of distinct complexity classes? Two complexity classes are distinct if and only if there is some problem that exists in one class but not in the other.

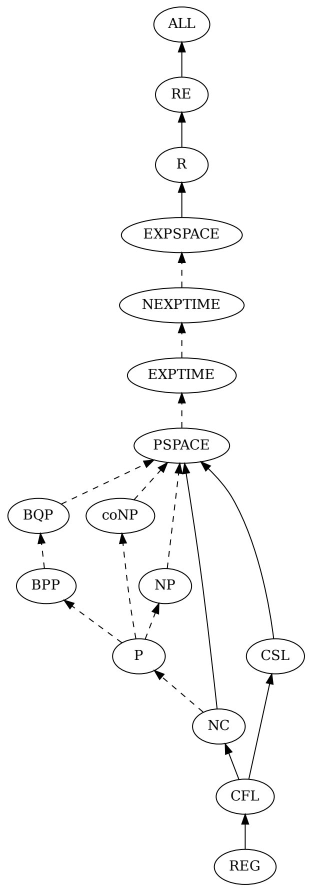

Illustration of Sample 4, where dashed lines indicate subset relations ($\subseteq$) and solid lines indicate proper subset relations ($\subsetneq$)

Input

Input begins with two space-separated integers $0 \le M, N \le 10^5$ such that $M + N > 0$. Each of the next M lines consists of two distinct space-separated complexity classes A_i and B_i, where a complexity class is a set of problems, denoted by one to eight uppercase or lowercase English letters. Each of these M lines indicates that $A_i \subseteq B_i$, meaning that every problem in A_i is also in B_i. Similarly, each of the next N lines consists of two distinct space-separated complexity classes A_j and B_j. Each of these N lines indicates that $A_j \subsetneq B_j$, meaning that every problem in A_j is also in B_j **and** that at least one problem in B_j is not in A_j.

There is at most one relation between any two distinct complexity classes specified in the input, and the relations between complexity classes will not imply any logical contradiction.

Output

Output two space-separated integers on a single line: the minimum and maximum number of distinct complexity classes among the specified complexity classes given in the relations in the input.

Sample Input 1

```
1 0
P NP
```

Sample Output 1

```
1 2
```

Sample Input 2

```
0 1
R ALL
```

Sample Output 2

```
2 2
```

Sample Input 3

```
3 0
QMIP MIPstar
MIPstar RE
RE QMIP
```

Sample Output 3

```
1 1
```

Sample Input 4

```
11 8
NC P
P BPP
P coNP
P NP
BPP BQP
coNP PSPACE
BQP PSPACE
NP PSPACE
PSPACE EXPTIME
EXPTIME NEXPTIME
NEXPTIME EXPSPACE
REG CFL
CFL NC
CFL CSL
NC PSPACE
CSL PSPACE
EXPSPACE R
R RE
RE ALL
```

Sample Output 4

```
7 16
```