

# Programiranje 1 — deveta domača naloga

Rok za oddajo: nedelja, 12. januarja 2020, ob 23:55

## Podatkovna baza

### Naloga

Podatkovna baza je sestavljena iz ene same tabele, ta pa je sestavljena iz elementov, razpojenih v vrstice in stolpce. Vsak element je lahko celoštevilskega, besedilnega ali logičnega tipa, vendar pa so v istem stolpcu vsi elementi istega tipa. Elementi celoštevilskega tipa so cela števila z intervala  $[-10^9, 10^9]$ , elementi besedilnega tipa so zaporedja največ 100 znakov (to so lahko samo črke angleške abecede, števke in podčrtaji), elementi logičnega tipa pa lahko zavzamejo le dve možni vrednosti: `true` in `false`. Na primer, tabela, prikazana kot tabela 1, vsebuje pet stolpcev: prvi in četrti sta celoštevilskega tipa, drugi in tretji besedilnega, peti pa logičnega tipa.

Tabela 1: Tabela iz obeh javnih testnih primerov.

1015	Jozhe	Gorishek	6	<code>true</code>
1033	Andrej	Gorishek	4	<code>false</code>
1036	Mojca	Kos	7	<code>true</code>
1048	Mojca	Kotnik	4	<code>true</code>
1075	Branko	Virk	-1	<code>true</code>
1099	Peter	Debeljak	10	<code>false</code>
1106	Jasna	Bizjak	5	<code>false</code>
1111	Mirko	Bizjak	7	<code>true</code>
1112	Mojca	Kotnik	4	<code>true</code>
1156	Dejan	Bizjak	7	<code>true</code>

Napišite program, ki z vhoda prebere tabelo, nato pa na njej po vrsti izvrši podano zaporedje ukazov in izpiše zahtevani element nastale tabele. Ukaze bomo opisali v sledečem razdelku.

### Vhod

V prvi vrstici vhoda je podano število vrstic ( $V \in [1, 100]$ ) in stolpcev ( $S \in [1, 100]$ ) tabele. V drugi vrstici je podano zaporedje  $S$  števil, ki podajajo tipe posameznih stolpcev: število 0 predstavlja celoštevilski, 1 besedilni, 2 pa logični tip. Sledi  $V$  vrstic s po  $S$  podatki, ki predstavljajo posamezne elemente tabele.

V naslednji vrstici je zapisano celo število  $U \in [0, 100]$ , nato pa sledi  $U$  vrstic z opisi posameznih ukazov. Prvi ukaz se nanaša na vhodno tabelo, drugi na tabelo, ki je rezultat prvega ukaza itd. Vsak ukaz zavzema eno od sledečih oblik:

- **stolpci**  $s_z \dots s_k$

Ta ukaz ohrani vse vrstice trenutne tabele, od stolpcev pa samo tiste z zaporednimi številkami (!) od vključno  $s_z$  do vključno  $s_k$  ( $1 \leq s_z \leq s_k \leq S$ ).

- **filtriraj**  $s$   $z$   $v$

Ta ukaz ohrani vse stolpce trenutne tabele, od vrstic pa samo tiste, pri katerih je  $s$ -ti element večji (če velja  $z = 1$ ), manjši (če velja  $z = -1$ ) ozziroma enak (če velja  $z = 0$ ) kot  $s$ -ti element v vrstici  $z$  zaporedno številko  $v$ . Pri stolpcih besedilnega in logičnega tipa vedno velja  $z = 0$ .

- **uredi**  $k$   $s_1$   $s_2 \dots s_k$

Ta ukaz uredi vrstice tabele glede na podanih  $k \in [1, S]$  kriterijev. Vrstice morajo biti primarno urejene po stolpcu  $z$  zaporedno številko  $|s_1|$  (absolutna vrednost števila  $s_1$ ). Vrstice z enakimi elementi v stolpcu  $|s_1|$  morajo biti urejene po stolpcu  $|s_2|$ . Vrstice, ki se ne razlikujejo niti po stolpcu  $|s_1|$  niti po stolpcu  $|s_2|$ , morajo biti urejene po stolpcu  $|s_3|$  itd. Predznak števila  $s_i$  podaja vrstni red urejenosti za stolpec  $s_i$ : pozitiven predznak predstavlja naraščajočo, negativen pa padajočo urejenost.

Pri besedilnem tipu je urejenost določena s sledečim pravilom: niz **a** sodi pred niz **b** natanko v primeru, če velja **a.compareTo(b) < 0** (pri naraščajočem urejanju) ozziroma **a.compareTo(b) > 0** (pri padajočem urejanju). Pri naraščajočem urejanju stolpca z logičnimi vrednostmi sodi **false** pred **true**, pri padajočem pa je ravno obratno.

Urejanje mora biti *stabilno*. To pomeni, da se vrstice, ki se ne razlikujejo v nobenem od stolpcev  $|s_1|, \dots, |s_k|$ , med seboj ne smejo menjati.

