## 

**Izbor i prevod zadataka za tekmičenje:**

**Državni nivo takmičenja – zadaci i rešenja**

**Dabar 2016/2017.**

Milan Lukić

Bojan Milosavljević

Svetlana Jakšić

Milan Rajković

**Programski odbor takmičenja:**

Milan Rajković (predsednik programskog odbora)

Milan Lukić (član programskog odbora)

Svetlana Jakšić (član programskog odbora)

Bojan Milosavljević (član programskog odbora)

Saša Jevtić (član programskog odbora)

Dragan Krstić (član programskog odbora)

**Tehnična podrška:** Milutin Spasić i Branko Dolić

## O takmičenju i priručniku

Draga deco i poštovane kolege,

Završili smo 4. po redu, takmičarsku Dabar godinu. Opet smo zajedno pomerili granice i još malo porasli! Ove godine Međunarodnom takmičenju Dabar uključilo je oko 36.000 učenika svih razreda osnovne i srednje škole, iz cele Srbije.

Informatičke izazove na koje ste nailazili rešavajući zadatke tokom državnog nivoa takmičenja, sabrali smo i objavili u ovom priručniku. Želja nam je da ovim podstaknemo dodatno interesovanje nastavnika i učenika u bavljenju temama i intelektualnim problemima koji su predstavljeni kroz zadatke.

Priručnik će učenicima i nastavnicima biti materijal za učenje i pripremu za buduća takmičenja, ali istovremeno i deo riznice Dabar intelektualnih problema, koja iz godine u godinu postaje sve bogatija. Pripručnik su pripremili organizatori takmičenja, a zadovljstvo koje ste vi pokazali rešavajući zadatke učinilo je da možemo biti ponosni na odabir zadatak;).

U takmičenje je uloženo mnogo rada i energije, tako da nas posebno raduje što takmičenje postaje sve masovnije i popularnije.

Pozivamo vas da pratite naše aktivnosti i da sa zajedno sa nama radujete novim zadacima.

Želimo vam da uživate u rešavanju zadataka!

Programski odbor takmičenja

Sadržaj

[O takmičenju i priručniku 3](#_Toc475906486)

[2016-AT-05 Pertle 6](#_Toc475906487)

[2016-CZ-05 Abecedna imena 9](#_Toc475906488)

[2016-LT-02-eng.odtSplavarenje 11](#_Toc475906489)

[2016-RU-07-eng – Tri u redu 12](#_Toc475906490)

[2016-SK-03 – Lanci 13](#_Toc475906491)

[2016-RU-08-1-eng – Sebične veverice 15](#_Toc475906492)

[2016-LT-06-eng Najkraći put 16](#_Toc475906493)

[2016-CH-23 - Igra 17](#_Toc475906494)

[2016-BE-03 - Prekidači 18](#_Toc475906495)

[2016-PK-01A-en B-Enigma 20](#_Toc475906496)

[2016 – CA-02 - Majmuni koji skaču 22](#_Toc475906497)

[2016 – AU -06 – Rekurzivno crtanje 23](#_Toc475906498)

[2016-CA-08a- MapReduce 25](#_Toc475906499)

[2016-NL-05-EN Funkcija TotalSwap(n,s) 26](#_Toc475906500)

[2012-LT-08 - Magično drvo 28](#_Toc475906501)

[2013–RU-08-EN Masa i visina 29](#_Toc475906502)

[2013–RU-07-EN Drevno dabarsko pismo 31](#_Toc475906503)

[2016-CA-09-Application 33](#_Toc475906504)

[2016-IE-04-eng Pozdravljanje timova 35](#_Toc475906505)

[Lavirint 37](#_Toc475906506)

[Tunel 37](#_Toc475906507)

[2016-BE-02 Pronađi lopova 38](#_Toc475906508)

[2016-CH-03-eng Zdravstveni centar 39](#_Toc475906509)

[2016-SK-07- Igranje 41](#_Toc475906510)

[Rođendanska torta 42](#_Toc475906511)

[Tajni recept 43](#_Toc475906512)

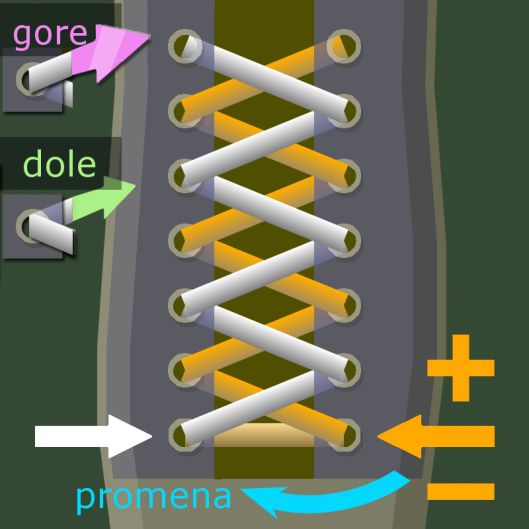
[Ključevi 44](#_Toc475906513)

[Planiranje putovanja 46](#_Toc475906514)

## 2016-AT-05 Pertle

Dabar obožava neobično vezivanje cipela. On želi robota koji će mu vezivati pertle na cipelama, ali se zapita: „Kako bih rekao robotu na koji način želim da mi se vežu cipele? Potrebna mi je neka vrsta programskog jezika. Kako bi izgledao taj jezik?”

Razmotrite tradicionalni način vezivanja pertli kao na sledećoj slici.



