

0 Modificări

v2 - 04/05/2014	Secțiunea 0	Creare secțiune
	Secțiunea 3	Modificare fișier intrare linia 2
	Secțiunea 5	Adăugare patru precizări

1 Enunț

O rețea de străzi dintr-un oraș este compusă din inteseccții și drumuri cu sens unic ce leagă aceste intersecții, fiecare intersecție având drumuri ce ies sau intră în intersecție.

Primăria orașului pune la dispoziție un fișier de date text ce reprezintă rețeaua de străzi a orașului în următorul format:

- prima linie conține numărul de intersecții N
- pe fiecare dintre următoarele N linii (i de la 2 la N+1) se găsesc informații despre fiecare intersecție:
 - numărul de drumuri care pleacă din intersecție: M
 - pentru fiecare drum
 - lungimea în km ca număr real
 - destinația ca întreg identificând intersecția destinație

O intersecție în acest oraș este identificată printr-un număr întreg de la 0 la N-1 iar un drum este identificat de cele două intersecții pe care le leagă (în ordinea origine, destinație)

O adresă din oraș este identificată de un drum și de distanța de la origine până la adresa respectivă. Distanța este între 0 și lungimea drumului.

2 Cerințe

1. Dându-se două adrese A1 și A2 să se determine traseul cel mai scurt dintre ele și să se afișeze sub forma unei liste de intersecții prin care trece.

Adresele se citesc de la intrare:

`origineA1 destinatieA1 distantaA1 origineA2 destinatieA2 distantaA2`

Se tipărește:

- `D I1 I2 I3 ...` - distanța în km (trei zecimale) și fiecare intersecție prin care

trece traseul (separate de un spațiu)

- `adresa invalida` – în cazul când una sau ambele adrese sunt invalide
- `traseu imposibil` – în cazul în care nu a fost găsit niciun traseu

2. Aceeași problemă ca la punctul 1 dar se va afișa o listă de instrucțiuni ce trebuie urmată pentru a ajunge de la A1 la A2.

Instrucțiunile sunt de tipul (vezi și exemplele):

- `mergeti inainte X.YYY km` – unde X.Y este distanța până la următoarea intersecție. Se vor afișa doar trei cifre după virgulă
- `virati catre P` – Unde P este următoarea intersecție
- `ati ajuns la destinatie`

3. BONUS

La acest pas programul va trebui să ghideze interactiv un șofer pentru a ajunge din poziția curentă la o adresă dată ca intrare (prin cele trei valori `origineA2` `destinatieA2` `distanțaA2`)

Programul va rula o buclă de interacțiune, fiecare interacțiune constând în citirea noii poziții a autovehicolului și afișarea unor instrucțiuni.

Noua poziție este dată de cele trei valori iar instrucțiunile pot fi:

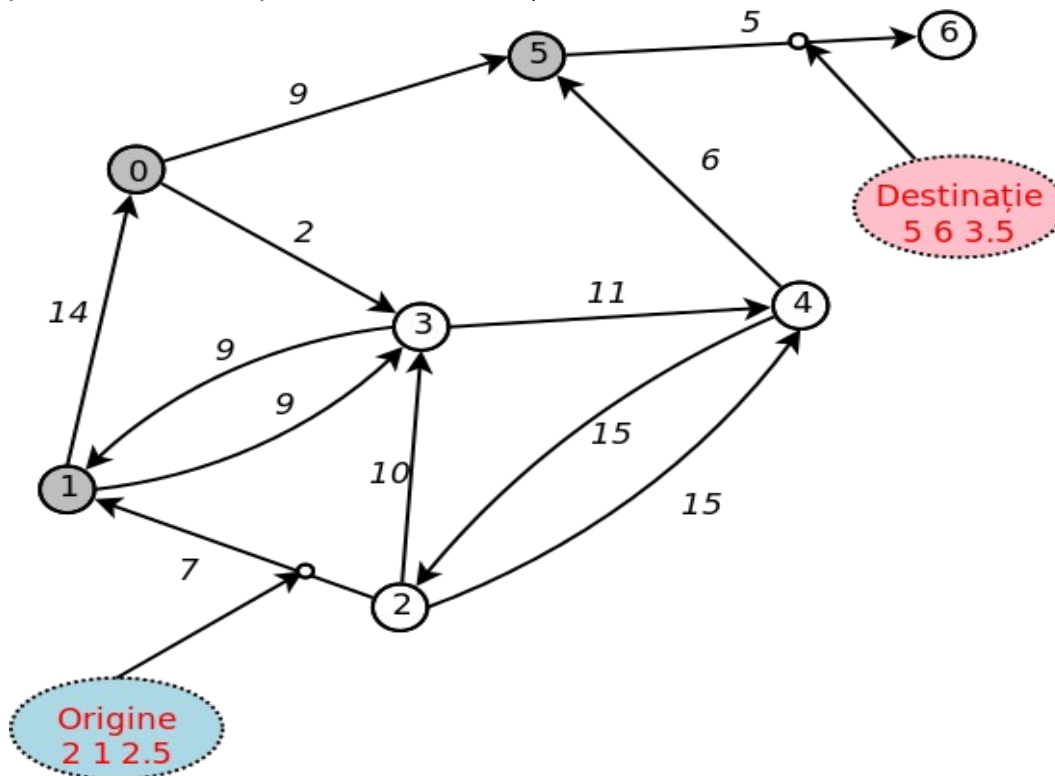
- `mergeti inainte`
- `in X metri virati catre P` – unde P este următoarea intersecție. Acest mesaj este afișat când poziția curentă este la mai puțin de 500m de următoarea intersecție. X este afișat ca un număr întreg de metri prin rotunjire.
- `in X metri ati ajuns la destinatie` – acest mesaj se afișează în aceleași condiții ca la punctul anterior
- `ati ajuns la destinatie` – se afișează când punctul curent este la cel mult 25m înainte sau după destinație
- `reconfigurare traseu` – se afișează aunci când poziția curentă este în afara traseului calculat anterior. În acest caz se recalculează un nou traseu și se afișează instrucțiunea corespunzătoare noului traseu și poziției curente.

