



ISPITNI
CENTAR
DRŽAVNO
TAKMIČENJE 2026.

AUTOR/AUTORKA TESTA _____

RECENZENT/RECENZENTKINJA _____

PODGORICA, _____ 20 _____ GODINE

SREDNJA ŠKOLA

PROGRAMIRANJE



Zadaci

Zadatak	Zadatak1	Zadatak2	Zadatak3
Izvorni kôd	zadatak1.java zadatak1.pas zadatak1.c zadatak1.cpp	zadatak2.java zadatak2.pas zadatak2.c zadatak2.cpp	zadatak3.java zadatak3.pas zadatak3.c zadatak3.cpp
Memorijsko ograničenje	256MB	256MB	256MB
Vremensko ograničenje (po test podatku)	1 sekunde	1 sekunda	1 sekunda
Ukupno bodova	33	33	34

Zadatak 1

Komisija treba da izvrši ocjenjivanje M maturalih radova. Za i -ti rad poznat je njegov broj strana $s[i]$, za svako $i = 0, 1, \dots, M-1$. Komisija je sačinjena od N članova, koji žele da podijele pregledanje radova tako što će svaki od njih pročitati određeni broj **uzastopnih** radova. Za pregledanje i -tog rada potrebno je vrijeme $K*s[i]$, gdje je K neki cijeli broj koji nam govori za koliko minuta se može pregedati jedna stranica.

Napisati program koji izračunava koliko je minimalno vremena potrebno da komisija ocijeni sve radove. Pretpostaviti da svi članovi mogu istovremeno da čitaju radove koji su im dodijeljeni (ne moraju da čekaju jedni druge).

Ulaz:

U prvom redu ulaza unose se brojevi $M(1 \leq M \leq 1\ 000\ 000)$, $N(1 \leq N \leq M)$ i $K(1 \leq K \leq 10^9)$, koji predstavljaju redom broj maturalih radova, broj članova komisije i koliko traje čitanje jedne stranice rada. U narednih M redova unose se vrijednosti $s[0], s[1], \dots, s[M-1](1 \leq s[i] \leq 10^2)$ tako da $s[i]$ predstavlja broj stranica i -tog rada.

Izlaz:

U jedinom redu izlaza štampati minimalno vrijeme (u minutima) potrebno da članovi komisije ocijene sve radove.

Ulaz:	Izlaz:
5 3 2 65 87 78 72 96	304

Pojašnjenje primjera:

Prvi član komisije čita prva dva rada za $(65+87)*2=304$ minuta, drugi sljedeća dva za $(78+72)*2=300$ minuta, i treći član čita posljednji rad za $96*2=192$ minuta. Potrebno vrijeme da svi završe je 304 minuta.

Rješenje:

Za rješavanje zadatka potrebno je primijeniti binarnu pretragu po maksimalnom broju strana koje čita jedan član komisije. Donju granicu binarne pretrage predstavlja najveći broj strana među svim radovima, jer neko mora ocijeniti najveći rad. Gornja granica je jednaka zbiru svih strana, što predstavlja situaciju u kojoj jedan član ocjenjuje sve radove.

Tokom binarne pretrage, za vrijednost x , koju provjeravamo, pokušavamo da dodijelimo radove članovima komisije tako što svaki član ocjenjuje najveći mogući broj uzastopnih radova koje može ocijeniti tako da ne pročita više od x strana, a zatim sljedeći član nastavlja od prvog neocijenjenog rada. Ako na taj način uspijemo rasporediti sve radove među najviše N članova komisije, tada pokušavamo da dodatno smanjimo maksimalan broj strana koje čita neki član komisije; u suprotnom, povećavamo broj strana. Kada se pretraga završi, dobijena vrijednost pomnožena sa K predstavlja minimalno vrijeme potrebno da komisija ocijeni sve radove.

