

Odometar sa kamenčićima

Ratar Demjan je izumeo originalni "odometar": kolica koja mogu meriti rastojanja tako što ispuštaju po jedan kamenčić svaki put kada točak napravi pun krug. Kasnije, brojanjem kamenčića dobijamo koliko puta se točak okrenuo, što nam dozvoljava da izračunamo rastojanje koje je odometar (kolica) prešao. Kao svaki pravi ratar, Demjan je dodao softversku kontrolu odometru, povećavajući njegovu funkcionalnost. Vaš zadatak je da programirate odometar na osnovu dole navedenih pravila.

Tabla kretanja

Odometar se kreće na zamišljenoj kvadratnoj tabli dimenzija 256×256 jediničnih kvadratnih polja. Svako polje može sadržati najviše 15 kamenčića i definisano je koordinatama (red, kolona), gde je svaka koordinata iz segmenta $[0..255]$. Polje (i, j) je susedno poljima $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ i $(i, j + 1)$, ako ona postoje. Svako polje koji se nalazi u prvom ili poslednjem redu ili u prvoj ili poslednjoj koloni se naziva *granično*. Odometar uvek počinje u polju $(0, 0)$ (gornji-levi ugao), usmeren nagore.

Osnovne komande

Odometar je moguće programirati koristeći sledeće komande.

- `left` — okreni se za 90 stepeni ulevo (suprotno od smera kazaljke na satu) i ostani na trenutnom polju (npr. ako je bio okrenut nadole pre ove komande, posle komande će biti okrenut udesno).
- `right` — okreni se za 90 stepeni udesno (u smeru kazaljke na satu) i ostani na trenutnom polju (npr. ako je bio okrenut ulevo pre ove komande, posle komande će biti okrenut nagore).
- `move` — pomeri se za jedno polje (u smeru u kome je odometar trenutno okrenut) u odgovarajuće susedno polje. Ako takvo polje ne postoji (tj. došli smo do graničnog polja za dati smer) tada ova komanda nema efekta.
- `get` — ukloni jedan kamenčić sa trenutnog polja. Ako trenutno polje nema kamenčića tada ova komanda nema efekta.
- `put` — dodaj jedan kamenčić na trenutno polje. Ako trenutno polje već sadrži 15 kamenčića tada ova komanda nema efekta. Odometru nikada neće ponestati kamenčića.
- `halt` — prekini izvršavanje komandi.

Odometar izvršava komande u redosledu u kome su date u programu, jedna komanda po liniji. Prazne linije se ignorišu. Simbol # označava komentar; tekst koji sledi iza ovog simbola, pa sve do kraja linije, se ignoriše. Ako odometar stigne do kraja programa, izvršavanje komandi se prekida.

Primer 1

Posmatrajmo sledeći program za odometar. Njegovo izvršavanje dovodi odometar do polja (0, 2), pri čemu je u tom polju okrenut udesno. (Primetimo da je prva `move` komanda ignorisana, jer je u tom trenutku odometar u gornjem-levom ugaonom polju i usmeren je nagore.)

```
move # nema efekta
right
# sada je odometar okrenut udesno
move
move
```

Labele, granice i kamenčići

Kako bi ponašanje obometra zavisilo od trenutnog stanja, možemo da koristimo labele. Labele su stringovi sastavljeni od `a, ..., z, A, ..., Z, 0, ..., 9` čija dužina nije veća od 128, pri čemu se velika i mala slova tretiraju kao različita. Nove komande koje se odnose na labele su navedene u nastavku teksta. U ostatku opisa `L` označava bilo koju ispravnu labelu.

- `L`: (tj. `L` za kojim sledi dvotačka `:`) — deklarise poziciju labele `L` u programu. Sve deklarisanе labele moraju biti jedinstvene. Deklarisanje labela nema uticaja na odometar.
- `jump L` — nastavlja izvršavanje programa bezuslovnim skokom na liniju sa labelom `L`
- `border L` — izvršavanje programa se nastavlja skokom na liniju sa labelom `L`, ako je odometar na granici okrenut licem ivici table (tj. komanda `move` nema efekta); inače izvršavanje programa se nastavlja normalno i ova komanda nema uticaja.
- `pebble L` — izvršavanje programa se nastavlja skokom na liniju sa labelom `L`, ako trenutno polje sadrži bar jedan kamenčić; inače izvršavanje programa se nastavlja normalno i ova komanda nema uticaja.

