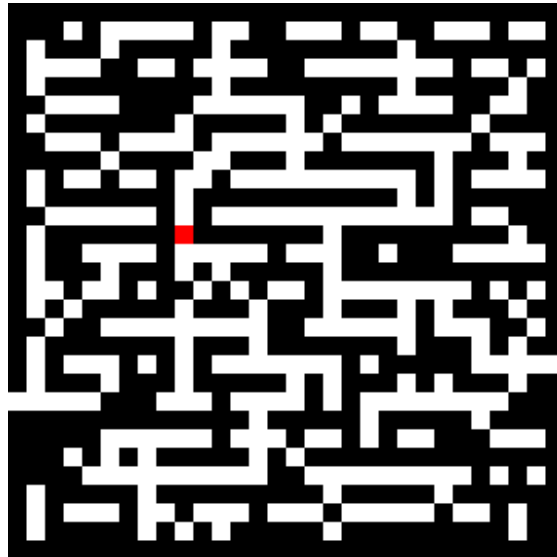


Zadanie: Pomoc Robotowi w wyjściu z labiryntu

Cel zadania:

Napisz program w języku Python, który umożliwi robotowi znalezienie wyjścia z labiryntu.



Treść zadania:

Zadaniem jest stworzenie funkcji, która dla danego labiryntu, załączonego jako plik tekstowy (.txt), określi, czy istnieje możliwość wyjścia dla Robota. Labirynt jest reprezentowany przez listę ciągów znaków, gdzie ' ' (spacja) oznacza drogę, '#' to ściana, a 'r' to pozycja startowa Robota. Funkcja powinna zwracać True, jeśli istnieje ścieżka do wyjścia, lub False, jeśli taka ścieżka nie istnieje. Labirynty mogą mieć różne kształty i nie muszą być kwadratowe ani prostokątne. Robot może poruszać się w cztery główne strony: w lewo, w górę, w prawo i w dół. Jeśli w labiryncie znajduje się więcej niż jeden robot, funkcja powinna zwrócić False. Program powinien również umożliwiać wczytanie labiryntu z załączonego pliku tekstowego przy użyciu okna dialogowego.

Kryteria oceny:

1. **Poprawność algorytmu:** Czy funkcja prawidłowo zwraca True lub False w zależności od tego, czy robot może znaleźć wyjście z labiryntu dla załączonych w zadaniu labiryntów oraz czy potrafi je pobrać z pliku tekstowego.
2. **Czytelność i organizacja kodu:** Czy kod jest łatwy do zrozumienia, dobrze skomentowany i zorganizowany.
3. **Dodatkowe testy:** Sprawdzenie, czy algorytm przechodzi dodatkowe testy, które nie zostały załączone w zadaniu.

Przykład wejściowy:

```
['# #',  
 '# r#',  
 '####']
```

Przykład wyjściowy: **True**

Przykład wejściowy:

```
['####',  
 '# r#',  
 '####']
```

Przykład wyjściowy: **False**

Poniżej kilka bardziej skomplikowanych labiryntów, wraz z poprawnymi odpowiedziami.

```
labirynt1 = ["###",  
             "#r#",  
             "###"]
```

Funkcja powinna zwrócić: **False**

```
labirynt2 = ["r"]
```

Funkcja powinna zwrócić: **True**

```
labirynt3 = ["###",  
             "#r ",  
             "###"]
```

Funkcja powinna zwrócić: **True**

```
labirynt4 = ["r ",  
             " r"]
```

Funkcja powinna zwrócić: **False**

```

Labirynt5 = ["#####",
             "# #r####",
             "# # #   ",
             "# # # ##",
             "# # # ##",
             "#       #",
             "#####"]

```

Funkcja powinna zwrócić: **True**

```

Labirynt6 = ["#####",
             "#r#  ## #",
             "# # #  #",
             "# # # ##",
             "# # #  #",
             "#       ##",
             "#####"]

```

Funkcja powinna zwrócić: **False**

```

Labirynt7 = [' # #  # ',
             '      ### ',
             '## #  ## ',
             '      ##### ',
             ' # ##### ',
             '## ##r # # ',
             '      # # # ',
             ' # # #  ## ',
             ' ###  ##### ',
             '      # # ' ]

```

Funkcja powinna zwrócić: **False**

Funkcja powinna zwrócić: **False**

Funkcja powinna zwrócić: **False**

```

Labirynt9 = ["#####",
"### # # # # #",
"# ### #####",
"# # # # # # #",
"# ##### # # #",
"## ### # # #",
"# ##### # #####",
"## # # # # #",
"##### # # # # #",
"# # # # # # #",
"# ##### # #####",
"## # # # # #",
"# #####r##### #",
"# # # # # # #",
"# ### # # # # #",
"# # # # # # #",
"### # # # # #",
"# ##### # # # # #",
"# # # # # # #",
"# ##### # # # # #",
"##### # # # # #",
"##### # # # # #",
"##### # # # # #",
"# ##### # # # # #",
"# # # # # # #",
"# ##### # # # # #",
"##### # # # # #"]

```

Funkcja powinna zwrócić: [True](#)