2018 Canadian Computing Olympiad
Day 1, Problem 1
Geese vs. Hawks

Time Limit: 1 second

Problem Description

Troy and JP are big hockey fans. Every hockey team played N games this season. Each game was between two teams and the team that scored more points won. No game ended in a tie.

Troy's favourite team is the Waterloo Geese and he recorded the outcome of all their games as a string S. $S_i = \texttt{W}$ if the Geese won their i-th game; otherwise $S_i = \texttt{L}$ if the Geese lost their i-th game. He also recorded that they scored A_i points in their i-th game.

JP's favourite team is the Laurier Hawks and he recorded the outcome of all their games as a string T. $T_j = \texttt{W}$ if the Hawks won their j-th game; otherwise $T_j = \texttt{L}$ if the Hawks lost their j-th game. He also recorded that they scored B_j points in their j-th game.

Troy and JP recorded wins/losses and points in the order that their favourite teams played.

A *rivalry* game is one where the Geese and Hawks played each other. Since neither Troy or JP recorded the opponents their favourite teams faced, they are not sure which games, if any, were rivalry games. They wonder what is the maximum possible sum of points scored by both their teams in rivalry games that matches the information they recorded.

Input Specification

The first line contains one integer N $(1 \leq N \leq 1\,000)$.

The second line contains string S of length N consisting of characters `W` and `L`.

The third line contains N integers $A_1, \cdots, A_N$ $(1 \leq A_i \leq 1\,000\,000)$.

The fourth line contains string T of length N consisting of characters `W` and `L`.

The fifth line contains N integers $B_1, \cdots, B_N$ $(1 \leq B_j \leq 1\,000\,000)$.

For 10 of the 25 available marks, $N \leq 10$.

Output Specification

Print one line with one integer, the maximum possible sum of points scored in potential rivalry games.

Sample Input 1

```
1
W
2
W
3
```

Output for Sample Input 1

```
0
```

Explanation for Output for Sample Input 1

Since both the Geese and Hawks won all their games, there could not have been any rivalry games.

Sample Input 2

```
4
WLLW
1 2 3 4
LWWL
6 5 3 2
```

Output for Sample Input 2

```
14
```

Explanation of Output for Sample Input 2

The fourth game each team played could have been a rivalry game where Geese won with 4 points to the Hawk's 2 points. The third game the Geese played and the second game the Hawks played could have been a rivalry game where the Hawks won with 5 points compared to 3 points of the Geese. The points scored by both teams is $4 + 2 + 5 + 3 = 14$ and this is the maximum possible.

Note that the first game played by the Geese was a win where they scored 1 goal: this game cannot be against the Hawks, since there is no game where the Hawks scored 0 goals. Similarly, the first game played by the Hawks cannot be used, since the Hawks lost and scored 6 goals, and the Geese never had a game where they scored at least 7 goals.

2018 Canadian Computing Olympiad
Day 1, Problem 2
Wrong Answer

Time Limit: 1 second

Problem Description

Troy made the following problem (titled WA) for a programming contest:

> There is a game with N levels numbered from 1 to N. There are two characters, both are initially at level 1. For $i < j$, it costs $A_{i,j}$ coins to move a character from level i to level j. It is not allowed to move a character from level i to level j if $i > j$. To win the game, every level (except level 1) must be visited by exactly one character. What is the minimum number of coins needed to win?

JP is a contestant and submitted the following Python solution.

```
def Solve(N, A):
  # A[i][j] is cost of moving from level i to level j
  # N is the number of levels
  x, y, sx, sy = 1, 1, 0, 0 # Initialize x and y to 1, sx and sy to 0
  for i in range(2, N + 1): # loop from 2 to N
    if sx + A[x][i] < sy + A[y][i]:
      sx += A[x][i]
      x = i
    else:
      sy += A[y][i]
      y = i
  return sx + sy
```

Troy is certain that JP's solution is wrong. Suppose for an input to WA, JP's solution returns X but the minimum number of coins needed is Y. To show how wrong JP's solution is, help Troy find an input N and $A_{i,j}$ such that $\frac{X}{Y}$ is maximized.

Input Specification

There is no input.

Output Specification

Print an input to WA in the following format:

On the first line, print one integer N $(2 \leq N \leq 100)$.
Then print $N-1$ lines; the i-th line should contain $N-i$ integers $A_{i,i+1}, \cdots, A_{i,N}$ $(1 \leq A_{i,j} \leq 100)$.

If your output is not the correct format, it will get an *incorrect* verdict on the sample test in the grader and score 0 points.

Otherwise, suppose that for your input, JP's solution returns X but the minimum number of coins needed is Y. Then you will receive $\left\lceil \min(25, \frac{X}{4Y}) \right\rceil$ points where $\lceil Z \rceil$ is the smallest integer that is not greater than Z.

Sample Output

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

Explanation for Sample Output

The optimal way to win the game is for one character to visit level 2 and the other character to visit levels 3, 4 and 5. This costs $(1) + (2 + 7 + 5) = 15$ coins. JP's solution returns 18. Thus $\frac{X}{4Y} = \frac{18}{4 \times 15} = 0.3$, so this output will receive $\lceil 0.3 \rceil = 1$ point.

2018 Canadian Computing Olympiad
Day 1, Problem 3
Fun Palace

Time Limit: 1 second

Problem Description

You are working hard to prepare a fun party for your fun friends. Fortunately, you have just located the perfect venue for the fun party: a *fun palace*. The fun palace has N *fun rooms* connected in a linear structure. The fun rooms are numbered from 1 to N, and for $1 \leq i \leq N-1$, fun rooms i and $i+1$ are connected by a *fun tunnel*. We say that such a fun tunnel is *incident* to fun rooms i and $i+1$. In addition, fun room 1 is incident to an *exit tunnel* leaving the fun palace.

Fun tunnels can be in one of two states: open or closed. When the fun tunnel between fun rooms i and $i+1$ is opened, fun friends may travel freely between the two rooms, in either direction.

By default, the fun tunnels will all be closed. However, they may temporarily be opened by a group of fun friends pressing down a required number of *fun buttons*. For each fun tunnel, there is a set of fun buttons present in the fun rooms connected to the fun tunnel. If all of the buttons in one of the rooms connected to a tunnel are pressed down by distinct fun friends, then the fun tunnel will open. Otherwise, the fun tunnel will immediately close. The fun tunnel between rooms i and $i+1$ is connected to a set of a_i buttons in room i and a set of b_i buttons in room $i+1$. To put this another way, if there are at least a_i friends in room i or if there are b_i friends in room $i+1$, then tunnel between room i and $i+1$ may be opened.

The exit tunnel operates under similar rules, but it is only connected to a single set of e buttons present in room 1.

You want to ensure your friends have maximum fun, and that obviously means keeping your fun friends trapped in the fun palace forever. What is the maximum number of fun friends that you can distribute to particular fun rooms such that the exit fun tunnel is never opened?

Input Specification

The first line will contain a single integer N ($1 \leq N \leq 1000$), the number of fun rooms. The next line contains a single integer e ($1 \leq e \leq 10\,000$). The next $N-1$ lines contain two space-separated integers each, with the ith of these lines containing a_i and b_i ($1 \leq a_i, b_i \leq 10\,000$).

For 3 of the 25 marks available, $1 \leq e \leq 200$, $a_i = 1$, $b_i = 1$.

For an additional 5 of the 25 marks available, $1 \leq e, a_i, b_i \leq 2$.

For an additional 12 of the 25 marks available, $N \leq 200$, $1 \leq e, a_i, b_i \leq 200$.

Output Specification

Output a single integer, the maximum number of fun friends over all possible distributions of fun friends to fun rooms such that there is no way for the fun friends to open the exit tunnel.

Sample Input 1

```
2
20
5 5
```

Output for Sample Input 1

```
19
```

Explanation for Output for Sample Input 1

If we had any more than 19 fun friends, then they would be able to move to fun room 1 and press all 20 fun buttons required in order to open the exit tunnel.

Sample Input 2

```
2
20
5 20
```

Output for Sample Input 2

```
24
```

Explanation for Output for Sample Input 2

Suppose we place 24 fun friends in fun room 2. In order to open the fun tunnel between fun rooms 1 and 2, 20 fun friends must stay in fun room 2 to press the fun buttons. This allows only 4 fun friends into fun room 1, which is not enough to press every fun button in the set of 5.

Sample Input 3

```
7
7
2 8
6 6
1 1
2 4
2 8
7 8
```

Output for Sample Input 3

```
23
```

Explanation for Output for Sample Input 3

One optimum distribution is to place 9 fun friends in fun room 2 and 14 fun friends in fun room 7.