```
andreea@HP:Tema 3$ python3 maze.py input/harta1
Simple A* expected path length = 26.0
Probabilistic A* expected path length =  19.736
Using 100 exploration steps, the probabilistic A* expected path length is 22.736
Using 1000 exploration steps, the probabilistic A* expected path length is 20.6
Using 10000 exploration steps, the probabilistic A* expected path length is 20.516
andreea@HP:Tema 3$ python3 maze.py input/harta2
Simple A* expected path length = 26.0
Probabilistic A* expected path length =  23.084
Using 100 exploration steps, the probabilistic A* expected path length is 22.544
Using 1000 exploration steps, the probabilistic A* expected path length is 23.54
Using 10000 exploration steps, the probabilistic A* expected path length is 23.012
andreea@HP:Tema 3$ python3 maze.py input/harta3
Simple A* expected path length = 24.0
Probabilistic A* expected path length =  14.176
Using 100 exploration steps, the probabilistic A* expected path length is 16.64
Using 1000 exploration steps, the probabilistic A* expected path length is 14.05
Using 10000 exploration steps, the probabilistic A* expected path length is 14.54
andreea@HP:Tema 3$ python3 maze.py input/harta4
Simple A* expected path length = 37.0
Probabilistic A* expected path length =  12.564
Using 100 exploration steps, the probabilistic A* expected path length is 18.085
Using 1000 exploration steps, the probabilistic A* expected path length is 11.842
Using 10000 exploration steps, the probabilistic A* expected path length is 13.243
andreea@HP:Tema 3$
```

README - Detalii implementare tema 3 IA - Musat Andreea 342C4

La task 1 am folosit algoritmul A* considerand ca portalurile sunt pereti. La task 2 i-am permis agentului sa foloseasca portalurile pentru a se teleporta, destinatia fiind aleasa random, dar tinand cont de distributia de probabilitate cunoscuta pentru fiecare portal. La task 3, deoarece presupunem ca agentul nu dispune de aceste distributii de probabilitate, este lasat sa exploreze mediul pentru a incerca sa le invete; strategia prin care face asta este sa ia fiecare portal prin rotatie si sa se teleporteze prin el pana cand ramane fara pasi disponibili (pozitia initiala -> portal -> destinatie -> urmatorul portal etc). Atunci cand probabilitatile sunt cunoscute, agentul are cele mai bune rezultate, deoarece ia in calcul cat de aproape de destinatie sunt, in medie, locurile in care ar putea fi teleportat. La task 3 insa, rezultatele sunt variabile, in functie de cat de bine reuseste sa invete probabilitatile si in functie de 'noroc'.