V zadnji vrstici vhoda sta podani celi števili  $v_0 \in [1, V]$  in  $s_0 \in [1, S]$ . Števili podajata zaporedni številki vrstice in stolpca elementa v končni tabeli, ki naj ga program izpiše.

Podatki znotraj iste vrstice so med seboj ločeni s po enim presledkom. Parametri ukazov in števili v zadnji vrstici vhoda se vedno nanašajo na dejansko obstoječe vrstice in stolpce trenutne tabele.

Skriti testni primeri 1–20 ne vsebujejo nobenih ukazov ( $U = 0$ ), primeri 21–30 vsebujejo samo ukaze **stolpci**, primeri 31–35 tudi ukaze **filtriraj**, primeri 36–50 pa tudi ukaze **uredi**, pri čemer za primere 36–40 velja  $k = 1$  (torej urejanje po enem samem stolpcu). Tabele v primerih 1–15 in 21–25 vsebujejo samo celoštivilske elemente.

## Izhod

Na izhod izpišite samo iskani element tabele, ki nastane po izvedbi celotnega zaporedja ukazov.

## Javni testni primer 1

Vhod:

```
10 5
0 1 1 0 2
1015 Jozhe Gorishek 6 true
1033 Andrej Gorishek 4 false
1036 Mojca Kos 7 true
1048 Mojca Kotnik 4 true
1075 Branko Virk -1 true
1099 Peter Debeljak 10 false
1106 Jasna Bizjak 5 false
```

```
1111 Mirko Bizjak 7 true
1112 Mojca Kotnik 4 true
1156 Dejan Bizjak 7 true
0
5 2
```

Izhod:

```
Branko
```

## Javni testni primer 2

Vhod:

```
10 5
0 1 1 0 2
1015 Jozhe Gorishek 6 true
1033 Andrej Gorishek 4 false
1036 Mojca Kos 7 true
1048 Mojca Kotnik 4 true
1075 Branko Virk -1 true
1099 Peter Debeljak 10 false
1106 Jasna Bizjak 5 false
1111 Mirko Bizjak 7 true
1112 Mojca Kotnik 4 true
1156 Dejan Bizjak 7 true
4
uredi 4 -5 4 -3 2
stolpci 3 5
filtriraj 2 1 8
filtriraj 1 0 3
2 3
```

Izhod:

```
true
```

Na sliki 1 so za javni testni primer 2 prikazane tabele po izvedbi posameznih ukazov.

## Oddaja naloge

Svoj program oddajte v obliki datoteke `DN09_vvvvvvvv.java`, pri čemer `vvvvvvvv` nadomestite s svojo vpisno številko. Enako kot pri prejšnji nalogi definirajte vse morebitne pomožne razrede kot statične notranje razrede.

Tabela po izvedbi ukaza **uredi** 4 -5 4 -3 2, ki uredi vrstice padajoče po stolpcu 5, naraščajoče po stolpcu 4, padajoče po stolpcu 3 in naraščajoče po stolpcu 2, vrstice, ki se ne razlikujejo po nobenem od teh štirih stolpcev, pa pusti v originalnem vrstnem redu:

1075	Branko	Virk	-1	<b>true</b>
1048	Mojca	Kotnik	4	<b>true</b>
1112	Mojca	Kotnik	4	<b>true</b>
1015	Jozhe	Gorishek	6	<b>true</b>
1036	Mojca	Kos	7	<b>true</b>
1156	Dejan	Bizjak	7	<b>true</b>
1111	Mirko	Bizjak	7	<b>true</b>
1033	Andrej	Gorishek	4	<b>false</b>
1106	Jasna	Bizjak	5	<b>false</b>
1099	Peter	Debeljak	10	<b>false</b>

Tabela po izvedbi ukaza **stolpci** 3 5:

Virk	-1	<b>true</b>
Kotnik	4	<b>true</b>
Kotnik	4	<b>true</b>
Gorishek	6	<b>true</b>
Kos	7	<b>true</b>
Bizjak	7	<b>true</b>
Bizjak	7	<b>true</b>
Gorishek	4	<b>false</b>
Bizjak	5	<b>false</b>
Debeljak	10	<b>false</b>

Tabela po izvedbi ukaza **filtriraj** 2 1 8, ki ohrani samo vrstice, pri katerih je element v stolpcu 2 večji od elementa v vrstici 8 in stolpcu 2, torej od števila 4:

Gorishek	6	<b>true</b>
Kos	7	<b>true</b>
Bizjak	7	<b>true</b>
Bizjak	7	<b>true</b>
Bizjak	5	<b>false</b>
Debeljak	10	<b>false</b>

Tabela po izvedbi ukaza **filtriraj** 1 0 3, ki ohrani samo vrstice, pri katerih je element v stolpcu 1 enak elementu v vrstici 3 in stolpcu 1, torej nizu Bizjak:

Bizjak	7	<b>true</b>
Bizjak	7	<b>true</b>
Bizjak	5	<b>false</b>

Slika 1: Obrazložitev javnega testnega primera št. 2.