

Zadanie: CUB

Cube3 - MegaCube



Etap ONTAK 2004. Dzień 3. Plik źródłowy cub.*

2004.08.25

Dostępna pamięć: 64 MB.

Grupa studentów pewnej uczelni, zafascynowana niedawno obejrzanym filmem - Cube2 - Hypercube, postanowiła nakręcić kolejną serię tego wspaniałego filmu. Z uwagi na niski budżet, jedyne na co mogą sobie pozwolić jest dwuwymiarowa kostka (no skoro dwuwymiarowa, to niech będzie przynajmniej prostokątna). W celu urozmaicenia filmu, każda komnata zostanie pomalowana na jeden z dostępnych kolorów. Główni bohaterowie filmu jako wskazówkę umożliwiającą im wydostanie się z labiryntu otrzymają mapę z zaznaczonymi kolorami poszczególnych komnat. Bohaterowie będą chodzić po labiryncie i starać się zlokalizować swą pozycję na podstawie mapy. Ponieważ jeden z aktorów przez przypadek zabrał ze sobą Palma, więc może sobie pozwolić na napisanie programu, który pozwoli określić pozycję w labiryncie. Pomóż mu w tym niebanalnym zadaniu.

Zadanie

Napisz program który:

- Wczyta opis labiryntu, po którym poruszają się bohaterowie oraz obszar zbadany przez nich (a znają tylko pewien obszar labiryntu, który jest reprezentowany przez prostokąt),
- wyliczy, w ilu różnych miejscach mogą się znajdować bohaterowie filmu,
- wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu standardowego wejścia zapisane są cztery liczby naturalne c_x, c_y, m_x, m_y ($1 \leq m_x \leq c_x \leq 1000, 1 \leq m_y \leq c_y \leq 1000$). c_x i c_y to odpowiednio wymiary labiryntu (czyli również jego mapy), a m_x i m_y to wymiary zwiedzzonego obszaru. W kolejnych c_y wierszach znajduje się po c_x liczb naturalnych (nie większych od 100), reprezentujących kolory poszczególnych pomieszczeń na mapie. Następnie znajduje się m_y wierszy zawierających po m_x liczb naturalnych (podobnie nie większych od 100), reprezentujących kolory kolejnych pomieszczeń, zwiedzonych przez bohaterów.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia powinna się znajdować jedna liczba naturalna, odpowiadająca liczbie miejsc, w których mogą się znajdować aktorzy.

Przykład

Dla pliku wejściowego cub.in:

```
5 3 2 2
```

1 2 2 1 3

2 3 1 2 2

1 4 2 3 3

1 2

2 3

poprawnym wynikiem jest plik wyjściowy cub.out:

2