Pretpostavimo da je pertla koja počinje na desnoj strani uvek narandžasta, a da je pertla koja počinje na levoj strani uvek bela. Dabar je predložio sledeći program za ovaj način vezivanja pertli:

narandžasta: gore

bela: gore

{narandžasta: + promena gore

bela: + promena gore}

Pojašnjenje:

{…} sve unutar zagrada biće ponovljeno koliko je puta potrebno

3 {…} kao iznad, ali ponovljeno tačno 3 puta, za druge brojeve važi taj broj ponavljanja

narandžasta: sledeće komande primeniće se samo na narandžastu pertlu

bela: sledeće komande primeniće se samo na belu pertlu

gore provlači pertlu kroz rupicu na cipeli u pravcu nagore na poziciju narandžaste ili bele strelice, prikazanih na slici iznad na svojim početnim položajima

dole isto kao prethodno, samo u pravcu nadole

+ pomera narandžastu ili belu strelicu jednu poziciju napred (na sledeću rupicu za pertlu)

- pomera strelicu za jednu rupicu unazad

promena narandžasta ili bela strelica se menja od sdesna na ulevo, ili od sleva na udesno (u zavisnosti od njene trenutne pozicije)

**Pitanje:**

Koja se šara od pertli kreira pomoću sledećeg programa?

narandžasta: gore

bela: gore

2 {

narandžasta: + promena gore

bela: + promena gore

}

narandžasta: + dole

bela: + dole

{

narandžasta: + promena gore

bela: + promena gore

}

Savet: Usredsredite se na jednu (bilo koju) pertlu.



**Objašnjenje:**

Tačan odgovor je “A”. Za dva para rupica program radi kao jednostavan program prikazan u postavci zadatka. Onda se pertle postavljaju kroz rupice ponovo, ali se strane nisu promenile. Pertle se onda provlače dole kroz rupice, po jednu na svakoj strani i onda se strane promene. Program se onda nastavlja kao pre.

**Informatička pozadina:**

Ovaj jednostavan programski jezik poseduje neke elemente tradicionalnih programskih jezika. Ima naredbe i petlje i omogućava neke ugrađene promenljive (npr. pozicije strelica).

## 2016-CZ-05 Abecedna imena

Ime je abecedno ukoliko se ono može sastaviti od abecedno sortirane liste njegovih slova (poređanih po abecednom redosledu) praćenjem sledećeg algoritma:

Kreiraj abecedno sortiranu listu slova.

Počni sa prvim slovom u listi.

Uzmi sledeće slovo iz liste i dodaj ga levo ili desno u imenu.

Nastavi tako dok se ne isprazni lista.

Ime HENRY je abecedno: njegova abecedna lista slova je EHNRY i možemo sastaviti ime sledećim redosledom akcija po algoritmu: E -> HE -> HEN -> HENR -> HENRY

Ime LUCY nije abecedno: njegova abecedna lista slova je CLUY. Nije moguće dodati drugo slovo sa liste L prvom slovu C jer slova C i L nisu susedna u reči LUCY.

**Pitanje:**

Koje od ovih imena je abecedno?

**Odgovor:**

A) JOHNNY

B) SUSAN

C) ISIDOR

D) ROBERT

**Objašnjenje:**

Tačan odgovor je pod D). ROBERT je abecedno ime.

Možemo kreirati dijagram za svako ime (kao na slikama ispod). Slova iz reči su postavljena na horizontalnu osu po redosledu u kom se pojavljuju u imenu. Redosled slova u abecedi je nacrtan po vertikalnoj osi.

Ako dijagram ima samo jedan vrh **v**, ime je abecedno. Vrh pokazuje početno slovo u formiranju reči. Svako drugo upotrebljeno slovo mora biti kasnije u abecednom redosledu i zato tačka spojena sa ovim slovom mora biti postavljena više na dijagramu. Ako neki dijagram ima vrh nagore **^**, ime nije abecedno.

Dijagrami za ponuđene odgovore su sledeći:

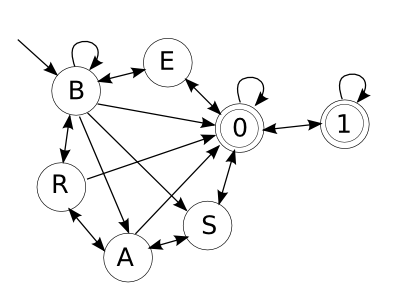
|  |  |
| --- | --- |
| 2014-CZ-05-expl4.gif | 2014-CZ-05-expl2.gif |
| 2014-CZ-05-expl3.gif | 2014-CZ-05-expl1.gif |

**Informatička pozadina:**

Zadatak se odnosi na algoritmizaciju kodiranja teksta, a takođe i na uprošćavanje rešavanja problema vizuelizacijom. Neka pravila koja se opisuju rečenicama ili opisima procedura zvuče složena, teška za razumevanje i proveru. Kada se grafički simbolizuju, često izgledaju jednostavnija. U našem zadatku, provera grafika je skoro trenutna, dok provera algoritma praćenjem opisa rečima oduzima mnogo više vremena.

## 2016-LT-02-Splavarenje

Dabrovi grade splavove i koriste ih u saobraćaju. Da bi se poštovala sva saobraćajna pravila, neophodno je i da svaki splav ima registracionu oznaku sa jedinstvenom tekstom. Registracioni tekst se sastoji od slova i brojeva i određuje se prema uputstvu: registracija uvek započinje slovom B a završava se ili brojem 0 ili brojem 1 (slika ispod).



Pitanje:Koja od navedenih registracija nije ispravna?

B00S01

BR0101

BSA011

BBA100

**Tačan odgovor:**

BBA100

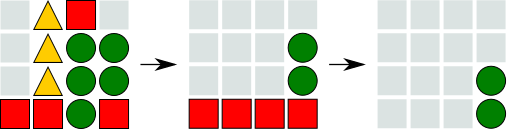
**Informatička pozadina:**

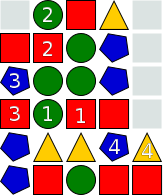
Konačni automat je važan deo informatičke nauke. Računari često čitaju niz znakova ili reči u dokumentu ili računarskom programu uz pomoć konačnih automata.

## 2016-RU-07 – Tri u redu

“Tri u redu” je popularna kompjuterska igrica gde igrači menjaju susedne parove kamenčića različitih oblika. Kada se tri ili više kamenčića istog oblika nađu jedan pored drugog u redu ili koloni (vertikalno ili horizontalno) oni nestanu, a ostali kamenčići padaju na niže. Nakon što preostali kamenčići padnu, proces se ponavlja.

Cilj igre je da svi kamenčići nestanu. Evo primera:





Pitanje: Koje kamenčiće treba igrač da zameni da bi svi nestali u primeru koji se nalazi na slici ispod?

