

Zadaci

Zadatak	JOZO	LIST
Izvorni kôd	jozo.pas jozo.c jozo.cpp	list.pas list.c list.cpp
Ulazna datoteka	jozo.in	list.in
Izlazna datoteka	jozo.out	list.out
Vremensko ograničenje (po test podatku)	5 sekundi	5 sekundi
Broj test podataka	10	10
Broj bodova (po test podatku)	10	10
Ukupno bodova	100	100
	200	

Beskućnik Jozo je kupio mjesečnu kartu za vožnju vlakom na svim linijama kako bi mogao spavati u toplim vagonima i sanjati o boljem životu.

Zadan je popis svih stanica zajedno s prugama koje ih povezuju i njihovim duljinama (tj. vremenima potrebnima da vlak prijeđe udaljenost između te dvije stanice). Pruge su dvosmjerne i put traje jednako dugo u oba smjera.

Zadan je i popis svih vlakova koji voze zajedno s vremenima njihovih polazaka i stanicama kroz koje prolaze. Vlak staje na svakoj stanici kroz koju prolazi.

Jozo se na početku (u prvoj sekundi) nalazi u **stanici broj 1** i mora se **vratiti** na tu istu stanicu **između** T1-e i T2-e sekunde (**uključivo**). Ako su dva vlaka u isto vrijeme na istoj stanici, on može skočiti iz jednog u drugi bez utroška vremena.

Napišite program koji će odrediti neki način na koji će se Jozo voziti vlakovima tako da **ukupno vrijeme provedeno u stanicama bude što manje**.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se cijeli brojevi N , P , V , T_1 i T_2 , $2 \leq N \leq 1000$, $1 \leq V \leq 1000$, $1 \leq T_1 \leq T_2 \leq 50,000$. Broj N je broj stanica, P je broj pruga, V je broj vlakova, T_1 i T_2 su vremena iz teksta zadatka.

U svakom od sljedećih P redaka nalaze se podaci o jednoj pruzi tj. tri cijela broja S_1 , S_2 i T . To znači da put između stanice S_1 i S_2 (i obratno) traje T sekundi, $1 \leq T \leq 600$.

U svakom od sljedećih V redaka nalaze se podaci o jednom vlaku. Prvi broj u tom retku je broj T_0 , vrijeme polaska s prve stanice, drugi broj je NS , $1 \leq NS \leq 1000$, broj stanica na liniji na kojoj vlak vozi (uključujući prvu i zadnju), sljedećih NS brojeva su redom stanice kroz koje taj vlak prolazi. Vlak vozi od prve do zadnje stanice gdje svi putnici izlaze iz vlaka a vlak više ne vozi.

Svi brojevi koji se nalaze u istom retku međusobno su odvojeni s po jednim razmakom.

JOZO

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati traženo vrijeme iz teksta zadatka.

Test primjeri

jozo.in

```
4 4 3 30 35
1 2 5
2 3 2
2 4 7
3 4 3
2 4 1 2 4 3
14 4 3 4 2 3
28 3 3 2 1
```

jozo.out

6

jozo.in

```
4 6 5 80 100
4 2 6
2 1 16
1 3 17
1 4 19
4 3 9
3 2 10
25 3 1 3 2
25 3 1 2 4
4 4 1 2 3 4
52 4 4 2 1 4
64 4 2 3 4 1
```

jozo.out

22

jozo.in

```
4 6 7 80 100
4 1 8
1 3 7
3 2 15
1 2 2
2 4 1
4 3 3
50 7 2 4 1 2 4 1 3
25 10 4 3 1 2 4 3 1 2 4 1
6 6 2 1 3 4 2 1
11 5 4 2 3 1 4
52 6 1 2 4 3 2 1
23 5 3 2 4 1 2
21 5 4 2 1 3 2
```

jozo.out

23

LIST

Mirko i Slavko igraju jednu jako zanimljivu igru.

Mirko na komadu papira nacrtava neko **stablo** (acikličan neusmjereni povezan graf) s N vrhova koje označi brojevima od 1 do N .

Zatim otkida vrhove s tog stabla na sljedeći način: pogleda koji od **listova** (tj. vrhova koji su povezani sa **samo jednim** nekim drugim vrhom) na stablu je označen **najmanjim** brojem, **makne** taj list iz stabla i **zapiše oznaku vrha s kojeg je taj list otkinut**.

To radi sve dok mu u stablu ne ostane samo jedan vrh i na taj način on dobije niz od $(N-1)$ -og broja.

Tada nastupa Slavko koji na temelju tog niza pokušava **rekonstruirati** originalno Mirkovo stablo, i ako uspije, pobijedio je u igri.

Napišite program koji će pomoći Slavku da pobijedi Mirka.

Ulazni podaci

U prvom retku ulazne datoteke nalazi se cijeli broj N , $2 \leq N \leq 100,000$ (sto tisuća), broj vrhova u stablu.

U drugom retku se nalazi $N-1$ brojeva međusobno odvojenih s po jednim razmakom. Ti brojevi redom predstavljaju brojeve koje je Mirko zapisivao.

Izlazni podaci

U $N-1$ redaka izlazne datoteke treba ispisati bridove traženog stabla **u bilo kojem poretku**. Bridove ispisujete tako da navedete oznake vrhova (**u bilo kojem poretku**) između kojih se taj brid nalazi.

Test primjeri

list.in

5
4 3 4 5

list.out

1 4
2 3
3 4
4 5

list.in

7
1 2 2 2 1 7

list.out

3 1
1 7
2 1
4 2
5 2
6 2

list.in

10
6 2 7 9 2 9 4 4 10

list.out

8 4
7 5
3 2
9 6
9 4
10 4
1 6
2 9
2 7