Buclă de interacțiune se oprește odată cu mesajul `ati ajuns la destinatie`

Programul pe care trebuie să-l scrieți va primi ca prim parametru în linia de comandă cerința pe care trebuie să o rezolve (1, 2 sau 3) și în al doilea parametru numele fișierului care descrie rețeaua de străzi. Va citi de la intrarea standard datele de intrare (conform cerinței specificate) și va afișa răspunsul la ieșirea standard.

3 Exemplu

Se dă rețeaua de străzi din figură cu punctul de plecare este între intersecțiile 2 și 1, la 2.5 km de 2 iar punctul de sosire pe drumul dintre 5 și 6, la 3.5 km de 5.



Pentru cerința 1

Exemplul 1

Date de intrare:

2 1 2.5 5 6 3.5

Răspunsul este:

31.000 1 0 5

Exemplul 2

Date de intrare:

2 1 8.5 5 6 3.5

Răspunsul este:

adresa invalida

Exemplul 3

Date de intrare:

5 6 3.5 2 1 2.5

Răspunsul este:

traseu imposibil

Pentru cerința 2

Exemplul 4

Date de intrare:

2 1 2.5 5 6 3.5

Răspunsul este:

mergeti inainte 3.500 km

virati catre 0

mergeti inainte 14.000 km

virati catre 5

mergeti inainte 9.000 km

virati catre 6

mergeti inainte 3.500 km

ati ajuns la destinatie

Pentru cerința 3

Exemplul 5

Date de intrare:

Intrare <i>stdin</i>	Ieșire <i>stdout</i>
5 6 3.5 (destinația)	
2 1 2.5 (prima poziție)	mergeti inainte
2 1 6.8	in 200 metri virati catre 0
2 1 6.9	in 100 metri virati catre 0
1 3 1.1 (a virat spre 3)	reconfigurare traseu mergeti inainte
1 3 8.6	in 400 metri virati catre 4
3 1 3 (a virat spre 1)	reconfigurare traseu mergeti inainte
1 0 1 (punct pe drumul 1 0)	mergeti inainte
0 5 8.51	in 499 metri virati catre 6

5 6 1	mergeti inainte
5 6 3.15	in 350 metri ati ajuns la destinatie
5 6 3.48	ati ajuns la destinatie

Pentru exemplul de mai sus rețeaua de străzi este dată astfel

7

2 2 3 9 5

2 14 0 9 3

3 7 1 10 3 15 4

2 9 1 11 4

2 15 2 6 5

1 5 6

0

4 Punctaj

Cerința 1	50p
Cerința 2	50p
Cerința 3	20p

5 Precizări

- Termenul limită de predare a temei este 23 mai
- Nu există limită pentru numărul de noduri. Desigur, datele de test vor avea un număr rezonabil de noduri care să permită oricărui program corect care folosește alocarea dinamică să funcționeze
- În ceea ce privește limita de timp, dacă veți folosi structurile de date și algoritmi din curs, nu veți avea probleme cu timpul de rulare. Limita de timp va fi diferită pentru fiecare test în parte și va conține o marjă suficientă
- Programul trebuie să elibereze memoria alocată pe heap înainte de a se termina
- Pentru cerința 3, datele de intrare nu trebuie validate