A. kamenčići obeleženi brojem 1

B. kamenčići obeleženi brojem 2

C. kamenčići obeleženi brojem 3

D. kamenčići obeleženi brojem 4

**Tačan odgovor:**

kamenčići obeleženi brojem 2

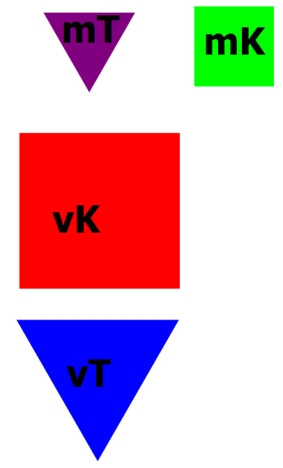
**Informatička pozadina**:

Odmah čim vide kod, programeri obično treba da shvate šta program radi i bez njegovog pokretanja. Problem u zadatku je isti – bez pokretanja igrice treba pronaći tačnu kombinaciju

## 2016-SK-03 – Lanci

Dabar Laki je programer i piše kompjuterske programe. On koristi naredbe da bi napisao program koji će mu nacrtati lančić sastavljen od trouglova i kvadrata.

Laki može koristiti sledeće naredbe:



vK – naredba koja crta veliki kvadrat

mK – naredba koja crta mali kvadrat

vT – naredba koja crta veliki trougao

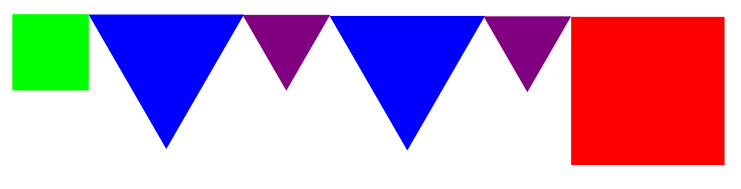
mT – naredba koja crta mali trougao

Ako želi da ponovi neke instrukcije dabar Laki može upisati sledeće:N[Inst]

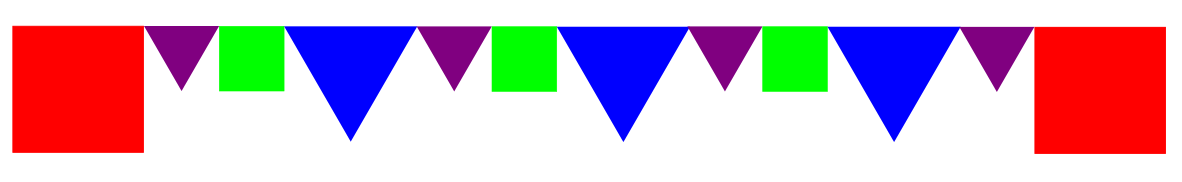
gde je N broj ponavljanja a Inst je set naredbi koje treba izvršiti

Kada Laki napiše N[Inst] to znači da će program N puta ponoviti set naredbi koje se nalaze unutar zagrada a koje su, u ovom slučaju, obeležene sa Inst.

Na primer, napisani program mK 2[vT mT] vK bi nacrtao ovakav lančić



Pitanje: Koji napisani program bi nacrtao lančić krikazan na slici ispod?



A) mK 3 [mT mK vT] mT mK

B) vK 3 [mT mK vT] vK

C) vK 3 [mT mK vT] mT vK

D) vK 2 [mT mK vT] mT vK

**Tačan odgovor:**

C) vK 3 [mT mK vT] mT vK

**Informatička pozadina:**

Program predstavlja niz uputstava. U ovom primeru, program izvršava uputstva u nizu „jedan po jedan“. Ako je program napisan tačno – računar radi ono što želite. Ukoliko program nije napisan tačno – računar radi ono što ste mu napisali ☺.

Set instrukcija u programu nazivamo blok naredbi. Ukoliko se blok naredbi ponavlja – to se obično naziva petlja.

## 2016-RU – Sebične veverice

Sebične veverice žive u rupama na drvetu. Na jednom drvetu postoji pet velikih rupa koje se nalaze jedna iznad druge. Na tom drvetu živi 16 sebičnih veverica i one su primorane da dele rupe.

Ali svakog dana sebične veverice proveravaju: broj trenutnih komšija, broj veverica koje žive u rupi ispod i u rupi iznad njihove u kojoj su trenutno. Ukoliko je u susednoj rupi manje veverica nego u njenoj, čim padne veče, sebične veverice se tajno pomeraju u tu rupu neznajući da tako rade sve sebične veverice iz svih rupa. Ako je broj veverica isti u susednim rupama iznad i ispod – veverica će ići u rupu iznad.

Na primer: ako je danas situacija u rupama sledeća: 5, 0, 0, 4, 7

5

0

0

4

7

Veverice sa vrha će u toku noći sići u rupu ispod (bolje je 0 nego 4 komšinice). Svih 7 veverica sa dna će preći u rupu iznad (bolje je 4 nego 6 komšinice) a sve 4 veverice iz pretposlednje rupe će se popeti u rupu iznad (bolje je 0 nego 3 komšinice). Tako da će situacija biti 0,5,4,7,0:

0

5

4

7

0

Pitanje: Ako je situacija u rupama skao na slici ispod, posle koliko dana će se sebične veverice naći u istoj rupi?

