



Problema 3. Zombies

Fișier header:	<code>zombies.h</code>
Limită de timp:	2 secunde
Limită de memorie:	1024 megabytes

Tocmai ți-ai descărcat ultima versiune a binecunoscutului joc Plante și Zombi. Jocul se joacă pe un teren infinit 2D în care vor apărea mai mulți zombi. Fiecare zombie are o poziție de început (s_x, s_y) în care apare, o viteză de v unități/secundă și un șir de exact 15 mutări pe care le va face. O mutare este codificată printr-un caracter din mulțimea $\{U, R, D, L\}$ (“up”, “right”, “down”, “left”).

Zombii vor executa mutările secvențial, pornind de la prima mutare. Fiecare zombie se va mișca în direcția indicată de mutarea curentă timp de exact o secundă, după care va trece la mutarea următoare. De exemplu, dacă un zombie are viteză 3, începe de pe poziția $(2, 4)$ și prima mutare este U , acesta va executa mutările $(2, 4) \rightarrow (2, 5) \rightarrow (2, 6) \rightarrow (2, 7)$ într-o secundă, după care va trece la următoarea mutare. Pentru mai multe informații, vezi explicațiile.

Dacă un zombie a epuizat șirul inițial de mutări, va continua cu șirul de mutări de la început, astfel că ei vor cicla prin setul de mutări la infinit. În mișcarea lor, doi zombi pot ocupa aceeași poziție în același timp, fără a interfera unul cu altul.

Scopul nostru este de a neutraliza zombii, plasând **plante-laser** (*eng. Laser Beans*). O plantă va fi plasată la poziția (l_x, l_y) și va avea setată una din cele patru direcții $\{U, R, D, L\}$. Plantele vor neutraliza tot ce se va afla în direcția lor. De exemplu, o plantă la poziția $(2, 5)$ direcționată spre U va neutraliza tot ce se va afla pe pozițiile $(2, 5), (2, 6), (2, 7), (2, 8), (2, 9), \dots$. Se observă că un zombie va fi neutralizat chiar și în momentul în care acesta va ajunge pe poziția plantei.

Care este numărul minim de plante-laser care trebuie plasate pentru ca fiecare zombie să fie neutralizat la un moment dat pe parcursul jocului?

Detalii de implementare

Veți implementa funcția cu următorul antet:

```
std::vector<Plant> minimum_plants(std::vector<Zombie> zombies)
```

Structurile de date folosite în cadrul interacțiunii sunt descrise în fișierul `zombies.h` și au următoarea descriere:

```
struct Zombie { int sx, sy, v; string moves; };  
struct Plant { int lx, ly; char dir; };
```

Funcția `minimum_plants` va fi apelată o singură dată, parametrii având specificațiile din enunț. **Se garantează că șirul de mutări ale fiecărui zombie are lungime exact 15.**

Pentru ca soluția returnată să fie validă, trebuie ca numărul de plante-laser returnate să fie minimul posibil și ca plantele să neutralizeze toți zombii. De asemenea, plantele trebuie plasate în poziții distincte, iar coordonatele l_x și l_y trebuie să fie numere întregi între -10^9 și 10^9 inclusiv.



Punctare

Subtask	Punctaj	Constrângeri
1	17 puncte	Numărul de zombi este între 1 și 100 Pentru fiecare zombie: <ul style="list-style-type: none"> • $-300 \leq s_x, s_y \leq 300$ • $1 \leq v \leq 300$
2	34 de puncte	Numărul de zombi este între 1 și 2 000 Pentru fiecare zombie: <ul style="list-style-type: none"> • $-10^5 \leq s_x, s_y \leq 10^5$ • $1 \leq v \leq 10^5$
3	21 de puncte	Numărul de zombi este între 1 și 20 000 Pentru fiecare zombie: <ul style="list-style-type: none"> • $-10^6 \leq s_x, s_y \leq 10^6$ • $1 \leq v \leq 10^6$
4	28 de puncte	Numărul de zombi este între 1 și 50 000 Pentru fiecare zombie: <ul style="list-style-type: none"> • $-10^6 \leq s_x, s_y \leq 10^6$ • $1 \leq v \leq 10^6$

Model de grader

Graderul va citi de la consolă datele de intrare în următorul format:

- linia 1: N , reprezentând numărul de zombii
- linia $1 + i$ ($1 \leq i \leq N$): $s_x s_y v moves$ (separate prin câte un spațiu)

Graderul va afișa la consolă răspunsul vostru în următorul format:

- linia 1: K , reprezentând numărul de lasere
- linia $1 + i$ ($1 \leq i \leq K$): $l_x l_y dir$ (separate prin câte un spațiu)

Exemple

intrare	ieșire
3 0 0 3 URURURURURURURU 10 5 1 LLLLLDDDRRRRRRRR 15 0 2 UUUUUUUUUUUUUUUU	1 -2 1 R
2 10 10 1 UUUUUUUUUUUUUUUU -1 -1 1 DDDDDDDDDDDDDDD	2 -1 11 R -1 12 D

Explicație

Aceasta este figura pentru primul exemplu:

