

| ZADATAK                | LINUX                                          | DELETE                                             | SREDINA                                                |
|------------------------|------------------------------------------------|----------------------------------------------------|--------------------------------------------------------|
| izvorni kôd            | linux.pas<br>linux.c<br>linux.cpp<br>linux.cxx | delete.pas<br>delete.c<br>delete.cpp<br>delete.cxx | sredina.pas<br>sredina.c<br>sredina.cpp<br>sredina.cxx |
| ulazni podaci          | standardni ulaz                                |                                                    |                                                        |
| izlazni podaci         | standardni izlaz                               |                                                    |                                                        |
| vremensko ograničenje  | 1 sekunda                                      | 5 sekundi                                          | 3 sekunde                                              |
| memorijsko ograničenje | 256 MB                                         |                                                    |                                                        |
| broj bodova            | 40                                             | 50                                                 | 60                                                     |
|                        | 150                                            |                                                    |                                                        |



Agencija za odgoj i obrazovanje  
Education and Teacher Training Agency

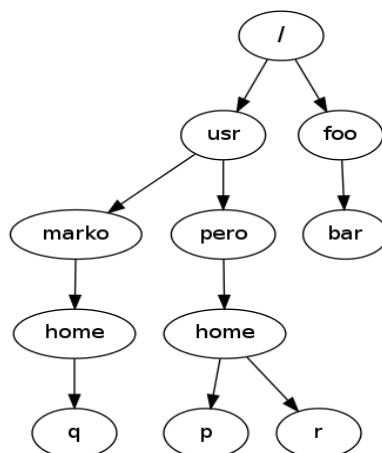


MINISTARSTVO ZNANOSTI, OBRAZOVANJA  
I ŠPORTA REPUBLIKE HRVATSKE

Napišite program koji će simulirati niz naredbi za rad s direktorijima sličnih (ali ne identičnih) onima u LINUX operativnom sustavu.

Za ovu priliku ime direktorija je niz od jednog ili više malih slova engleske abecede. Naravno, direktoriji mogu **sadržavati druge direktorije** te tako čine hijerarhiju kao na slici dolje. Poseban slučaj je takozvani korijenski ili "root" direktorij koji se nalazi na vrhu hijerarhije te nema ime. Dozvoljeno je da dva različita direktorija imaju isto ime, ali **ne u slučaju ako se nalaze u istom direktoriju**.

*Putanja* je niz znakova koji jedinstveno označava pojedini direktorij te se definira na sljedeći način: Ako su  $D_1, D_2, \dots, D_N$  direktoriji te je  $D_1$  sadržan u root direktoriju, a svi ostali sadržani u prethodnom direktoriju u nizu ( $D_2$  je sadržan u  $D_1$ ,  $D_3$  je sadržan u  $D_2$  itd.) onda je putanja direktorija  $D_N$  jednaka  $'/D_1/D_2/\dots/D_N'$ . Putanja root direktorija je samo  $'/'$ . Tako je na primjeru na Slici 1 putanja direktorija s imenom 'pero' jednaka  $'/usr/pero'$ , a putanje do dva direktorija s imenom 'home' su  $'/usr/marko/home'$  i  $'/usr/pero/home'$ .



Slika 1: Primjer hijerarhije direktorija

Na početku postoji samo root direktorij, koji je postavljen kao *trenutni direktorij*. Napišite program koji će čitati naredbe, za svaku naredbu simulirati zadane operacije na direktorijima te ispisati izlaz prema sljedećim pravilima:

- Naredba: **'mkdir X'**, gdje je X ime direktorija
  - Ukoliko u trenutnom direktoriju već postoji direktorij s imenom X, ispisuje se 'greska'.
  - Inače se stvara novi direktorij s imenom X u trenutnom direktoriju te ispisuje putanja novog direktorija.
  - Trenutni direktorij se ne mijenja.
- Naredba: **'rmdir X'**, gdje je X ime direktorija
  - Ukoliko u trenutnom direktoriju ne postoji direktorij s imenom X ispisuje se 'greska'.
  - Ukoliko X postoji, ali nije prazan tj. ako X sadrži druge direktorije ispisuje se 'greska'.
  - Inače se direktorij X briše te se ispisuje putanja obrisano direktorija.
  - Trenutni direktorij se ne mijenja.
- Naredba: **'cd X'**, gdje je X ime direktorija
  - Ukoliko u trenutnom direktoriju ne postoji direktorij s danim imenom ispisuje se 'greska'.
  - Trenutni direktorij postaje X te se ispisuje nova putanja trenutnog direktorija.
- Naredba: **'cd ..'** ('cd' pa točno jedan znak razmaka, pa dvije točke)
  - Ukoliko je trenutni direktorij korijenski direktorij ispisuje se 'greska'
  - Inače trenutni direktorij postaje direktorij u kojem je neposredno sadržan stari trenutni direktorij te se ispisuje nova putanja trenutnog direktorija.