6

3

3

0

4

A. 3

B. 4

C. 6

D. Nikada

**Tačan odgovor:**

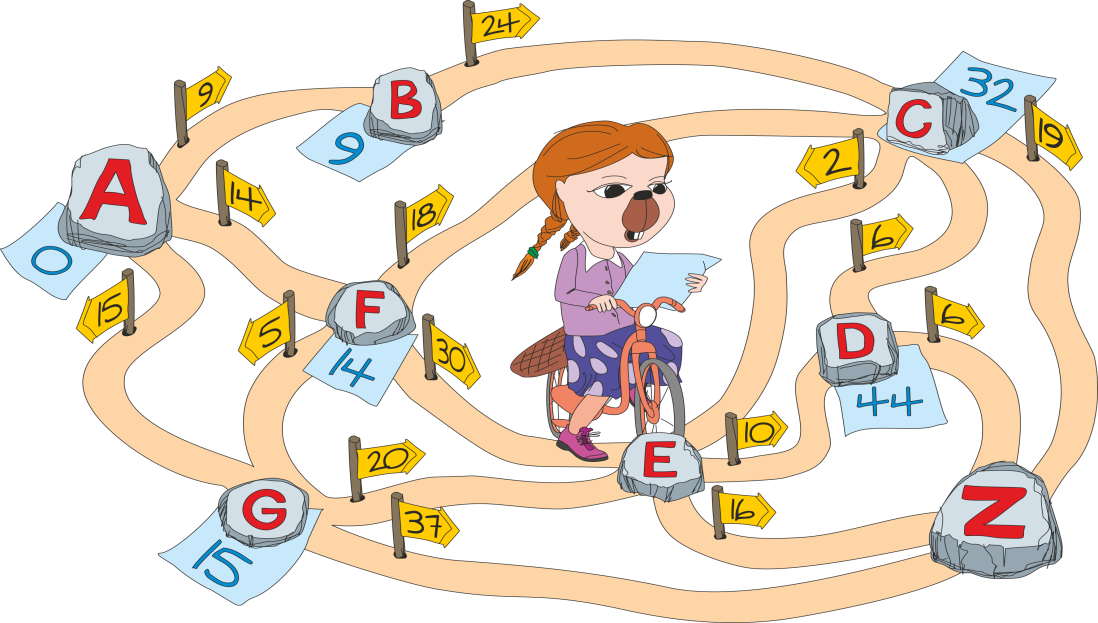
A. 3

**Informatička pozadina**:

U pitanju je inteligencija jata (kolektivna inteligencija -Swarm Intelligence). Inteligencija jata je novi koncept veštačke inteligencije. Priroda je najčešći izvor inspiracije inteligencije jata (mravi, pčele, jato ptica). Ove tehnike su pokazale izuzetne rezultate kod rešavanja problema koji su teško rešivi uobičajenim računarskim tehnikama. Najčešće korišćena i najviše proučavana je široko primenljiva „Optimizacija pomoću mravlje kolonije“. Kao rezultat njihove organizovanosti, mravlje kolonije mogu obavljati veoma složene zadatke, koje daleko prevazilaze sposobnosti mrava kao jedinke.

## 2016-LT- Najkraći put

Dabrica Bica vozi biciklu po svim biciklističkim stazama prikazanim na slici, tažeći najkraći put od tačke A do tačke Z. Ona na svakoj raskrsnici ostavlja jedno parče papira na kome napiše broj. Inače, svi putevi su jednosmerni.



Pitanje: Koji broj će dabrica Bica napisati na raskrsnici E?

a. 34

b. 35

c. 44

d. 30

**Tačan odgovor:**

34

**Informatička pozadina:**

Najkraća putanja je jedan od osnovnih informatičkih zadatka na svakodnevnim aplikacijama. Algoritmi najkraćeg puta se primenjuju tako da se automatski nađu pravci između fizičkih lokacija. kao što su vozni pravci na sajtovima veb mapiranja (npr. Google Maps). Dajkstrin algoritam je jedan od popularnih algoritama koji se koriste da za pronalazak najkraćeg puta.

## 2016-CH - Igra

Dabar Bane poziva prijatelja Saleta da igraju tzv. igru pomeranje. On objašnjava igru svom prijatelju: "Igru igraju dva igrača i započinje tako što je na stolu 13 kamenčića. Igrač br.1 počinje igru tako što može skloniti sa stola 1, 2 ili 3 kamenčića. Zatim je na redu Igrač br.2, koji takođe može skloniti 1, 2 ili 3 kamenčića. Onda ponovo igra Igrač br.1, onda Igrač br.2, i tako dalje. Igrač koji ukloni poslednji kamenčić dobija partiju."

Dabar Bane započinje igru.

Savet: Ako su ostavljena 4 kamenčića, Bane ne može da uzme poslednji par. On treba da izbegne ovu situaciju!

Pitanje: Koliko kamenčića dabar Bane treba da skloni u svom prvom pokušaju da bi sigurno pobedio?

A. Jedan kamenčić

B. Dva kamenčića

C. Tri kamenčića

D. Nebitno koliko – uvek pobeđuje

**Tačan odgovor:**

Jedan kamenčić

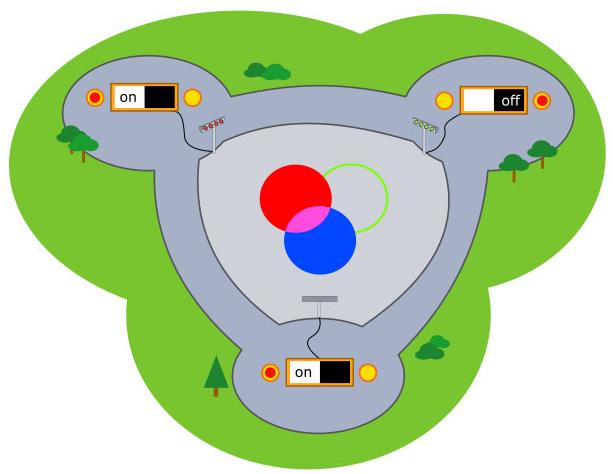
**Informatička pozadina:**

Ovo je klasičan slučaj strateške igre sa dva igrača sa naizmeničnim potezima. Nakon svakog obrta, računar analizira različite moguće pokrete i izračunava najbolju verovatnoću da se dobije igra. Zatim računar izvršava svoj potez i počinje sa ponovnom analizirom.

Postoji tačan algoritam za ovu igru. Za druge igre, koje nemaju tačan algoritam za rešavanje, ili kada je tačan algoritam usporen, primenjuju se heurističke metode - metode primena stečenog znanja. Godine 1997. prvi put u istoriji, šahovski prvak sveta je poražen od strane računara uz pomoć heuristike.

## 2016-BE - Prekidači

Dabar Džile je napravio veoma lep i veliki ekran u centru grada. Ekran može praviti razne boje i ima čak osam različitih izgleda. Izgled velikog ekrana zavisi od pozicije 3 prekidača (svaki prekidač ima poziciju „on“ što znači uključen i „off“ što znači isključen).



Džile je veliki stručnjak, majstor svog zanata, ali ipak želi da proveri da li ekran radi dobro u svih 8 različitih on/off kombinacija prekidača.

Na žalost, prekidači su međusobno udaljeni po 1km i Džile će morati da pešači kako bi isprobao sve kombinacije.