Primer 2

Sledeći program nalazi prvi kamenčić u redu 0 i tu staje; ako nema kamenčića u redu 0, program se završava na ivici kod kraja reda. Program koristi dve labele `leonardo` i `davinci`.

```
right
leonardo:
pebble davinci # kamenčić je nađen
border davinci # kraj reda
move
jump leonardo
davinci:
halt
```

Odometar počinje program okretanjem udesno. Petlja startuje sa definicijom labele `leonardo:` i završava se komandom `jump leonardo`. U petlji, odometar proverava da li postoji kamenčić ili

je došao do granice na kraju reda; ako ovaj uslov nije ispunjen, odmetar izvršava `move` sa trenutnog polja $(0, j)$ na susedno $(0, j + 1)$ pošto to polje sigurno postoji. Primetimo da komanda `halt` nije ovde neophodna, pošto se program svakako završava.

Postavka problema

Vaš zadatak je da napišete program na jeziku odometra, koji će pokretati odometar na željeni način. Svaki od podzadataka (vidi ispod) opisuje kako bi odometar trebalo da se ponaša i daje dodatna ograničenja za program. Ova ograničenja mogu biti:

- *Veličina programa* — program mora biti dovoljno kratak. Veličina programa je jednaka broju komandi u njemu. Definicije labela, komentare i prazni redovi se ne računaju u veličinu programa.
- "Dužina izvršavanja" ' Program mora da se završi dovoljno brzo. Dužina izvršavanja je broj obavljenih "koraka": Svako izvršavanje komandi se broji kao korak, bez obzira da li je izvršavanje komande imalo efekta ili ne; definicije labela, komentari i prazni redovi se ne broje u korake.

U Primeru 1, veličina programa je 4 i dužina izvršavanja je 4. U Primeru 2, veličina programa je 6 i kada se program izvrši na tabli sa jednim kamenčićem u polju $(0, 10)$, dužina izvršavanja je 43 koraka: `right`, 10 iteracija petlje, gde svaka iteracija ima 4 koraka (`pebble davinci`; `border davinci`; `move`; `jump leonardo`), i na kraju `pebble davinci i halt`.

Podzadatak 1 [9 bodova]

Na početku postoji x kamenčića na polju $(0, 0)$ i y kamenčića na polju $(0, 1)$, dok su sva ostala polja prazna. Setimo se da polje može sadržati najviše 15 kamenčića. Napišite program koji dovodi odometar u polje $(0, 0)$ ako je $x \leq y$, odnosno u polje $(0, 1)$ u suprotnom. (Smer u kome je odometar okrenut na kraju nije bitan. Osim toga, nije bitno koliko je kamenčića na kraju na tabli niti gde se oni nalaze.)

Ograničenja: veličina programa ≤ 100 , dužina izvršavanja $\leq 1\,000$.

Podzadatak 2 [12 bodova]

Isti kao i prethodni podzadatak ali kada se program završi, polje $(0, 0)$ mora sadržati tačno x kamenčića, a polje $(0, 1)$ tačno y kamenčića.

Ograničenja: veličina programa ≤ 200 , dužina izvršavanja $\leq 2\,000$.

Podzadatak 3 [19 bodova]

Postoje tačno dva kamenčića negde u redu 0: jedan je u polju $(0, x)$, a drugi u polju $(0, y)$: x i y su međusobno različiti i $x + y$ je paran broj. Napišite program koji ostavlja odometar u polju $(0, (x + y) / 2)$, odnosno tačno na pola između dva polja koja inicijalno sadrže kamenčiće. Stanje table po završetku nije bitno.

Ograničenja: veličina programa ≤ 100 , dužina izvršavanja $\leq 200\,000$.

Podzadatak 4 [do 32 boda]

Postoji najviše 15 kamenčića na tabli, nikoja dva na istom polju. Napišite program koji ih sve sakuplja u gornji-levi ugao; preciznije, ako postoji x kamenčića na tabli na početku, na kraju treba biti tačno x kamenčića u polju $(0, 0)$ i nijedan kamenčić u nekom drugom polju.

Broj bodova za ovaj podzadatak zavisi od dužine izvršavanja poslatog programa. Preciznije, ako je L maksimalna dužina izvršavanja programa među svim test primerima, vaš rezultat će biti:

- 32 boda, ako je $L \leq 200\,000$;
- $32 - 32 \log_{10}(L / 200\,000)$ bodova, ako je $200\,000 < L < 2\,000\,000$;
- 0 bodova, ako je $L \geq 2\,000\,000$.

