

Twoim zadaniem jest napisanie programu, który będzie grał w grę. Następnie odbędzie się turniej programów (detale w sekcji ocenianie). Oto zasady gry:

- Gra odbywa się na grid'ie 2D o rozmiarze  $n \times m$ .
- W grę grają dwaj gracze (dwa programy): gracz niebieski i gracz czerwony.
- Każde pole gry zawiera: ścianę lub gracza lub pocisk skierowany w jedną w czterech stron (lub pociski).
- Pola na obrzeżach planszy są zawsze ścianami.
- Gracze zaczynają w dwóch rogach planszy.
- Gracze w każdej turze mogą wykonać jedną z następujących akcji:
  - 0 – Idź w górę.
  - 1 – Idź w dół.
  - 2 – Idź w lewo.
  - 3 – Idź w prawo.
  - 4 – Strzel w górę.
  - 5 – Strzel w dół.
  - 6 – Strzel w lewo.
  - 7 – Strzel w prawo.
  - 8 – Nie rób nic.
- Każda tura wygląda następująco:
  - Gracze wykonują swoje akcje (próba wejścia na ścianę jest zastępowana nie robieniem niczego).
  - Pociski ruszają się o jedno pole. Jeśli pocisk trafia na ścianę, to zawraca.
  - Jeśli gracze weszli na siebie to są cofani na poprzednie pole.
  - Jeśli jakiś gracz znajduje się na tym samym polu co jakiś pocisk to przegrywa tą rundę (jeśli obaj gracze znajdują się na takim polu to gra kończy się remisem).
- Jeśli gra osiągnie limit rund, to kończy się remisem.
- Plansza generowana jest losowo, ale tak, aby być symetryczna dla obu graczy.

Twój program będzie grał w grę poprzez liczenie dokładnie jednego ruchu. Na wejściu znajdzie się pełny stan gry, a na wyjściu powinien znaleźć się numer akcji, którą chcesz wykonać. Program będzie uruchamiany tyle razy ile rund będzie miała gra.

## Parametry gry w rozgrywkach

- Wysokość planszy – między 15 a 50.
- Szerokość planszy – między 20 a 50.
- Przybliżona liczba ścian – między 10 a 200.
- Liczba rund, po której następuje remis – między 100 a 2000.

Dokładne parametry zostaną ustalone w trakcie trwania OFF-a, i będą zależeć między innymi od liczby uczestników, średnich czasów działania programów.

## Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite  $n$  i  $m$  – wysokość i szerokość planszy. W następnych  $n$  wierszach znajduje się opis planszy, każdy składający się z  $m$  opisów pola (detale poniżej).

W kolejnym wierszu znajduje się jedna liczba całkowita – numer rundy rozgrywki. W ostatnim wierszu znajduje się jeden znak R lub B, który oznacza gracza, którym rusza się twój program.

Każdy opis pola składa się z 4 znaków, określających to co na danym polu się znajduje. Spacja oznacza brak obiektu, znak # oznacza ścianę, znak R oznacza gracza czerwonego, a znak B oznacza gracza niebieskiego. Znak ^ oznacza pocisk lecący w górę, znak v oznacza pocisk lecący w dół, znak < oznacza pocisk lecący w lewo, a znak > oznacza pocisk lecący w prawo.

Zauważ że na polu nigdy nie mogą znaleźć się więcej niż cztery istotne obiekty (spośród gracza, ściana, pociski) oraz, że nie ma znaczenia ile pocisków lecących w tą samą stronę znajduje się na danym polu (ma znaczenie jedynie czy jest ich 0 czy więcej niż 0).

## Wyjście

Na wyjściu powinna znaleźć się jedna liczba całkowita – numer ruchu. Jeśli twój program wypisze inne wyjście, to będzie to potraktowane jako poddanie partii.

## Przykład

Wejście dla testu gra0:

```
6 5
# # # # #
# # # # #
# R # # #
# # # # #
# < B #
# # # # #
2
R
```

Wyjście dla testu gra0:

```
2
```

**Uwaga:** Każde pole planszy jest opisane przez cztery znaki (patrz sekcja: Wejście).

## Limity

Twój program będzie posiadał około 150ms na wykonanie jednego ruchu (oraz rozsądny limit pamięci, który zostanie ustalony wkrótce, możesz założyć, że będzie to przynajmniej 64MiB). Osiągnięcie timeout'u zostanie potraktowane jako ruch 8 – nie robienie niczego.

**Uwaga:** parametry te mogą ulec zmianie, ale nie powinny być gorsze niż podane powyżej.

## Ocenianie

Ocenianie odbędzie się w formie turnieju, w którym grać będzie każdy z każdym (możliwe, że wiele razy, ale nie więcej niż 5). Za wygraną utrzymuje się 1 punkt, za remis 0.5 punktu, za przegraną 0 punktów.

Dopuszczamy możliwość, w której liczba zgłoszeń nie pozwoli na przeprowadzenie wszystkich meczy w tej formie, z powodu zbyt dużej liczby meczy. W takim wypadku zostanie zastosowany nieco inny system:

- Odbędzie się faza grupowa, w której grać będzie każdy z każdym,
- po której odbędzie się faza pucharowa, w której będzie grać ileś najlepszych programów z każdej grupy.

Liczba punktów na SIO2 za pierwsze miejsce w rankingu wynosi 100 punktów (pozostałe miejsca otrzymują punkty SIO2 proporcjonalnie do liczbie punktów z turnieju).

## Uruchamianie lokalne

W sekcji pliki znajduje się plik `gra.py`, który pozwala na uruchomienie gry lokalnie podając ścieżki do plików wykonywalnych implementujących obu graczy (to może być ten sam program). W celu poznania detali uruchom `python3 gra.py -help`.

## Zgłaszanie błędów

W celu zgłaszania wszelkich błędów związanych z treścią lub programem `gra.py`, proszę pisać na SIO2 lub łąpać kadrę bezpośrednio.