Pitanje: Koliko najmanje kilometara dabar Džile mora preći da bi isprobao svih 8 različitih on/off kombinacija?

Odgovor

a. 6 km

b. 8 km

c. 10 km

d. 12 km

**Tačan odgovor:**

6km

**Informatička pozadina:**

činjenica je da se svih 8 kombinacija prekidača mogu generisati u samo 7 koraka okretanjem jednog prekidača na svakom koraku. Ako napišem 0 za „off“ i 1 za „on“ sve kombinacije se mogu napisati i kao niz bitova. Naše rešenje odgovara sekvenci:

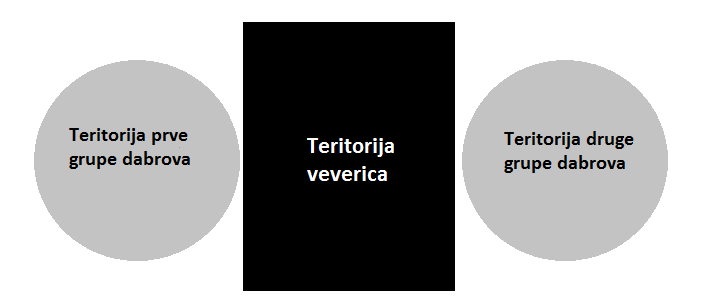
000 100 110 010 011 111 101 001

Ovakve sekvence se nazivaju i Grejov kod. Grejov kod dužine n>1 je niz k-torki bita (k>=1) b1, b2,...bn takvih da se u nizu b svaka dva uzastopna člana, kao i prvi i poslednji član razlikuju na tačno jednoj poziciji.

## 2016-PK--Enigma

Dabrovi igraju igru sa vevericama. Dabrovi su podeljeni u dve grupe i pokušavaju da prenesu poruke jedni dugima ali tako da veverice ne otkriju sadržinu poruke.

Dabrovi moraju da komuniciraju šifrovano, zato što poruke prolaze kroz zonu koju kontrolišu veverice.



Zato su dabrovi odlučili da koriste B-Enigmu, mašinu za šifrovanje poruka.

B-Enigma se sastoji od dva točkića na kojima se nalaze slova i radi na sledeći način:

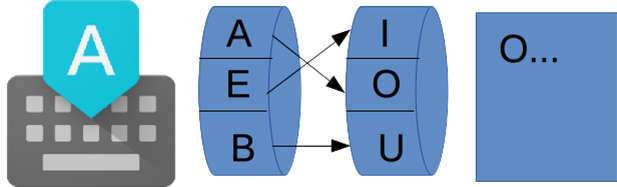
- šifruje se cela reč, slovo po slovo, počev od prvog

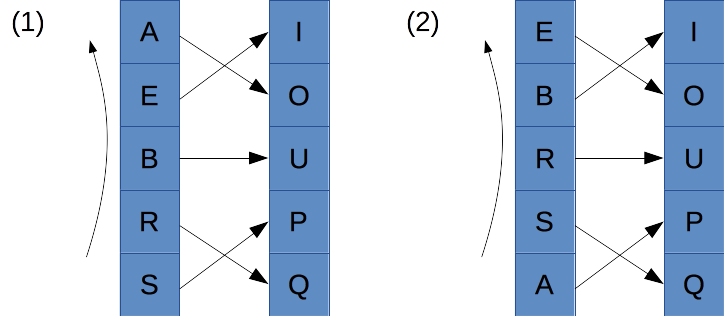
- kada dođe na red slovo koje treba šifrovati, to slovo se pronađe na levom točkiću i strelica ka desnom točkiću pokazuje na slovo u koje će biti šifrovano (u prvom koraku slovo A će biti šifrovano slovom O)

- posle kucanja jednog slova, levi točkić se pomera za jednu poziciju (za jedno slovo) u pravcu koji je prikazan strelicom i staje na poziciju (2) ali se strelice ne pomeraju

- desni točkić se nikada ne pomera

Na slici ispod prikazani su točkići sa pet slova.





Pitanje: Dabrovi iz prve grupe žele da pošalju poruku „BEBRAS“ dabrovima iz druge grupe. Kako će izgledati šifrovana poruka ako se krene od pozicije (1)?

Odgovor:

A. UOOAEB

B. UOUQOP

C. UOOOIP

D. UOOOPQ

**Tačan odgovor:**

UOOOIP

**Informatička pozadina:**

B-Enigma u zadatku je pojednostavljena verzija Enigma mašine, koja je korišćena tokom Drugog svetskog rata od strane nemačke vojske, za šifrovanje poruka. Tokom čitavog rada, britanci su pokušavali da dešifruju ovu mašinu. Prvi časovi kriptografije, u većini slučajeva, počinju sa prezentacijama o Enigmi. Sam rad na dešifrovanju Enigme doveo je do razvoja prvih računara.

## 2016 – CA - Majmuni koji skaču

Oko žbunastog drveta su dva listopadna drveta i dve palme.

Pet tipova banana (P, Q, R, S, T) se nalaze na drveću, različit tip banane za svako drvo. Majmun skače sa drveta na drvo, uživa jedući bananu i kada je pojede, onda skoči na sledeće drvo. Majmunu je potrebno:

tri sekunde da skoči sa žbunastog drveta na neko drugo drvo i da pojede bananu

dve sekunde da skoči sa listopadnog drveta na palmu ili obrnuto i da pojede bananu

sedam sekundi da skoči sa listopadnog na listopadno drvo (ili sa palme na palmu) usput izbegavajući žbunasto drvo i da pojede bananu

Majmun je skaao i pojeo banane tipa P, Q, S, R, T, R, P.

Pitanje: Koji tip banana je na žbunastom drvetu ako je ukupno vreme, dok majmun skače sa drveta na drvo i jede banane, najmanje moguće?

Odgovor:

P ili Q ili T

P ili S ili T

Q ili S ili T

Q ili R ili S

**Tačan odgovor:**

P ili S ili T

**Informatička pozadina:**

Pozadina ovog problema se veoma često koristi u informatici.Ovaj problem uključije pronalazak najboljeg, optimalnog, rešenja problema. Računari se često koriste kako bi se pronašle maksimalne ili minimalne vrednosti nekog merenja (granične vrednosti).