Ograničenja: veličina programa ≤ 200 .

Podzadatak 5 [do 28 bodova]

Svako polje table može sadržati proizvoljan broj kamenčića (naravno, između 0 i 15). Napišite program koji nalazi minimum, tj., koji dovodi odometar na polje (i, j) takvo da svako drugo polje sadrži bar onoliko kamenčića koliko sadrži polje (i, j) . Posle završetka programa, broj kamenčića u svakom polju mora biti isti kao i na početku.

Broj bodova za ovaj podzadatak zavisi od veličine poslatog programa (P). Preciznije, vaš rezultat će biti:

- 28 bodova ako je $P \leq 444$;
- $28 - 28 \log_{10}(P / 444)$ bodova ako je $444 < P < 4\,440$;
- 0 bodova ako je $P \geq 4\,440$.

Ograničenja: dužina izvršavanja $\leq 44\,400\,000$.

Detalji implementacije

Potrebno je poslati po jedan fajl (program) za svaki podzadatak, ispisan na osnovu gore definisanih sintaksnih pravila. Svaki poslati fajl ne sme biti veći od 5 MiB. Za svaki podzadatak, vaš odometar-program će biti testiran na nekoliko test primera i dobićete neke informacije u vezi sa resursima koje je koristio vaš program (veličina programa i dužina izvršavanja). U slučaju da program nije sintaksno korektan (i prema tome nemoguće ga je testirati) dobićete informaciju o tipu sintaksne greške.

Nije neophodno da među trenutno poslatim fajlovima (odometer-programima) budu prisutni programi za sve podzadatke. Ako vaš trenutno poslati skup fajlova (submission) ne sadrži odometar-program za podzadatak X, poslednji poslati fajl za podzadatak X je automatski uključen u testiranje; ako do sada nije poslat odometar-program za podzadatak X, taj podzadatak dobija 0 poena.

Kao i obično, rezultat za poslate fajlove (submission) je suma rezultata ostvarenih na svakom podzadatku, a finalni rezultat za zadatak je maksimalna vrednost rezultata poslatih fajlova na koje su iskorišćeni tokeni, kao i rezultata poslednjeg poslatog skupa fajlova.

Simulator

Za potrebe testiranja, biće vam obezbeđen odometar-simulator, kome možete prosleđivati vaše odometar-programe i table sa definisanim početnim rasporedom kamenčića. Odometar-programi za simulator koriste istu sintaksu kao i oni koje treba slati na evaluaciju (tj. sintaksa opisana gore).

Opis table biće zadat u sledećem formatu: svaka linija fajla mora sadržati tri broja, R, C i P, koji označavaju da polje u preseku reda R i kolone C sadrži tačno P kamenčića. Za polja koja nisu pomenuta u opisu table se podrazumeva da sadrže 0 kamenčića. Na primer, posmatrajmo sledeći fajl:

```
0 10 3
4 5 12
```

Tabla opisana ovim fajlom sadrži 15 kamenčića: 3 na polju (0, 10) i 12 na polju (4, 5).

Odometar-simulator možete pokrenuti pozivanjem koda `simulator.py` (koji se nalazi u vašem folderu za zadatak) prosleđivanjem naziva fajla odometar-programa kao argument. Odometar-simulator prihvata sledeće opcije iz komandne linije:

- `-h` daje kratak opis svih mogućih opcija;
- `-g GRID_FILE` učitava opis table iz fajla `GRID_FILE` (ukoliko se ne prosledi ime fajla, odometar-simulator podrazumeva da je cela tabla prazna);
- `-s GRID_SIDE` postavlja veličinu table na `GRID_SIDE` x `GRID_SIDE` (default vrednost je 256, kao i u opisu problema); mogućnost korišćenja manjih tabli je korisno zbog debugg-ovanja.
- `-m STEPS` ograničava dužinu izvršavanja na najviše `STEPS` koraka;
- `-c` ulazi u "kompajlirajući" mod; u kompajlirajućem modu, odometar-simulator vraća potpuno isti izlaz kao i bez `-c` opcije, ali umesto da radi simulaciju u Python-u, generiše i kompajlira mali C program. Uz ovu opciju, inicijalizacije u programu traju duže, ali se kasnije dobijaju značajno brži rezultati; savetujemo vam da koristite ovu opciju kada se od vašeg odometar-programa očekuje da izvrši broj koraka reda veličine 10 000 000.

Broj submission-a

Maksimalan broj submission-a za ovaj zadatak je 128.

