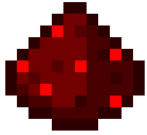


Czerwony proszek



Zapewne każdy z nas słyszał o sandboxowej grze, jaką jest Minecraft (coś wspaniałego 😊). Jako że zawsze bardzo interesowała mnie informatyka, najbardziej lubiłem tworzyć różnego rodzaju mechanizmy, bramki logiczne itp.

Czerwony proszek — tzw. redstone był jednym z kluczowych dla mnie elementów rozgrywki, który służył do zasilania maszyn, bramek logicznych, ... Zmodyfikujmy trochę zasady gry i przenieśmy tę analogię — napiszmy program, który sprawdzi, czy jesteśmy w stanie przeprowadzić energię przez obszar o wymiarach 10 na 10. Zaczynamy w punkcie (0,0), wprowadzając tu energię na redstone, która musi dotrzeć do punktu (9,9), również na redstone. Wyjaśnijmy

zasady. Energia może poruszać się tylko w poziomie i pionie, po czerwonym proszku (oznaczamy 'R'). Energia może przemieszczać się w zasięgu maksymalnie pięciu jednostek, długością tzw. sznura. Jeśli po pięciu jednostkach nie napotka wzmacniacza, tzw. repeatera (oznaczamy 'P'), to nie będzie w stanie pójść dalej. Jeśli zaś napotka wzmacniacz po pięciu jednostkach lub wcześniej, będzie w stanie przemieścić się kolejne pięć jednostek, w każdą następną stronę. Energia nie jest w stanie przenosić się bezpośrednio ze wzmacniacza na wzmacniacz. Drogi nie łączą się oraz nie zapętłają, również nie do chodzi do sytuacji, w której czerwony proszek przebiega bezpośrednio obok drugiego, łącząc się z nim, tworząc tzw. sieci(poła). Całą resztę oznaczamy dowolnymi innymi znakami np. 'O'.

Wejście

Na wejście programu zostanie podana pewna nieokreślona ilość 100-znakowych łańcuchów znaków, reprezentujących zawartość obszaru. Każdy łańcuch oddzielony jest znakiem nowej linii.

Wyjście

Na wyjściu programu ma pojawić się ciąg binarny, odpowiadający wejściu. Jeśli jesteśmy w stanie przeprowadzić energię z punktu (0,0) do punktu (9,9), wypisujemy 1, jeśli nie, to 0.

Przykład

Wejście:

```
RR0000000000RR0000000000RPO0000000000RR0000000000RR0000000000RPO0000000000RR0000000000RR0000000000RPO0000000000R  
R0000000000R0000000000R0000000000R0000000000R0000000000P0000000000R0000000000R0000000000R0000000000RRPRRRRRRR
```

Wyjście:

```
1  
0
```