## 2016 – AU – Rekurzivno crtanje

Pod rekurzivnom metodom se podrazumeva metod koji poziva sam sebe tj. ukoliko neki postupak zahteva da delovi problema, koje je razdvojio od drugih, bivaju nezavisno podvrgnuti istom tom postupku, taj postupak je rekurzivan.

U realnom svetu primer bi bio kada se dva ogledala postave jedno ka drugom. Dobijeni lik u ogledalu se ogleda rekurzivno.

Evo primera rekurzije:

Naredba CrtajKvadrat (x, y, s) nalaže računaru da radi sledeće korake:

Nacrtajte kvadrat dužine stranice s tako da mu se dijegonale seku u tački sa koordinatama x, y.

Ako je stranica kvadrata veća od 2 piksela tada,

CrtajKvadrat (x+s/2, y, s/2)

CrtajKvadrat (x-s/2, y, s/2)

Pitanje:

Koja od slika ispod može biti stvorena komandom CrtajKvadrat?

|  |  |  |
| --- | --- | --- |
| A |  | Netačno |
| B |  | Tačno |
| C |  | Netačno |
| D |  | Netačno |

## Tačan odgovor:

## B

**Informatička pozadina:**

Većina programskih jezika podržava rekurziju. Ovo omogućava da procedure sebe pozivaju da bi rešili manje podprogram. Ovo može biti korisno kada se radi sa složenim strukturama podataka.

## 2016-CA - Map Reduce

Alonsov računar obrađuje informacije na specifičan način, koristeći malo operacija:

Prva operacija:

(R f (x1, x2, … xn)) koja kao rešenje daje

x1 f x2 f … f xn

pri čemu je f je matematički simbol a xi su brojevi

Primer 1: (R + (1, 2, 3, 4)) će dati 1 + 2 + 3 + 4 a to je 10.

Druga operacija:

(M f (x1 x2 ... xn)) koja kao rešenje daje

(f(x1), f(x2), … f(xn))

Pri čemu je f funkcija a xi su brojevi.

Primer 1: f(x) = -x i (M f (1, 2, 3, 4)) će dati (-1, -2, -3, -4).

Primer 2: t(x) = 3x + 2 i (M t (1, 2, 3)) će dati (5, 8, 11).

Pitanje:

Neka je t(x) = 3x + 2 i q(x) = -x. Šta se dobija kada se obavi sledeća operacija:

(R + (R + (M t (0, 2, 4))) (R + (M q (M t (3, 5)))))?

A. 7

B. 0

C. -7

D. -4

**Tačan odgovor:**

-4

**Informatička pozadina:**

Ovo pitanje bavi se funkcionalnim programiranjem i sposobnošću ugnežđavanja izraza. Funkcije koje operišu sa funkcijama su funkcije višeg reda. Ovakve funkcije utiču na skraćenje procesa programiranja. Ovaj koncept je jedan od osnova funkcionalnog programiranja.

Koncepti mapiranja i smanjene forme čine osnovu za mnoge "big data" algoritme, kao što su MapReduce i Hadoop. Ovakvi modeli olakšavaju obradu i kreiranje velike količine podataka.

## 2016-NL - Funkcija Total Swap (n,s)

Funkcija TotalSwap koristi dva ulaza: broj i reč. Izlaz je modifikovana reč. Ispod, u tabeli, možete videti kako se funcija TotalSwap ponaša sa ulazima (3, 'računar'):

TotalSwap(3, 'računar')

računar

Rotiraj slova na poziciji 1 i 3

čarunar

Rotiraj slova na poziciji 2 i 4

čuranar

Rotiraj slova na poziciji 3 i 5

čunarar

Rotiraj slova na poziciji 4 i 6

čunarar

Rotiraj slova na poziciji 5 i 7

čunarar

Pitanje: Koji će biti rezultat funkcije TotalSwap(4, 'BEBRAS')?

1. RABBES

2. RASBBE

3. RASBEB

4. RABSEB

**Tačan odgovor:**

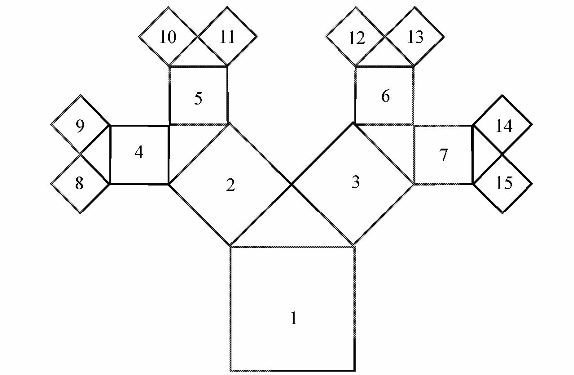
RASBEB

**Informatička pozadina:**

U programiranju, zamena dve promenljive se odnosi na međusobnu razmenu vrednosti promenljivih. Funkcije su važan deo svih programskih jezika. One su veoma pogodne za ponovno korišćenje koda kao i za izradu programa lakih za razumevanje. Funkcije su, takođe, sastavni deo osnova pragramiranja.

## 2012-LT - Magično drvo

Magično drvo je puno kućica(koja se crta od kvadrata i trougla) i svaka ima svoj broj kao na slici ispod:



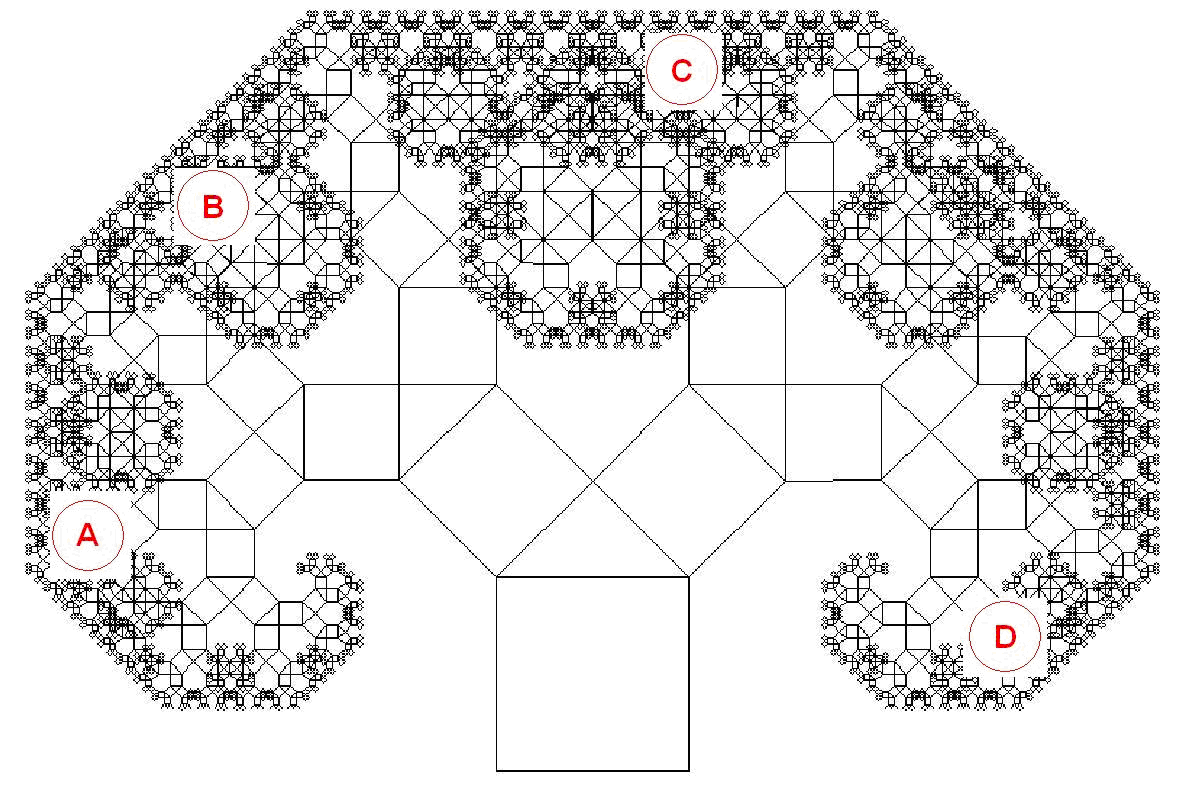
Magično drvo se može nacrtati ukoliko se prate sledeći koraci:

Start: uvek se počinje crtanjem kvadrata koji se obeležava rednim brojem 1;

Korak 1: na vrhu svakog nacrtanog kvadrata nacrta se trougao i dva manja kvadrata koja se obeležavaju sledećim brojevima ali uvek u smeru kazaljke na satu.

Korak 2: idi na Korak 1;

Pitanje: Na kom delu je kvadrat pod rednim brojem 138, na slici ispod?



A

B

C

D

**Tačan odgovor:**

A

## 2013–RU- Masa i visina

Četiri dabra Adam, Beti, Čarli i Dana stali su, svaki posebno, na vagu i evo rezultata:

Adam: 14 kg,

Beti: 13 kg,

Čarli: 12 kg,

Dana: 10 kg

Zatim su izmerili svoje visine:

Adam: 5 dm

Beti: 7 dm

Čarli: 10 dm

Dana: 11 dm

Dabrovi imaju formulu za izračunavanje idealnih mera: M\*masa+N\*Visina (M>= 0, N >= 0, dok je masa u kg a visina u cm).

Na primer, ako je M = 2 i N = 3, tada se Adamova idealna mera računa ovako: 2\*14 + 3\*50 = 175.

Pitanje

Koji dabar nikada neće imati najveću idealnu meru bez obzira koje brojeve M i N izaberemo?

A) Adam

B) Beti

C) Čarli

D) Dana

**Tačan odgovor:**

Beti

## 

## 2013–RU- Drevno dabarsko pismo

U dabrovskom muzeju nalazi se ploča na kojoj je ispisano drevno dabarsko pismo.



Dabar Mitros je, plivajići u reci Tajn, našao četiri kamena na kojima je nešto napisano drevnim dabarskim pismom.

Sigurno je da je samo jedan od četiri nađena kamena pravi a da su ostala tri lažna. Dabar Mitros je smislio metodu kojom će odrediti koji je kamen pravi. Provera koja daje najveći rezultat ukazuje na pravi kamen.

Evo Mitrosove metode:

Sve simbole koji su napisani na kamenu, Mitros deli na uzastopne parove simbola (broj parova će uvek biti za jedan manji od ukupnog broja simbola na kamenu). Zatim će svaki par simbola proveriti da li ga ima i ako ga ima koliko puta se ponavlja na ploči koji se nalazi u muzeju. Dobijene brojeve će izmnožiti. Tako će uraditi za svaki kamen.

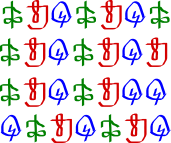
Na primer, ukoliko je na kamenu zapisano  , Mitros će zapis podeliti na sledeće parove: ,  i . Zatim će svaki par znakova proveriti da li ga ima i ako ga ima koliko puta se ponavlja na ploči koji se nalazi u muzeju.

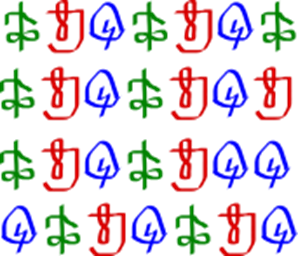
Par simbola  se pojavljuje tri puta,  dva puta dok se  pojavljuje jednom.

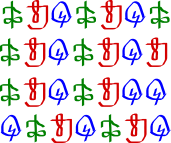
Na kraju, Mitros računa 3\*2\*1=6.

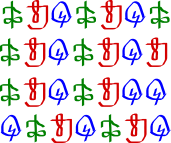
Pitanje:

Koji kamen je, na osnovu Mitrosove metode, pravi?

1. 

2. 

3. 

4. 

**Tačan odgovor:**



## 

## 2016-CA-Application

Jelena želi da se igra sa Lazarom. Jelena ima nekoliko karata sa geometrijskim oblicima i želi da ih zameni za neke karte koje ima Lazar. Jelena koristi ova pravila zamene u igri:

**→**  i **→**

Prema pravilima igre :

* Karta sa pravougaonim oblikom se zamenjuje sa dve karte sa trougaonim oblicima,
* Jedna karta sa trougaonim oblikom se zamenjuje sa jednom kartom sa pravougaonim oblikom, jedno kartom sa trougaonim oblikom i još jednom kartom sa pravougaonim oblikom.

