Codeforces Round #190 (Div. 1)

A. Ciel and Robot

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel has a robot on a 2D plane. Initially it is located in (0, 0). Fox Ciel code a command to it. The command was represented by string s. Each character of s is one move operation. There are four move operations at all:

- 'U': go up, (x, y) $\rightarrow$ (x, y+1);
- 'D': go down, (x, y) $\rightarrow$ (x, y-1);
- 'L': go left, (x, y) $\rightarrow$ (x-1, y);
- 'R': go right, (x, y) $\rightarrow$ (x+1, y).

The robot will do the operations in s from left to right, and repeat it infinite times. Help Fox Ciel to determine if after some steps the robot will located in (a, b).

Input

The first line contains two integers a and b, ($-10^9 \le a, b \le 10^9$). The second line contains a string s ($1 \le |s| \le 100$, s only contains characters 'U', 'D', 'L', 'R') — the command.

Output

Print "Yes" if the robot will be located at (a, b), and "No" otherwise.

Sample test(s)

input

```
2 2
RU
```

output

```
Yes
```

input

```
1 2
RU
```

output

```
No
```

input

```
-1 1000000000
LRRLU
```

output

```
Yes
```

input

```
0 0
D
```

output

```
Yes
```

Note

In the first and second test case, command string is "RU", so the robot will go right, then go up, then right, and then up and so on.

The locations of its moves are (0, 0) $\rightarrow$ (1, 0) $\rightarrow$ (1, 1) $\rightarrow$ (2, 1) $\rightarrow$ (2, 2) $\rightarrow$ ...

So it can reach (2, 2) but not (1, 2).

B. Ciel and Duel

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel is playing a card game with her friend Jiro.

Jiro has n cards, each one has two attributes: $position$ (Attack or Defense) and $strength$. Fox Ciel has m cards, each one has these two attributes too. It's known that position of all Ciel's cards is Attack.

Now is Ciel's battle phase, Ciel can do the following operation many times:

1. Choose one of her cards X. This card mustn't be chosen before.
2. If Jiro has no alive cards at that moment, he gets the damage equal to (X's strength). Otherwise, Ciel needs to choose one Jiro's alive card Y, then:

   - If Y's position is Attack, then (X's strength) $\geq$ (Y's strength) must hold. After this attack, card Y dies, and Jiro gets the damage equal to (X's strength) - (Y's strength).
   - If Y's position is Defense, then (X's strength) $>$ (Y's strength) must hold. After this attack, card Y dies, but Jiro gets no damage.

Ciel can end her battle phase at any moment (so, she can use not all her cards). Help the Fox to calculate the maximal sum of damage Jiro can get.

Input

The first line contains two integers n and m $(1 \leq n, m \leq 100)$ — the number of cards Jiro and Ciel have.

Each of the next n lines contains a string $position$ and an integer $strength$ $(0 \leq strength \leq 8000)$ — the position and strength of Jiro's current card. Position is the string "`ATK`" for attack, and the string "`DEF`" for defense.

Each of the next m lines contains an integer $strength$ $(0 \leq strength \leq 8000)$ — the strength of Ciel's current card.

Output

Output an integer: the maximal damage Jiro can get.

Sample test(s)

input

```
2 3
ATK 2000
DEF 1700
2500
2500
2500
```

output

```
3000
```

input

```
3 4
ATK 10
ATK 100
ATK 1000
1
11
101
1001
```

output

```
992
```

input

```
2 4
DEF 0
ATK 0
0
0
1
1
```

output

```
1
```

Note

In the first test case, Ciel has 3 cards with same $strength$. The best strategy is as follows. First she uses one of these 3 cards to attack "ATK 2000" card first, this attack destroys that card and Jiro gets $2500 - 2000 = 500$ damage. Then she uses the second card to destroy the "DEF 1700" card. Jiro doesn't get damage that time. Now Jiro has no cards so she can use the third card to attack and Jiro gets 2500 damage. So the answer is $500 + 2500 = 3000$.

In the second test case, she should use the "1001" card to attack the "ATK 100" card, then use the "101" card to attack the "ATK 10" card. Now Ciel still has cards but she can choose to end her battle phase. The total damage equals $(1001 - 100) + (101 - 10) = 992$.

In the third test case note that she can destroy the "ATK 0" card by a card with strength equal to 0, but she can't destroy a "DEF 0" card with that card.

C. Ciel the Commander

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Now Fox Ciel becomes a commander of Tree Land. Tree Land, like its name said, has n cities connected by $n - 1$ undirected roads, and for any two cities there always exists a path between them.

Fox Ciel needs to assign an officer to each city. Each officer has a rank — a letter from '`A`' to '`Z`'. So there will be 26 different ranks, and '`A`' is the topmost, so '`Z`' is the bottommost.

There are enough officers of each rank. But there is a special rule must obey: if x and y are two distinct cities and their officers have the same rank, then on the simple path between x and y there must be a city z that has an officer with higher rank. The rule guarantee that a communications between same rank officers will be monitored by higher rank officer.

