

Tretji izpitni rok pri predmetu Programiranje 1

3. september 2024

Dopolnite in oddajte datoteke Prva.java, Druga.java, Tretja.java in Cetrta.java. Testirate jih lahko takole:

(1) tj.exe Prva.java . . (2) tj.exe Druga.java . . (3) tj.exe (4) tj.exe

Vsa števila, ki nastopajo v navodilih in testnih primerih tega izpita, so cela.

- ① Ada je izdelala nekaj nepraznih zaporedij števil z intervala $[1, 10^9]$ in jih po vrsti zapisala v vrstico, pri čemer je po koncu vsakega zaporedja zapisala še ničlo. Napišite program, ki izpiše število *strogo* padajočih Adinih zaporedij. Skupno število števil na vhodu znaša med 2 in 10^5 .

Primer 1 (vhod/izhod):

7 6 3 0 5 4 4 2 0 15 0 12 1 0 8 7 9 6 0

3

V tem primeru imamo pet zaporedij $\langle 7, 6, 3 \rangle$, $\langle 5, 4, 4, 2 \rangle$, $\langle 15 \rangle$, $\langle 12, 1 \rangle$ in $\langle 8, 7, 9, 6 \rangle$, od katerih so tri $\langle 7, 6, 3 \rangle$, $\langle 15 \rangle$ in $\langle 12, 1 \rangle$ strogo padajoča.

- ② Podanih je $n \in [1, 10^3]$ dokumentov. Vsak dokument je zaporedje besed iz fiksne besednjake z $b \in [1, 10^3]$ besedami. V prvi vrstici vhoda sta podani števili n in b , nato pa sledi še n vrstic; i -ta od njih se prične s številom $k_i \in [0, 10^3]$ (število besed v i -tem dokumentu), nato pa sledi k_i števil z intervala $[0, b - 1]$, ki podajajo indekse besed, ki tvorijo i -ti dokument.

Napišite program, ki izpiše indeks besede, ki nastopa v največ različnih dokumentih. V vseh testnih primerih je iskani indeks enolično določen.

V 50% testnih primerov so vse besede v okviru istega dokumenta med seboj različne.

Primer 1 (vhod/izhod):

6 8
5 3 0 5 7 3
7 2 1 2 0 1 2 4
6 3 5 4 2 0 4
1 3
0
4 2 1 2 3

3

Beseda z indeksom 3 nastopa v štirih dokumentih (v dokumentih z indeksi 0, 2, 3 in 5), kar je največ med vsemi besedami. (Za primerjavo: beseda 2 nastopa v treh, beseda 4 pa v dveh dokumentih.)

- ③ Televizijski program (razred **Program**) je neprazno zaporedje oddaj (razred **Oddaja**) v okviru istega dneva. Oddaja je lahko film (razred **Film**), informativna oddaja (razred **Info**) ali športna oddaja (razred **Sport**). Vsaka oddaja ima svoj naziv in trajanje v minutah (atributa **naziv** in **trajanje**). Oddaje si v programu sledijo zaporedno, brez vmesnih premorov.

V razredu **Program** dopolnite sledeče metode:

- [24%] `public Cas konec(Cas zacetek)`

Vrne časovni trenutek, ko se program zaključi, če se prične ob trenutku **zacetek**. (V vseh pripadajočih testnih primerih se program zaključi do 24. ure.)

- [26%] `public Film zadnjiFilm()`

Vrne zadnji film v programu **this**. Če program ne vsebuje nobenega filma, naj metoda vrne **null**.

- [26%] `public boolean infoDovoljPogosto(int maksRazmak)`

Vrne **true** natanko v primeru, če za vsako informativno oddajo (razen za zadnjo) velja, da časovni razmak med njenim zaključkom in pričetkom naslednje informativne oddaje ne presega **maksRazmak** minut. V vseh pripadajočih testnih primerih velja, da program vsebuje vsaj dve informativni oddaji.

- [24%] `public static Program sportVesDan(String osnova, int trajanje)`

Izdela in vrne program, ki je sestavljen iz športnih oddaj z nazivi *osnova_1*, *osnova_2*, ... Vsaka oddaja naj traja **trajanje** minut, njihovo število pa naj bo tolikšno, da pokrijejo celoten dan (na koncu dneva mora ostati manj kot **trajanje** minut). Na primer, če je parameter *osnova* niz *Nogomet*, parameter **trajanje** pa število 100, naj bo program sestavljen iz 100-minutnih oddaj z nazivi *Nogomet_1*, *Nogomet_2*, ..., *Nogomet_14*.

- ④ Dopolnite metodi, navedeni v nadaljevanju. Kot ponavadi se elementa *a* in *b* obravnavata kot enaka natanko v primeru, če velja *a.equals(b)*.

- [50%] `public static <T> Set<T> skupni(List<List<T>> s)`

Vrne množico elementov, ki nastopajo v vseh seznamih podanega seznama seznamov. Na primer, za seznam seznamov `[['E', 'G', 'C', 'F'], ['G', 'A', 'C', 'I'], ['F', 'C', 'G']]` je rezultat množica `{'C', 'G'}`.

- [50%] `public static <T> void poDolziniBrezDuplikatov(List<List<T>> s)`

Uredi sezname znotraj podanega seznama seznamov po naraščajočem številu medsebojno različnih elementov. Lahko predpostavite, da se vsi seznam v seznamu seznamov po tem kriteriju med seboj razlikujejo.

Na primer, metoda naj seznam seznamov `[['D', 'B', 'C'], ['E', 'F', 'E', 'F', 'F'], ['E', 'C', 'E', 'F', 'D'], ['E', 'F', 'E', 'F', 'F'], ['D', 'B', 'C'], ['E', 'C', 'E', 'F', 'D']]` uredi kot `[['E', 'F', 'E', 'F', 'F'], ['D', 'B', 'C'], ['E', 'C', 'E', 'F', 'D']]`. V urejenem seznamu seznamov ima prvi seznam dva medsebojno različna elementa, drugi tri, tretji pa štiri.