**Olimpijada znanja** **2025.**

**Takmičenje iz Programiranja za **srednju** školu**

**16. 5. 2024.**

**Uputstva takmičarima**

Ovo takmičenje sastoji se od rješavanja **3 problemska zadatka** u vremenu od 3 **sata** (180 minuta). Zadatke je potrebno rješavati u jednom od sljedećih programskih jezika: C, C++ ili Java. 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 ***Takmicenje2025*** 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 zadataka 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. Takmičari mogu svaki zadatak

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 zadataka. Dozvoljena su pitanja **koja razjašnjavaju nejasnoće u tekstu zadatka**. Ne smijete postavljati pitanja u vezi rješavanja zadataka. Prije nego postavite pitanje, pročitajte još jednom zadatak, jer je moguće da ste u prethodnom čitanju preskočili dio teksta zadatka.

**Nemojte štampati ništa što se u zadatku ne traži**, kao npr. poruke tipa ‘Rjesenje je:’ ili ‘Unesite brojeve’ i slično!

**Nepoštovanje ovih pravila ili nepridržavanje formata izlaznih podataka rezultiraće nepovratnim gubitkom bodova.**

Srećno i uspješno takmičenje!

**VAŽNO za jezik Java!**

Ne kreirajte pakete za vaše zadatke, već koristite podrazumijevani (default) paket.

**Zadaci**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zadatak** | **Zadatak1 (A)** | **Zadatak2 (B)** | **Zadatak3 (C)** |
| **Izvorni kôd** | A.java  A.pas  A.c  A.cpp | B.java  B.pas  B.c  B.cpp | C.java  C.pas  C.c  C.cpp |
| **Memorijsko ograničenje** | 256MB | 256MB | 256MB |
| **Vremensko ograničenje**  **(po test podatku)** | 1 sekunde | 1 sekunda | 2 sekunda |
| **Ukupno bodova** | **35** | **35** | **30** |

**A: Kampanja**

Gospodin Kraba se sprema nove izbore za gradonačelnika Koralova. Kao dio svoje predizborne kampanje izjavio je da će ponovo učiniti Koralovo čistim.

Do početka izbora nema mnogo vremena, pa je Sunđer Bob, PR‑menadžer Gospodina Krabe odlučio da očisti samo đubre koje se nalazi u glavnoj ulici grada. Naime, u glavnoj ulici postoji **n** prljavih mjesta, označenih brojevima od 1 do **n** sa zapada na istok. Sunđer Bob želi da se đubre sa **i**‑tog mjesta ukloni tokom **i**‑tog dana.

Kao pravi patriota Koralova i jedina osoba u timu Gospodina Krabe koja dobro barata aritmetikom, zamoljeni ste da izračunate minimalan broj radnika koji treba da učestvuju u čišćenju. Taj broj se određuje prije početka procesa i ostaje isti svih **n** dana.

Nakon razgovora sa stručnjakom za čišćenje ustanovili ste da jedan radnik tokom dana **i** može da obavi jednu od tri akcije:

1. ukloni 1 kilogram đubreta na mjestu **i**;
2. premjesti tačno 2 kilograma đubreta sa mjesta **i** na mjesto **i + 1** (naravno, ovo nije moguće ako je **i = n**);
3. ne radi ništa.

Ulica se smatra čistom ako na kraju **n**‑tog dana nema đubreta ni na jednom mjestu. Dakle, Sunđer Bob mora da angažuje više ljudi kako bi mogli da rade istovremeno.

Pomozite Sunđer Bobu da izračuna minimalan broj ljudi potrebnih za čišćenje.

### Ulaz:

Prva linija sadrži jedan cio broj **n** (1 ≤ n ≤ 2·105) — broj prljavih mjesta u glavnoj ulici.

Druga linija sadrži **n** cijelih brojeva **t₁, t₂, …, tₙ** (0 ≤ tᵢ ≤ 2·105) — količinu đubreta na **i**‑tom mjestu, izraženu u kilogramima.

### Izlaz:

Ispišite jedan cio broj — minimalan broj radnika koji treba angažovati.