Zadatak 2

Aleksej je u muzeju i želi vidjeti što je moguće više slika. Muzej se može predstaviti kao **matrica** dimenzija $n \times m$. Svako polje je ili prazno ili neprohodno. Prazna polja su označena sa '.', a neprohodna sa '*'. Svaka dva susjedna polja različitih tipova (jedno prazno i jedno neprohodno) razdvojena su zidom na kojem se nalazi jedna slika.

Na početku se Aleksej nalazi u nekom praznom polju. U svakom trenutku može se pomjeriti u bilo koje prazno polje koje dijeli stranu sa poljem u kojem se trenutno nalazi.

Za datu matricu koja predstavlja muzej i nekoliko početnih pozicija treba izračunati **maksimalan broj slika** koje Aleksej može vidjeti. Aleksej može vidjeti sliku samo ako se nalazi u polju koje je susjedno zidu sa tom slikom. Aleksej ima puno vremena, pa će pregledati svaku sliku koju može vidjeti.

Ulaz

Prvi red ulaza sadrži tri cijela broja n , m i k koji predstavljaju redom dimenzije muzeja i broj početnih pozicija za obradu. Za unijete brojeve važi $3 \leq n, m \leq 1000$ i $1 \leq k \leq \min(n \cdot m, 100000)$.

Svaki od narednih n redova sadrži m znakova '.' ili '*' koji predstavljaju opis muzeja. Garantovano je da su sva granična polja neprohodna, tako da Aleksej ne može izaći iz muzeja.

Svaki od narednih k redova sadrži po dva cijela broja x i y ($1 \leq x \leq n, 1 \leq y \leq m$) — red i kolonu jedne od Aleksejevih početnih pozicija. Garantovano je da su sve početne pozicije prazna polja.

Izlaz

Ispišite k cijelih brojeva takvih da i -ti broj predstavlja maksimalan broj slika koje Aleksej može vidjeti ako počne sa pozicije date u redu $n+i+1$ standardnog ulaza.

Primjer:

Ulaz:	Izlaz:
5 6 3 ***** *..*. ***** *....* ***** 2 2 2 5 4 3	6 4 10
4 4 1 **** *..* *..** **** 3 2	8

Rješenje

Potrebno je primijetiti da sva polja koja se nalaze u istoj komponenti povezanosti daju isti rezultat. Tako da je potrebno pokrenuti neki od algoritama pretrage kako bi svakom polju dodijelili broj koji predstavlja njegovu komponentu povezanosti.

Tokom obilaska za svako posjećeno slobodno, polje potrebno je prebrojati koliko od njegovih susjeda su označeni kao zauzeti i dodati u vrijednosti brojaču za tu komponentu. Tako dobijamo za svaku komponentu broj slika koje su dostupne iz bilo kojeg njenog polja.

Nakon preprocesiranja, za svaki upit (x,y) , potrebno je štampati vrijednost za komponentu kojoj polje (x,y) pripada.

Vremenska složenost rješenja je $O(n \cdot m)$ za preprocesiranje plus $O(1)$ po upitu. Memorija: $O(n \cdot m)$.

Zadatak 3

Dat je niz nenegativnih cijelih brojeva koji predstavljaju profil terena posmatran bočno, pri čemu svaki element označava stub širine 1. Potrebno je izračunati koliko jedinica kišnice ostaje zarobljeno između stubova nakon padavina.

Ulaz: U prvoj liniji nalazi se cio broj n koji predstavlja broj stubova ($1 \leq n \leq 10^5$). U drugoj liniji nalazi se n nenegativnih cijelih brojeva h_i ($0 \leq h_i \leq 10^4$), razdvojenih razmacima, koji predstavljaju visine stubova.

Izlaz: Ispisati jedan cio broj koji predstavlja ukupnu količinu zarobljene vode.

Ulaz:	Izlaz:
12 0 1 0 2 1 0 1 3 2 1 2 1	6
6 4 2 0 3 2 5	9



Slika koja odgovara prvom primjeru.

Rješenje

Primijetimo da je maksimalna količina vode koja se može zadržati iznad stuba na poziciji i određena maksimalnom visinom s lijeve strane, maksimalnom visinom s desne strane, i visinom samog stuba.