- Naredba: **'find X'**, gdje je *X* ime direktorija
  - Ukoliko u trenutnom direktoriju ne postoji direktorij s imenom *X*, ispisuje se 'greska'.
  - Inače se ispisuje putanja direktorija *X* te putanje svih direktorija, direktno ili indirektno, sadržanih u direktoriju *X*. Putanje se ispisuju proizvoljnim redoslijedom, odvojene znakom ':' (dvotočka).
  - Trenutni direktorij se ne mijenja.

Obratite pažnju da je u svim navedenim naredbama *X* ime direktorija, a ne putanja. Dakle naredbe poput 'cd ../foo', 'mkdir /a/b' ili 'rmdir mirko/slavko' nisu dopuštene u ovom zadatku.

### **ULAZNI PODACI**

U prvom redu nalazi se prirodan broj  $N$  ( $1 \leq N \leq 100$ ), broj naredbi.

U svakom od sljedećih  $N$  redova nalazi se po jedna naredba. Svaka naredba sastoji se od malih slova engleske abecede, razmaka i znaka '.' (točka) te će točno odgovarati jednoj od naredbi iz teksta zadatka. Ime svakog direktorija *X* u naredbama 'mkdir *X*', 'rmdir *X*', 'cd *X*' i 'find *X*' sastojat će se od najviše 10 malih slova engleske abecede.

### **IZLAZNI PODACI**

Potrebno je ispisati  $N$  redova, rezultate izvršavanja svake zadane naredbe kako je opisano u tekstu zadatka, onim redoslijedom kojim su naredbe zadane u ulazu.

### **BODOVANJE**

U 30% test podataka neće se pojavljivati niti jedna od dvije varijante naredbe 'cd'.

U drugih 20% test podataka neće se pojavljivati naredba 'find'.

PRIMJERI TEST PODATAKA

|                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ulaz</b><br><br>10<br>mkdir a<br>find a<br>mkdir a<br>find a<br>mkdir b<br>find c<br>mkdir c<br>find c<br>rmdir d<br>rmdir b<br><br><b>izlaz</b><br><br>/a<br>/a<br>greska<br>/a<br>/b<br>greska<br>/c<br>/c<br>greska<br>/b | <b>ulaz</b><br><br>13<br>cd ..<br>mkdir a<br>mkdir a<br>cd b<br>cd a<br>mkdir b<br>cd ..<br>rmdir b<br>rmdir a<br>cd a<br>rmdir b<br>cd ..<br>rmdir a<br><br><b>izlaz</b><br><br>greska<br>/a<br>greska<br>greska<br>/a<br>/a/b<br>/<br>greska<br>greska<br>/a<br>/a/b<br>/<br>/a | <b>ulaz</b><br><br>11<br>mkdir alfa<br>cd alfa<br>mkdir b<br>cd b<br>mkdir p<br>mkdir q<br>cd ..<br>find beta<br>find b<br>cd ..<br>find alfa<br><br><b>izlaz</b><br><br>/alfa<br>/alfa<br>/alfa/b<br>/alfa/b<br>/alfa/b/p<br>/alfa/b/q<br>/alfa<br>greska<br>/alfa/b:/alfa/b/p:/alfa/b/q<br>/<br>/alfa:/alfa/b:/alfa/b/p:/alfa/b/q |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Mirko se opet bavi UNIX-om. Nedavno je otkrio da prilikom brisanja datoteka u komandnoj liniji ne mora nužno upisati točno ime datoteke koju briše nego može koristiti jednostavne regularne izraze.

Za ovu priliku se imena datoteka sastoje od samo malih slova engleske abecede. *Regularan izraz* je niz znakova sastavljen od malih slova engleske abecede i znakova `*` (zvjezdica). Tako su na primjer `*`, `ab`, `a*` i `b*c` regularni izrazi.

Kažemo da ime datoteke  $X$  odgovara regularnom izrazu  $R$  ako se  $X$  može dobiti od  $R$  tako da se svaka zvjezdica u  $R$  zamjeni proizvoljnim nizom malih slova koji može biti i prazan (drugim riječima zvjezdica se može i obrisati). Zvjezdice na različitim pozicijama mogu se zamijeniti različitim nizovima slova.

Ako kod brisanja datoteke umjesto imena datoteke koju želimo izbrisati napišemo regularni izraz, tada će se od svih datoteka u direktoriju obrisati točno one čije ime odgovara tom regularnom izrazu. Na primjer, brisanje izraza `bb*c` obrisat će datoteke `bbc`, `bbcd`, `bbabaccccd`, ali neće obrisati datoteke `bbd`, `abbc`, `bacd`, `abc`.

Na njegovom računalu nalazi se više direktorija, a svaki sadrži jednu datoteku koju Mirko želi obrisati te još nekoliko datoteka koje ne smije obrisati. Napišete program koji će, za svaki zadani direktorij, odrediti neki najkraći regularan izraz kojim Mirko može obrisati zadanu datoteku bez da obriše ostale datoteke u tom direktoriju.

## ULAZNI PODACI

U prvom redu nalazi se prirodan broj  $D$  ( $1 \leq D \leq 5$ ) koji označava broj različitih direktorija nakon čega slijedi prazan red. Nakon toga slijedi  $D$  blokova međusobno odvojenih praznim redom, od kojih svaki blok predstavlja jedan direktorij.

U prvom redu bloka nalazi se ime datoteke koju je potrebno obrisati. U drugom redu bloka nalazi se prirodan broj  $N$  ( $1 \leq N \leq 50$ ), koji označava broj datoteka koje ne smiju biti obrisane. Sljedećih  $N$  redova bloka sadrži imena datoteka koje ne smiju biti obrisane, svako ime u jednom redu. Imena svih datoteka u jednom direktoriju (uključujući i datoteku koju je potrebno obrisati) će biti različita i sastojati će se od najviše 20 malih slova engleske abecede.

## IZLAZNI PODACI

Izlaz se sastoji od  $D$  redova, gdje svaki red sadrži najkraći izraz kojim se briše zadana datoteka bez da ostale datoteke u direktoriju budu obrisane.

**Napomena:** Rješenje će uvijek postojati, ali ne mora biti jedinstveno.

PRIMJERI TEST PODATAKA

|                                                                                                                                               |                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>ulaz</b><br><br>2<br><br>aac<br>2<br>bd<br>aae<br><br>zatvor<br>3<br>zatvora<br>bzatvor<br>czatvorddd<br><br><b>izlaz</b><br><br>*c<br>z*r | <b>ulaz</b><br><br>1<br><br>bbabaccccd<br>4<br>bbd<br>abbc<br>bacd<br>abc<br><br><b>izlaz</b><br><br>bba* |
|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|

Medijan niza (ne nužno različitih) cijelih brojeva definiramo kao **srednji element** u nizu koji se dobije njegovim sortiranjem. Ukoliko je niz parne duljine, tada medijanom smatramo **manji od dva srednja broja** nakon sortiranja.

Na primjer, medijani nizova (1), (1, 3), (4, 5, 4), (4, 1, 2, 3) i (5, 4, 1, 2, 3) su redom 1, 1, 4, 2 i 3.

Ako je zadana tablica nenegativnih cijelih brojeva koja se sastoji od  $R$  redaka i  $S$  stupaca te prirodni broj  $K$ , napišite program koji će odrediti broj pravokutnika u tablici takvih da je medijan svih brojeva u tom pravokutniku barem  $K$ .

### ULAZNI PODACI

U prvom redu nalaze se tri prirodna broja  $R$ ,  $S$  i  $K$  ( $1 \leq R \leq 5\,000$ ,  $1 \leq S \leq 50$ ,  $1 \leq K \leq 1\,000\,000\,000$ ).

U svakom od sljedećih  $R$  redova nalazi se niz od točno  $S$  nenegativnih cijelih brojeva odvojenih razmakom. Svi elementi tablice su nenegativni cijeli brojevi manji ili jednaki  $1\,000\,000\,000$ .

### IZLAZNI PODACI

U prvi i jedini redak ispišite broj pravokutnika čiji medijan iznosi barem  $K$ .

### BODOVANJE

U 10% test podataka  $R$  i  $S$  će biti najviše 20.

U dodatnih 20% test podataka  $R$  će biti najviše 100.

### PRIMJERI TEST PODATAKA

|                                                                                  |                                                                                                    |                                                                                                                                                   |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ulaz</b><br><br>3 3 5<br>5 8 1<br>6 3 4<br>2 1 4<br><br><b>izlaz</b><br><br>8 | <b>ulaz</b><br><br>4 4 5<br>1 2 3 4<br>1 7 7 2<br>4 5 6 4<br>4 3 2 1<br><br><b>izlaz</b><br><br>21 | <b>ulaz</b><br><br>7 5 5<br>0 6 5 4 2<br>8 6 1 7 6<br>9 2 2 3 3<br>2 9 1 3 6<br>2 1 5 5 6<br>5 0 1 0 1<br>3 8 6 5 6<br><br><b>izlaz</b><br><br>93 |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|

Pojašnjenje prvog primjera:

5 8 1  
6 3 4  
2 1 4

Pravokutnik kojeg čine podebljani elementi (presjek prva dva retka i prva dva stupca) ima medijan 5 jer kada zapišemo njegove elemente te sortiramo niz dobivamo 3, 5, 6, 8.

Još neki pravokutnici koji imaju medijan barem 5 su i sami elementi 5, 6 i 8, kao i cijeli prvi stupac i cijeli prvi redak.

Kada prebrojimo sve takve pravokutnike dobivamo da ih ima točno 8.