

Školsko takmičenje iz informatike

Prva bošnjačka gimnazija Sarajevo

27.03.2021.

Propozicije takmičenja

1. Takmičenje počinje u 10:00. Za takmičenje je predviđeno 2 sata za izradu zadataka, te 10 minuta za slanje zadataka nakon kraja takmičenja.
2. Nije dozvoljeno koristiti druga elektronička sredstva poput mobitela, digitrona itd.
3. Zabranjena je upotreba USB stickova, CD-ova itd.
4. U slučaju da dva učenika imaju isti broj bodova, prednost se daje učeniku koji je završio i predao zadatke ranije.
5. Pitanja je dozvoljeno postavljati samo u prvih 30 minuta takmičenja.
6. Zadaci se rade isključivo u CodeBlocks-u
7. Kamera mora biti uključena tokom čitavog trajanja takmičenja. Mobilni će se koristiti kao druga kamera koja mora biti okrenuta da se vidi ekran, stol i tastatura. Mikrofon treba biti ugašen osim prilikom postavljanja pitanja.
8. Zadaci se šalju svi zajedno pri završetku na mail: erol.terovic1@bosnjackagim.edu.ba obratite pažnju, da se samo jednom smiju poslati zadaci, prepravke nisu dozvoljene.
9. Zadaci se šalju kao odvojeni .cpp file-ovi koji su preimenovani u format:
Z#_Ime_Prezime.cpp gdje # predstavlja broj zadatka.
10. Prepisivanje će rezultirati diskvalifikacijom s takmičenja.
11. Nije dozvoljeno gledati u programe osim CodeBlocks-a, ove postavke takmičenja, a Internet pretraživač se smije isključivo koristiti za unmuting prilikom postavljanja pitanja. Smijete otvoriti folder zadatka za potrebe testiranja zadatka ako trebate koristiti input ili output file.

O rezultatima ćete biti obaviješteni putem školskog e-maila. Sretno Vam Svima ☺.

Zadatak 1

Lijeva rotacija elemenata niza prirodnih brojeva s veličinom **n** se vrši tako što se svaki elemenata niza pomjeri za **1** u lijevo. Ako je dat prirodni broj **d**, treba rotirati niz za d elemenata u lijevo i ispisati rezultat. Također specificira se u inputu string „**LIJEVO**“ ili „**DESNO**“ koji odrede smjer rotacije.

Primjer:

smjer=“LIJEVO“

d=2

[1,2,3,4,5] nakon 2 rotacije postaje [3,4,5,1,2].

Format ulaza i izlaza:

Input se čita iz datoteke **input.txt** a output se upisuje u datoteku **output.txt**.

Prvi red sadrži dva prirodna broja koji predstavljaju broj elemenata niza n, te broj rotacija d.

Drugi red sadrži string koji predstavlja smjer rotacije.

Treći red predstavlja n razmaknutih prirodnih brojeva koji predstavljaju elemente niza.

Unos se ponavlja do kraja dokumenta, tj. može biti više setova ulaza u jednom dokumentu.

Primjer ulaza i izlaza 1

INPUT	OUTPUT
5 4 LIJEVO 1 2 3 4 5	5 1 2 3 4

Primjer ulaza i izlaza 2

INPUT	OUTPUT
5 4 LIJEVO 1 2 3 4 5 10 1 DESNO 10 20 30 40 50 60 70 80 90 100 5 3 LIJEVO 5 4 3 2 1	5 1 2 3 4 100 10 20 30 40 50 60 70 80 90 2 1 5 4 3

Objašnjenje: da bi se desile 4 rotacije u lijevo, niz se mijenja ovako:

[1,2,3,4,5]->[2,3,4,5,1]->[3,4,5,1,2]->[4,5,1,2,3]->[5,1,2,3,4]

Zadatak 2

Tea je pristupila izvjesnoj društvenoj mreži kako bi ostala u kontaktu sa svojim prijateljima. Kad je pravila račun, trebala je da izabere korisničko ime i password. Password mora biti jak. Web stranica smatra izvjesni password jakim ako zadovoljava sljedeći kriterij:

- Dužina mu je najmanje 6 karaktera
- Sadrži barem jednu cifru
- Sadrži barem jedno malo slovo engleske abecede
- Sadrži barem jedno veliko slovo engleske abecede
- Sadrži jedan specijalni karakter (Jedan od ovih: ! @ # \$ % ^ & * () - +)

Ukucala je random string dužine **n** u polje za password ali nije bila sigurna je li jak password ili ne. Ako znamo string koji je ukucala, treba pronaći minimalni broj karaktera što mora dodati da bi njen password postao jak. Treba napisati **funkciju** `minimumNumber` koja **prima** string i int, koji predstavljaju password i dužinu passworda koji se testira, a **vraća** minimalni broj karaktera koje moramo dodati.

Primjer:

Password="2bbbb"

Ovaj password ima 5 karaktera i fali mu veliko slovo i specijalni karakter. Minimalni broj karakter koji se moraju dodati je 2.

Password="2bb#A"

Ovaj password ima 5 karaktera i sadrži barem po jedan od svih tipova karaktera. Samo se treba dodati 1 karakter.

Format ulaza i izlaza:

Prva linija predstavlja **integer n**, dužinu passworda, druga linija sadrži **string password**. Zagarantovano je da će password sadržavati samo karaktere koji su gore spomenuti, tj [a-z], [A-Z], [0-9], ili [!@#%&*()-+].

Primjer ulaza i izlaza 1

INPUT	OUTPUT
3 Er1	3

Objašnjenje 1: Može ga napraviti jakim tako što doda 3 karaktera, kao npr #ol pa će password postati Er1#ol što je jak password.

Primjer ulaza i izlaza 2

INPUT	OUTPUT
11 #HackerRank	1

Objašnjenje 2: Dovoljno je dug, ali fali mu broj, pa je dovoljno samo jedan broj dodati da postane jak.

Zadatak 3

Ako dobijemo kvadratnu matricu (jednako redova i kolona), treba izračunati vrijednost razlike sume prostih brojeva koji se nalaze u njene dvije dijagonale (glavna i sporedna). Ako element dijagonale nije prost broj, onda se taj broj ne treba uzimati u sumu elemenata dijagonale. Broj je prost ako i samo ako je djeljiv samo s 1 i samim sobom, pri tome mora biti veći od 1.

Primjer:

Za matricu :

2 2 3

4 3 6

9 8 5

Suma prostih brojeva glavne dijagonale je: $2+3+5=10$, suma prostih brojeva sporedne je $3+3=6$ (9 se ne pika jer nije prost broj). Razlika je $10-6=4$

Treba napisati funkciju `diagonalDifference` koja prima matricu predstavljenu kao vektor vektora, a vraća vrijednost razlike.

Format ulaza i izlaza:

Prva linija predstavlja int n, broj redova i kolona matrice, svaki od narednih n unosa predstavlja red matrice.

Primjer ulaza i izlaza 1

INPUT	OUTPUT
3 11 2 4 4 5 6 10 8 -12	11

Objašnjenje: Glavna dijagonala je:

11

5

-11

Suma prostih elemenata ove dijagonale je: $11+5=16$

Sporedna je:

4

5

10

Suma prostih elemenata ove dijagonale je: 5

Pa razlika te dvije sume je $16-5=11$ i to je rezultat.

Zadatak 4

Mark ima 2D ploču A veličine $H \times W$ sa H redova i W kolona. Ploča je podijeljena u ćelije veličine 1×1 , i svaka ćelija ima koordinate (i,j) . U svakoj ćeliji se nalazi neki prirodni broj. Mark za svaku ćeliju uzme njen broj, i toliko kocki dimenzija $1 \times 1 \times 1$ poreda na tu ćeliju. Ako znamo izgled ploče i brojeve koji su povezani sa svim ćelijama, pronaći površinu 3D objekta proizvedenog od kocki.