Ako označimo:

- \max_l — najveća visina stuba lijevo od pozicije i (uključujući i),
- \max_r — najveća visina stuba desno od pozicije i (uključujući i),
- h — visinu stuba na poziciji i ,

tada je količina vode iznad stuba i data formulom:

$$v_i = \max(0, \min(\max_l, \max_r) - h)$$

Dakle, potrebno je u linearnom vremenu izračunati za svaku poziciju najveću visinu lijevo i najveću visinu desno. Nakon što imamo te vrijednosti, dovoljno je za svaku poziciju izračunati pripadnu količinu vode koristeći gornju formulu i sabrati ih u ukupni rezultat.

UPUTSTVO

Ovo takmičenje sastoji se od rješavanja **3 problemska zadatka** u vremenu od **4 sata** (240 minuta). Zadatke je potrebno rješavati u jednom od sljedećih programskih jezika: Pascal, C, C++ ili Java. Takmičari koji koriste Pascal moraju programirati u programskom alatu **FreePascal**. Takmičari u C-u i C++-u moraju koristiti programski alat **CodeBlocks**. Za programski jezik Java predviđena je upotreba platforme **Eclipse**. Dozvoljeno je koristiti editor po izboru i pomoću navedenih alata prevoditi izvorni kod u izvršnu datoteku.

Tokom takmičenja **ne smijete komunicirati** ni sa jednom osobom, osim dežurne osobe takmičenja. To znači da morate **raditi samostalno** i **ne smijete koristiti Internet**. Takođe, zabranjena je upotreba bilo kakvih ranije napisanih programa ili dijelova programa.

Po isteku vremena predviđenog za takmičenje, na desktopu u folderu sa imenom **Takmicenje2026** moraju se nalaziti datoteke sa snimljenim izvornim kôdovima rješenja. Vaša rješenja možete upload-ovati na sistem za takmičenje, ako je dostupan, i provjeriti koliko bodova ste dobili. Nakon takmičenja, komisija će testirati vaša rješenja još jednom. Na kraju svakog zadatka dati su primjeri test podataka. Ti primjeri služe da bi vam tekst zadatka bio što je moguće jasniji i za provjeru formata ulaza i izlaza, a ne služe za provjeru ispravnosti vašeg programa. Ako vaš program radi na tim primjerima, to **nije garancija** da će raditi na službenim podacima za testiranje.

Ukupan broj bodova na nekom zadatku jednak je zbiru bodova test podataka koji se poklapaju sa službenim rješenjem. Ukupan broj bodova jednak je zbiru bodova na svim zadacima.

Sve informacije o zadacima (ime zadatka, vremensko i memorijsko ograničenje, način bodovanja) možete naći na uvodnoj stranici s naslovom Zadaci. Ako vam nije jasno nešto u vezi načina organizacije ovog takmičenja, odmah postavite pitanje dežurnom da vam to razjasni. Tokom cijelog takmičenja možete postavljati pitanja dežurnom u vezi zadatka. Dozvoljena su pitanja **koja razjašnjavaju nejasnoće u tekstu zadatka**. Ne smijete postavljati pitanja u vezi rješavanja zadatka. Prije nego postavite pitanje, pročitajte još jednom zadatak, jer je moguće da ste u prethodnom čitanju preskočili dio teksta zadatka.

Nepoštovanje ovih pravila ili nepridržavanje formata izlaznih podataka rezultiraće nepovratnim gubitkom bodova. Nemojte štampati ništa što se u zadatku ne traži, kao npr. poruke tipa 'Rjesenje je:' ili 'Unesite brojeve' i slično!

Srećno i uspješno takmičenje!

VAŽNO za jezik Java!

Ne kreirajte pakete za vaše zadatke, već koristite podrazumijevani (default) paket. Klase treba da budu imenovani tačno kako je navedeno u tabeli sa detaljima implementacije.

SREĆAN RAD!