

OPIS PROJEKTU:

Dana jest prostokątna mapa podzielona na kwadraty. Każde pole mapy może być puste (nieprzejezdne), być fragmentem drogi (przejezdnej w każdą stronę) lub miastem (przejezdnym tak jak droga). Nazwa każdego z miast jest podana na mapie i każda litera zajmuje jedno pole (pole z literą jest nieprzejezdne). Przebycie jednego pola zajmuje minutę. Poruszać się możemy tylko pomiędzy polami sąsiadującymi bokami. Dodatkowo wybrane miasta połączone są jednokierunkowymi połączeniami lotniczymi. Nazwa miasta będzie sąsiadowała (bokiem lub rogami) z miastem pierwszą lub ostatnią literą. Nazwa będzie jednoznacznie przypisana do miasta. Pole przed i za nazwą miasta (o ile będzie takie pole) nie pojawi się cyfra lub litera.

Uwagi implementacyjne:

Nie można używać klasy string ani kontenerów z biblioteki STL.

Można używać elementów języka C++ takich jak klasy, referencje, itp.

Można zaimplementować własną klasę string.

Czytelność kodu jest ważna, a komentarze (które są oczywiście dozwolone) niekoniecznie ją zapewniają.

Smartpointer'y z biblioteki standardowej są zabronione, ale nadal można zaimplementować własną klasę inteligentnych wskaźników.

Proszę pamiętać o odzyskiwaniu pamięci podczas usuwania elementów implementowanych struktur danych.

Przykłady:/Examples:

Tak może być:/This can be:

```
*GDANSK....OPOLE....RUMIA
...*GDYNIA.....*.....*...
*SOPOT.....*.....*
...PUCK*.*KRAKOW.REDA.HEL
```

Tak nie będzie:/This is incorrect:

```
**GDANSK.....OPOLE.
.....*GDYNIA*.....
*SOPOTHEL.....
.....*..REDA.*PUCK*..
```

Z A można dotrzeć do B:/It is possible to arrive from point A to B:

```
.....
A*####...#####*B..
.....#.C.#.....
.....##*##.....

.....
.A**B.
.....
```

Z A nie można dotrzeć do B:/It is not possible to arrive from point A to B:

```
....##*B
A*##....

....##*B
A*##*C...

.....*.....
A*##C##*B
```

Wejście:

W pierwszej linii wejścia pojawią się wymiary mapy: szerokość w i wysokość h. W następnych h liniach (każda po w znaków) podany będzie opis mapy. Każdy znak opisu to kropka (.) oznaczająca puste pole, krzyżyk (#) reprezentujący drogę, gwiazdka (*) symbolizujący miasto lub litera bądź cyfra, będąca częścią nazwy miasta.

W kolejnej linii pojawi się jedna liczba k – liczba połączeń lotniczych. Następne k linii to opis połączeń w postaci źródło cel czas, gdzie źródło to nazwa miasta startowego, cel to nazwa miasta docelowego a czas jest czasem przelotu w minutach. W kolejnej linii pojawi się jedna liczba q będąca liczbą zapytań. Każde zapytanie pojawi się w osobnej linii i będzie miało format: źródło cel typ. Jest to zapytanie o najkrótszy czas przejazdu od miasta źródło do miasta cel. Typ równy zero oznacza zapytanie tylko o czas. Gdy zapytanie ma typ równy jeden, należy także podać trasę przejazdu.

Wyjście:

Na wyjściu należy wypisać dla każdego zapytania jedną linię. Na początku linii powinna pojawić się liczba, będąca najkrótszym czasem podróży pomiędzy miastami. Jeżeli zapytanie ma typ równy jeden, należy także wypisać, po spacji, wszystkie miasta pośrednie (bez startowego i końcowego) w kolejności ich odwiedzania

Przykład:

```
Wejście:/Input:
20 20
.....GDANSK.....
.....*.....
.....#.....
.....#.....
*#####.
#SZCZECIN.....#.
#.....#.
##.....#.
.#####*#####.
.#...WARSZAWA.....
.#.....
.#####.
.#.....#.
.#..WROCLAW.##.....
.#.*.....*.....
.####.....#KIELCE.
.....*##.#####.
.OPOLE..#.*.....#.
.....#.KRAKOW..#.
.....#####.
0
3
KIELCE KRAKOW 0
KRAKOW GDANSK 0
KRAKOW GDANSK 1

Wyjście:/Output:
5
40
40 KIELCE SZCZECIN
```