Format ulaza i izlaza:

Prva linija se sastoji od H i W , predstavljaju visinu i širinu ploče. Slijedećih H linija sadrži po W prirodnih brojeva.

Površinu objekta treba ispisati.

Ograničenja:

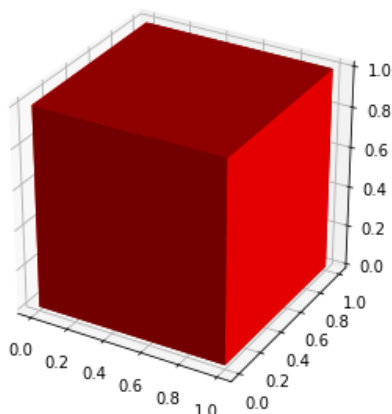
$1 \leq H, W \leq 100$

$1 \leq \text{brojevi} \leq 100$

Primjer ulaza i izlaza 1

INPUT	OUTPUT
1 1 1	6

Objašnjenje 1: Ovaj slučaj je trivijalan, ploča je 1×1 i ima vrijednost 1 u jedinoj ćeliji. Očigledno je površina objekta $1 \times 1 \times 1 = 6$



Primjer ulaza i izlaza 2

INPUT	OUTPUT
3 3 1 3 4 2 2 3 1 2 4	60

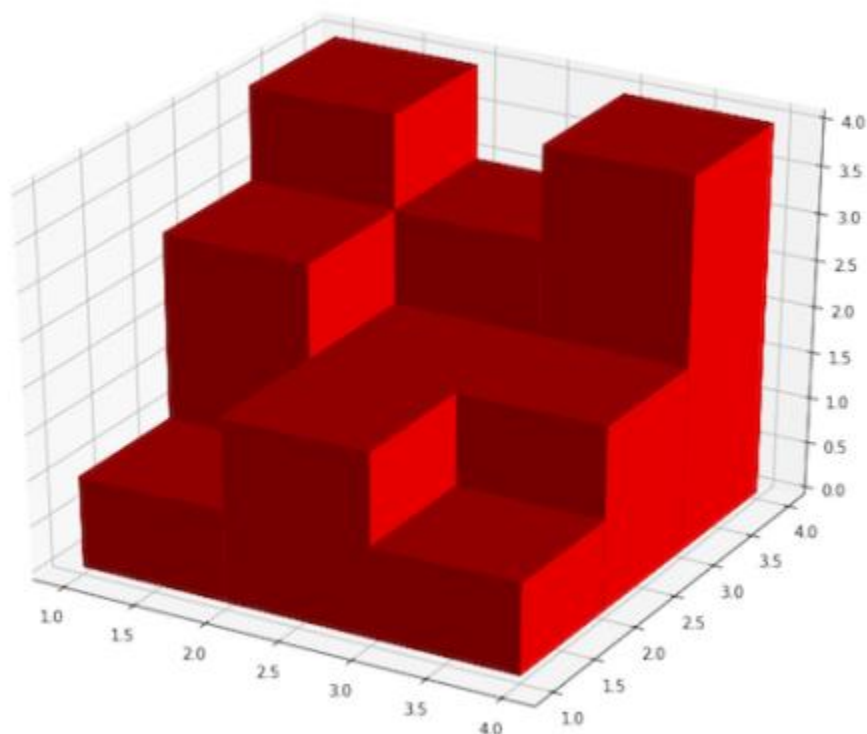
Objašnjenje 2: Dimenzije ploče su 3x3. Vrijednosti u ćelijama su

1 3 4

2 2 3

1 2 4

I ove vrijednosti predstavljaju broj naslaganih kocki dimenzija 1x1x1 u respektivnim poljima ploče (Ploča je prikazana horizontalno na ovoj slici, a kocke su naslagane na njoj). Kad se izračuna površina dobije se 60.



(Na ovoj slici bi trebale koordinate ići od 0 do 3 a ne od 1 do 4 po H i W osama, ali to ne mijenja suštinu zadatka)