USC CSCI 270 Programming Assignment
Fall 2018

Background

You are Brian! You have spent the entire night painting "ROMANES EUNT DOMUS" all over Jerusalem. There is just one spot left. Unfortunately, it's far away, and your grammar is wrong, so if any of the guards catch you, they will correct you on the proper use of the vocative plural declension and imperative conjugation. This will exhaust poor Brian, who has been up all night.

Fortunately, Brian has the support of the Judean People's Front (or was it the People's Front of Judea?), who can encourage Brian and remove some of his exhaustion by inviting Brian to their conference on "Other than sanitization, medicine, education, and order, what have the Romans ever done for us?".

Description

Brian is at the top left of an $N \times N$ grid, and needs to get to the bottom right in order to paint his message. To reach his destination as soon as possible, he'll only move along the shortest way, i.e. **moving rightward and downward**. Along the way, Brian can encounter Roman guards marked as negative integers, where the magnitude is the amount of damage they deal. The positive integers are tiles that heal Brian that amount of HP, `P` are tiles that prevent Brian's next damage, and `D` are tiles that double Brian's next healing effect. If Brian reaches 0 HP or less, he collapses and the mission fails, so prevent that!

No matter how many `P` tiles and `D` tiles Brian encounters, their effects don't stack. For example, if Brian walks on a `D` tile when he's already in the state of "doubling next healing effect", nothing happens. When Brian next encounters a healing tile, the healing will be doubled. Afterwards, Brian returns to a "normal" state. The same applies for `P` tiles.

What is the minimum amount of life Brian needs in order to complete his task?

Input Format

Read from standard input. The input is guaranteed to be well-formed, as described below:

The first line contains an integer N. Next N lines describe the grid, each containing N strings separated by space. For each string:

- If the string is `P`, it indicates tiles that prevent Brian's next damage.
- If the string is `D`, it indicates tiles that double Brian's next healing effect.
- Otherwise, the string should actually be a non-zero integer x.
  1. $x > 0$: the tile will heal Brian x HP.
  2. $x < 0$: guard on the tile will deal $|x|$ damage to Brian.

Output Format

Output one integer to standard output: the minimum life Brian needs to complete his task.

Sample

Input 1	Input 2	Input 3	Input 4

`3`
`2 -1 -13`
`-2 -30 20`
`15 -13 -1`
	`4`
	`3 P D 10`
	`-5 -6 -2 -100`
	`1 -4 -5 -24`
	`-3 4 6 8`
		`5`
		`5 -6 7 -8 9`
		`-6 7 -8 9 -10`
		`7 -8 9 -10 11`
		`-8 9 -10 11 -12`
		`9 -10 11 -12 13`
			`5`
			`-1 -1 -1 -1 -1`
			`D 1 2 3 4`
			`-1 -2 -3 1 2`
			`5 10 -1 1 2`
			`10 5 50 -1 -300`

Output 1	Output 2	Output 3	Output 4
`1`	`2`	`5`	`210`

Constraints

For all test cases, $2 \leq N \leq 100$, no guard deals more than 1000 damage and no tile heals more than 1000 HP.

The table below shows types of tiles each test case includes.

Test case No.	Guards	Healing tiles	`P` tiles	`D` tiles
1-5	No	Yes	No	No
6-25	Yes	No	No	No
26-40	Yes	Yes	No	No
41-50	Yes	Yes	Yes	Yes

Solving easy test cases first may help you work out other test cases; If you cannot solve all of them, try to solve as many as possible.

Submission Format

You can use C, C++, Python, or Java in this assignment, so long as you submit **a single file** containing the source code of your solution. `grid.cpp` contains the skeleton code for C++ solution. However, if you decide to use C, Python, or Java, please confirm that your program meets all the requirements and notify the grader the way to execute your program in comments of your code.

Grading

The full score of this assignment is 100, 2 points for each test case you pass.

You pass a test case if and only if your output is correct and your program terminates within 1 second with return value 0.