Help Ciel to make a valid plan, and if it's impossible, output "`Impossible!`".

Input

The first line contains an integer n $(2 \le n \le 10^5)$ — the number of cities in Tree Land.

Each of the following $n - 1$ lines contains two integers a and b $(1 \le a, b \le n, a \ne b)$ — they mean that there will be an undirected road between a and b. Consider all the cities are numbered from 1 to n.

It guaranteed that the given graph will be a tree.

Output

If there is a valid plane, output n space-separated characters in a line — i-th character is the rank of officer in the city with number i.

Otherwise output "`Impossible!`".

Sample test(s)

input

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

output

```
A B B B
```

input

```
10
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
```

output

```
D C B A D C B D C D
```

Note

In the first example, for any two officers of rank '`B`', an officer with rank '`A`' will be on the path between them. So it is a valid solution.

D. Ciel and Flipboard

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel has a board with n rows and n columns, there is one integer in each cell.

It's known that n is an odd number, so let's introduce $x = \frac{n+1}{2}$. Fox Ciel can do the following operation many times: she choose a sub-board with size x rows and x columns, then all numbers in it will be multiplied by -1.

Return the maximal sum of numbers in the board that she can get by these operations.

Input

The first line contains an integer n, ($1 \le n \le 33$, and n is an odd integer) — the size of the board.

Each of the next n lines contains n integers — the numbers in the board. Each number doesn't exceed 1000 by its absolute value.

Output

Output a single integer: the maximal sum of numbers in the board that can be accomplished.

Sample test(s)

input

```
3
-1 -1 1
-1 1 -1
1 -1 -1
```

output

```
9
```

input

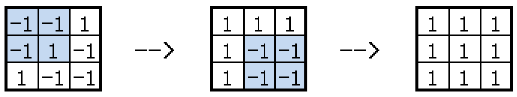
```
5
-2 0 0 0 -2
0 -2 0 -2 0
0 0 -2 0 0
0 -2 0 -2 0
-2 0 0 0 -2
```

output

```
18
```

Note

In the first test, we can apply this operation twice: first on the top left 2×2 sub-board, then on the bottom right 2×2 sub-board. Then all numbers will become positive.



E. Ciel and Gondolas

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Fox Ciel is in the Amusement Park. And now she is in a queue in front of the Ferris wheel. There are n people (or foxes more precisely) in the queue: we use first people to refer one at the head of the queue, and n-th people to refer the last one in the queue.

There will be k gondolas, and the way we allocate gondolas looks like this:

- When the first gondolas come, the q_1 people in head of the queue go into the gondolas.
- Then when the second gondolas come, the q_2 people in head of the remain queue go into the gondolas.
  ...
- The remain q_k people go into the last (k-th) gondolas.

Note that q_1, q_2, ..., q_k must be positive. You can get from the statement that $\sum_{i=1}^{k} q_i = n$ and $q_i > 0$.

You know, people don't want to stay with strangers in the gondolas, so your task is to find an optimal allocation way (that is find an optimal sequence q) to make people happy. For every pair of people i and j, there exists a value u_{ij} denotes a level of unfamiliar. You can assume $u_{ij} = u_{ji}$ for all i, j $(1 \le i, j \le n)$ and $u_{ii} = 0$ for all i $(1 \le i \le n)$. Then an unfamiliar value of a gondolas is the sum of the levels of unfamiliar between any pair of people that is into the gondolas.

A total unfamiliar value is the sum of unfamiliar values for all gondolas. Help Fox Ciel to find the minimal possible total unfamiliar value for some optimal allocation.

Input

The first line contains two integers n and k $(1 \le n \le 4000$ and $1 \le k \le min(n, 800))$ — the number of people in the queue and the number of gondolas. Each of the following n lines contains n integers — matrix u, $(0 \le u_{ij} \le 9$, $u_{ij} = u_{ji}$ and $u_{ii} = 0)$.

Please, use fast input methods (for example, please use BufferedReader instead of Scanner for Java).

Output

Print an integer — the minimal possible total unfamiliar value.

Sample test(s)

input

```
5 2
0 0 1 1 1
0 0 1 1 1
1 1 0 0 0
1 1 0 0 0
1 1 0 0 0
```

output

```
0
```

input

```
8 3
0 1 1 1 1 1 1 1
1 0 1 1 1 1 1 1
1 1 0 1 1 1 1 1
1 1 1 0 1 1 1 1
1 1 1 1 0 1 1 1
1 1 1 1 1 0 1 1
1 1 1 1 1 1 0 1
1 1 1 1 1 1 1 0
```

output

```
7
```

input

```
3 2
0 2 0
2 0 3
0 3 0
```

output

```
2
```

Note

In the first example, we can allocate people like this: {1, 2} goes into a gondolas, {3, 4, 5} goes into another gondolas.

In the second example, an optimal solution is : {1, 2, 3} | {4, 5, 6} | {7, 8}.