Ako je Jelena započela igru sa kartom na kojoj je pravougaoni oblik, poštujući pravila igre, nakon tri zamene ona ima sledeće karte:

**→** **→** **→**

Koje od novih pravila igre bi po završetku igre, za rezultat imalo sledeće karte:

Odgovori:

A) **→** , **→** , **→**

B) **→** **→** , **→**

C) **→** **→** **→**

D) **→ →**  **→**

**Tačan odgovor je:**

B) **→** **→** , **→**

Primetimo sledeće:

→ → →

Kako bismo isključili ostale 2 mogućnosti, moramo da istanemo sledeća zapažanja.

Za odgovor pod (A), ako bismo počeli igru sa kartom na kojoj je krug ili trougao, nikada ne bismo došli do karte sa pravougaonim oblikom . Ako bismo započeli sa kartom sa pravougaonim oblikom, imali bio

→ → →

naredni koraci bi doveli do obrasca koji je previše dugačak.

Za dogovor pod (C), ako započnemo igru sa kartom koja ima trougaonu oznaku , za rezultat bismo imali samo trougaone karte.

Ako započnemo sa → → → → ... i prethodne četiri trougaone karte nikada neće biti zamenje sa .

Ako započnemo sa → → →

.... i , kao i u prethodnom slučaju, prethodne četiri trougaone karte nikada neće biti zamenje sa .

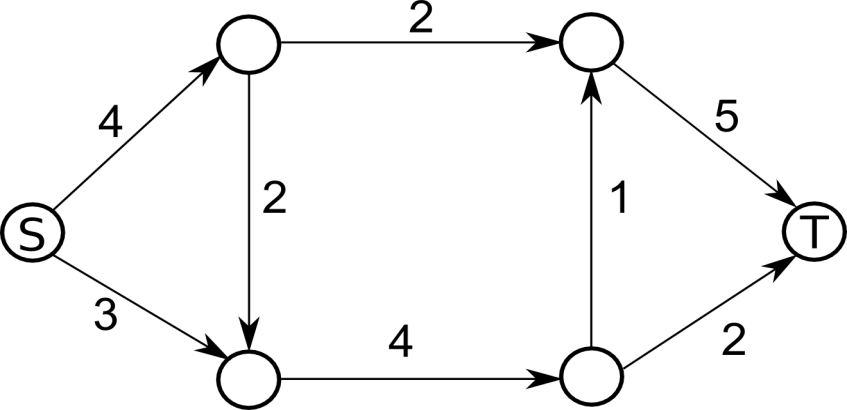
Za odgovor pod (D), ne možemo generisati dati raspored.

### Informatička pozadina

Pravila koja su ovde opisana pokazuju niz prepisivanaj pravila, koja se koriste u “context-free grammar” ili drugim gramatičkim sistemima. Ovi gramatički sistemi mogu opisati stvari kao što su:  
• prirodne pojave, kao što je rast biljaka,  
• prirodni jezik, kao što su gramatička pravila formiranja rečenice,  
• formalni jezici, kao su programski jezici strukturirani.  
Pitanje ovde traži izvođenje ili raščlanjavanje datih reči koristećenjem različitih pravila.  
Raščlanjavanje je jedan važan korak u prevođenju programa sa reči tj. "čitljivo za ljude", na binarne brojeve, tj. "čitljivo za računare".

**2016-CA-07- Prevoz trupaca**

Dabar Milana mora da tegli trupce preko sistema kanala. Trupci se između stanica koje su obelezene krugovima, moraju prevoziti u pravcu u kome je usmerena strelica. Kanali za prevoz imaju ograničenja u broju trupaca koji se mogu prevesti u toku jednog dana, što je prikazano brojevima iznad strelica. Iz jedne stanice Milan može prevoziti trupace različitim kalima.



Pitanje:

Koji je najveći broj trupaca koji Milan može da preveze iz stanice S do stanice T u toku jednog dana?

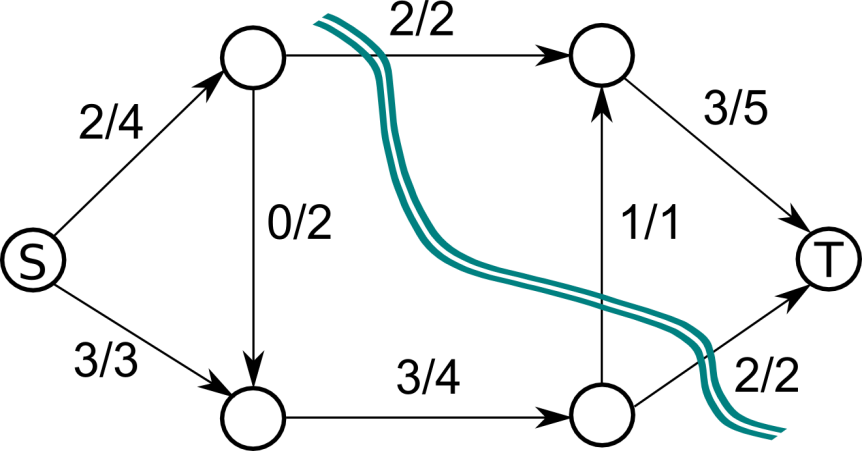
Odgovor

(A) 4

(B) 5

(C) 6

(D) 7

**Tačan odgovor:**

5

Obzirom na sledeću situaciju, gde prvi broj na strelicama određuje broj trupaca koje Milan prevozi preko svakog kanala, a drugi broj određuje njegov maksimalni kapacitet:

**Informatička pozadina**

U ovom zadatku imamo problem odrđivanja maksimalnog protoka u grafu.

Postoji teorema (FordFulkersonova, 1956) koja kaže da je maksimalan protok putanje u grafu određen je najmanjom vrednošću koju ima njena pojedinačna grana.

U našem problemu, grane sa najmanjom vrednošću su označen krivom linijom: uklanjanje ovih grana bi prekinulo protok od S to T. