

CCC 2013 Stage 2

Day 1, Problem 1: All Your Base Belong to Palindromes

Problem Description

Most of the time, humans have 10 fingers. This fact is the main reason that our numbering system is base-10: the number 257 really means $2 \times 10^2 + 5 \times 10^1 + 7 \times 10^0$. Notice that each digit in base-10 is in the range from $0 \dots 9$.

Of course, there are other bases we can use: binary (base-2), octal (base-8) and hexadecimal (base-16) are common bases that really cool people use when trying to impress others. In base- b , the digits are in the range from $0 \dots b - 1$, with each digit (when read from right to left) being the multiplier of the next larger power of b .

So, for example 9 (in base-10) is:

- 9 in base-16
- 11 in base-8 ($1 \times 8^1 + 1 \times 8^0 = 9$)
- 1001 in base-2 ($1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9$)

Noticing the above, you can see that 9 is a palindrome in these 3 different bases. A *palindrome* is a sequence which is the same even if it is written in reverse order: English words such as “dad”, “mom” and “racecar” are palindromes, and numbers like 9, 11, 1001 are also palindromes.

Given a particular number X (in base-10), for what bases b ($2 \leq b < X$) is the representation of X in base- b a palindrome?

Input Specification

There will be one line, containing the integer X ($2 \leq X \leq 1\,000\,000\,000$).

For test cases worth 80% of the points, you may assume $X \leq 10\,000$.

Output Specification

The output should consist of a sequence of increasing integers, each on its own line, indicating which bases have the property that X written in that base is a palindrome. Note that we will only concern ourselves with bases which are less than X , and that the first possible valid base is 2.

Sample Input

9

Output for Sample Input

2

8

Explanation of Output for Sample Input

The number 9 was shown to be a palindrome in base-2 and base-8 in the problem description. The other bases do not lead to palindromes: for example, in base-3, 9 is expressed as 100, and in base-5, 9 is expressed as 14.

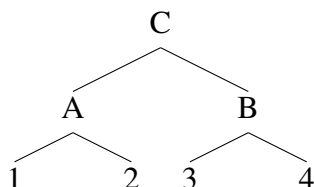
CCC 2013 Stage 2

Day 1, Problem 2: Tourney

Problem Description

Don Cherry has been hired to run 24-hour coverage of a series of single-elimination, bracket-style, furniture disassembly tourneys (tournaments). Each competitor has a furniture disassembly skill level, an integer between 1 and 1 000 000 000. In every head-to-head match, the competitor with the larger skill level wins and moves on, while the other is eliminated from the tourney. It is guaranteed that, at any time, the skill levels of all competitors are distinct, so there are no ties.

There are 2^N ($1 \leq N \leq 20$) competitor positions in the tourney tree, numbered $1, 2, \dots, 2^N$ from left to right. In the first round, competitors 1 and 2 face off in a furniture disassembly race, as do competitors 3 and 4, etc. In each subsequent round, the winners of the first two matches from the previous round compete, as do the winners of the following two, etc. After N rounds, a single winner remains. For example, when $N = 2$, the tourney tree looks like this:



where A represents the winner of the match between competitors 1 and 2, B represents the winner of the match between competitors 3 and 4, and C represents the winner of the match between A and B. The winner of this tourney is C.

Because of sponsorship contracts, some competitors will be replaced over time. After any new person comes in, a new tourney is held.

In order to help Don Cherry, you must write a program to compute certain tourney statistics at various points in time, given a sequence of M ($1 \leq M \leq 1\,000\,000$) commands — see the input format below.

Input Specification

The first line of input contains two integers, N ($1 \leq N \leq 20$) and M ($1 \leq M \leq 1\,000\,000$), separated by one space.

The next 2^N lines, for i from 1 to 2^N , each contain one integer S_i , indicating the skill level of the initial competitor at position i in the tourney tree.

Each of the following M lines will be a command in one of three formats:

- “R i S ” means that the competitor at position i is removed, and replaced with a new one with skill level S . A new tourney is then held.
- “W” means that your program should determine who won the current tourney. Print out the position i (between 1 and 2^N) of this competitor.
- “S i ” means that your program should print out the number of rounds that the competitor at position i won in the current tourney.

Output Specification

For each “W” or “S i ” command in the input, print out the corresponding integer on a line by itself.

Sample Input

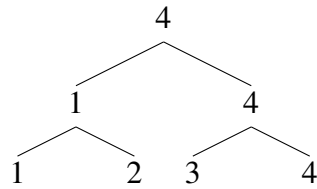
```
2 8
30
20
10
40
S 1
W
R 4 9
S 4
W
R 2 35
S 2
W
```

Output for Sample Input

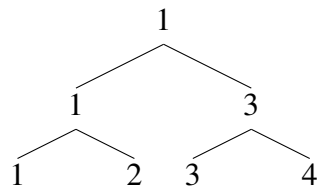
```
1
4
0
1
2
2
```

Explanation of Output for Sample Input

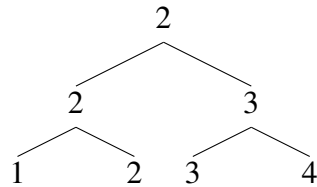
The results of the initial tourney are as follows:



As can be seen, competitor 1 wins 1 match, and competitor 4 wins the entire tourney. After this, competitor 4 is replaced by a new competitor with skill level 9. As can be seen below, this causes a different outcome for the tourney held after the third command:



Finally, the state of the tourney tree after the next competitor replacement (caused by the sixth command) is:



CCC 2013 Stage 2

Day 1, Problem 3: LHC

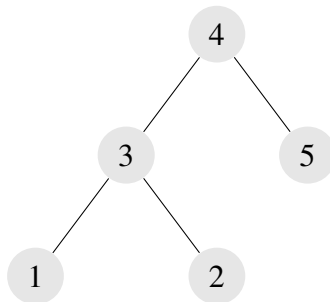
Problem Description

For your latest top-secret experiment, you will need a large quantity of Higgs bosons. To obtain these elusive particles, you will need to build a *large hadron collider*, a long circular tunnel that you can use to accelerate particles and smash them into each other.

You already have access to an extensive network of tunnels, which is guaranteed to be connected and free of cyclic paths. In other words, the existing tunnels form a *tree* structure. This system can be represented by N *junctions*, labelled 1 through N , connected by $N - 1$ *tunnels*, each of which connects two junctions. Tunnels can be traversed in either direction (i.e., if there is a junction from a to b , that junction also goes from b to a).

By adding exactly one tunnel to the network, you can create a cyclic path, which you will use to build your large hadron collider. You wish to form the longest possible collider in this way, where we define length as the number of tunnels in a cycle. Also, you would like to compute the number of ways to do this— that is, the number of distinct pairs of junctions such that adding a tunnel between them forms a cycle of maximum length.

For example, in the following network, we can form a collider of length 4 by building a tunnel between junctions 1 and 5, or between 2 and 5:



Input Specification

The first line of each test case will contain N ($3 \leq N \leq 400\,000$), the number of junctions. The next $N - 1$ lines will each contain two space-separated integers i and j , indicating that there is a tunnel between junctions i and j ($1 \leq i, j \leq N$).

Output Specification

The output should consist of a single line with two space-separated integers: the length of the longest possible collider, and the number of ways to achieve that length. Note that you may need a 64-bit integer to store the answer (`long long` in C/C++, `LongInt` in Pascal).

For test cases worth 40% of the points, you may assume $N \leq 5\,000$.

Sample Input

```
5
1 3
2 3
3 4
4 5
```

Output for Sample Input

```
4 2
```