### Primjeri:

|  |  |
| --- | --- |
| **Ulaz:**  3  5 1 2  **Izlaz:**  4 | **Ulaz:**  4  8 1 2 7  **Izlaz:**  7 |

U **prvom primjeru** Sunđer Bob može da očisti ulicu angažovanjem **4 radnika**:

* **Prvi dan**  –  troje radnika uklanja po 1 kg đubreta (ukupno 3 kg), dok jedan radnik premjesti 2 kg na sledeće prljavo mjesto.
* **Drugi dan**  –  na mjestu 2 ima 3 kg đubreta; sva četvorica radnika su tu: trojica uklanjaju, jedan ne radi ništa.
* **Treći dan**  –  četvoro radnika uklanjaju preostala 2 kg đubreta.

Sa tri radnika nije moguće očistiti ulicu jer biste morali prvo da premjestite 4 kg sa mjesta 1 na mjesto 2. Zatim bi morali drugog dana da premjestite 4 kg sa mesta 2 na mjesto 3. Nа kraju biste imali 6 kg đubreta na trećem (poslednjem) mjestu, а 3 radnika ne mogu da uklone svih 6 kg u jednom danu.

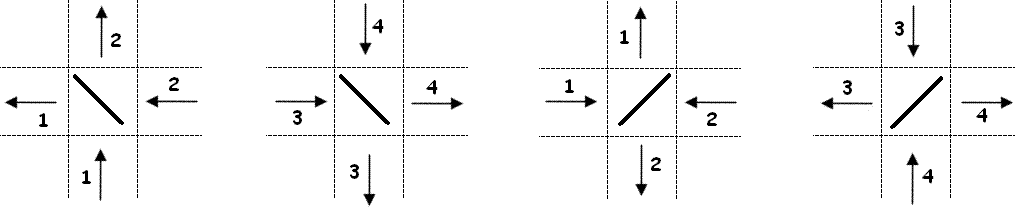
U **drugom primeru** odgovor ne može biti manji od **7**, jer се na poslednjem mjestu nalazi 7 kg đubreta, pa Sunđer Bob mora da angažuje bar 7 radnika da ga uklone.

**B: Voyager**

Voyager 1 je svemirska sonda lansirana davne 1977. godine i u ovom trenutku polako napušta Sunčev sistem. Putujući dalje kroz svemir, u svakom sistemu planeta na koji naiđe sonda treba da ostavi poruku u obliku radio signala koja će što duže svjedočiti o njenom prolasku.

Pretpostavimo da se planetarni sistem može prikazati u obliku pravougaone mreže sa **N** **redova** i **M** **kolona,** koje prostor dijele na **N** **puta** **M** **jednakih** **polja**. U jednom polju može se nalaziti **planeta**, **crna** **rupa** ili **prazan** **prostor**. Sonda je smještena u unaprijed zadatom **praznom** **polju** i iz njega treba u jednom od četiri moguća smjera (“U”-gore, “R”-desno, “D”-dolje, “L”-lijevo) odaslati radio signal.

Signal se nakon slanja bez prekida kreće pravolinijski kroz vrstu ili kolonu sve dok ne stigne do neke planete, gdje skreće za ugao od 90 stepeni i nastavlja dalje u odgovarajućem smjeru. Postoje dvije vrste planeta koje ćemo označiti znakovima „/“ i „\“. Način odbijanja signala prikazan je na slici:



Signal izlazi iz sistema kad se nađe na polju u kome je crna rupa ili kada izađe van okvira zadate pravougaone mreže. Poznato je da signalu za prelaz iz jednog u drugo polje treba jedna sekunda.

Napišite program koji će odrediti i ispisati smjer u kojem je potrebno odaslati radio signal da bi se on u sistemu zadržao **što** **više** **vremena**, kao i traženo vrijeme izraženo u sekundama. Ako signal može ostati zauvijek zarobljen u sistemu, umjesto traženog vremena ispišite poruku „Voyager“.

**Ulaz:**

U prvoj vrsti nalaze se prirodni brojevi **N** (1 ≤ **N** ≤ 500) i **M** (1 ≤ **M** ≤ 500).

U sljedećih **N** vrsta nalazi se po **M** karaktera (“/” ili “\” ili “C” ili “.”) pri čemu su “/” i “\” oznake planeta, “C” oznaka crne rupe, a “.” oznaka praznog prostora.

U posljednjoj vrsti nalaze se prirodni brojevi **SR** (1 ≤ **SR** ≤ **N**), broj vrste, i **SS** (1 ≤ **SS** ≤ **M**), broj kolone u kojoj je smještena sonda.

**Izlaz:**

U prvoj izlaza vrsti ispišite oznaku smjera (“U”, “R”, “D” ili “L”). U slučaju da postoji više validnih rješenja, prioritet pri odabiru ima “U”, pa “R”, pa “D” te na kraju “L”. U drugoj vrsti izlaza ispišite traženo vrijeme ili poruku "Voyager", zavisno od rješenja.

**Podzadaci:**

U test podacima ukupno vrijednim 50% bodova, poruka neće moći ostati zauvijek zarobljena u sistemu.

**Primjeri:**

|  |  |  |
| --- | --- | --- |
| **Ulaz:**  5 5  ../.\  .....  .C...  ...C.  \.../  3 3  **Izlaz:**  U  17 | **Ulaz:**  5 5  ....\  \..\.  ./\..  \../C  .\../  1 1  **Izlaz:**  D  12 | **Ulaz:**  5 7  /.....\  ../..\.  \...../  /.....\  \.\.../  3 3  **Izlaz:**  R  Voyager |

**Pojašnjenje** **prvog** **primjera:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| početak | smjer 'U' | smjer 'R' | smjer 'D' | smjer 'L' |
| ../.\  .....  .CP..  ...C.  \.../ | \*.\*\*\*  \*.\*.\*  \*C..\*  \*..C\*  \*\*\*\*\* | ../.\  .....  .C.\*\*  ...C.  \.../ | ../.\  .....  .\*...  ...C.  \.../ | ../.\  .....  .\*...  ...C.  \.../ |
|  | 17 sekundi | 3 sekunde | 3 sekunde | 1 sekunda |

**Napomena:** Potrebno je poslati jedan fajl sa imenom B.c/B.cpp/B.java.

**Napomena:** Naziv klase u java fajlu mora takođe biti B.

**C. Kornjače**

Dvije kornjače, Mila i Boki, nalaze se u datom neusmjerenom grafu. Mila želi da dođe do kupusa, a Boki do šargarepe. Međutim, postoji ograničenje – ne smiju se u istom trenutku naći na istom čvoru ili istoj grani grafa. U jednom koraku, kornjače se mogu pomjeriti na jedan od susjednih čvorova ili ostati u mjestu. Napišite program koji provjerava da li obje kornjače mogu doći do svoje hrane, bez međusobnog konflikta, i ispišite njihove putanje.

**Ulaz:** U prvom redu ulaza dati su brojevi N - broj čvorova u grafu, M - broj grana u grafu, A - čvor u kojem se nalazi Mila, B - čvor u kojem se nalazi Boki, C - čvor u kojem se nalazi kupus i D - čvor u kojem se nalazi šargarepa. Zatim su u sljedećih M redova date grane grafa.

**Izlaz:** U prvom redu izlaza  štatmpati ”DA” ukoliko je moguće pronaći putanje, a ”NE” u suprotnom. Ukoliko putanje postoje, u drugom redu izlaza štampati redosljed čvorova za Milu, a u trećem redu redosljed čvorova za Bokija. Ukoliko postoji više rješenja, štampati bilo koje od njih.

**Primjeri:**

|  |  |  |
| --- | --- | --- |
| **Ulaz:**  6 6 0 2 5 3  0 1  0 2  1 3  2 4  3 5  4 5  **Izlaz:**  DA  0 1 3 5  2 0 1 3 | **Ulaz:**  3 2 0 2 2 0  0 1  1 2  **Izlaz:**  NE | **Ulaz:**  5 5 0 1 4 3  0 1  1 2  1 3  2 3  3 4  **Izlaz:**  DA  0 1 3 4  1 2 2 3 |

**Podzadaci:**

Za 50% poena dovoljno je odštampati odgovor DA ili NE.

**Napomena:** Potrebno je poslati jedan fajl sa imenom C.c/C.cpp/C.java.

**Napomena:** Naziv klase u java fajlu mora takođe biti C.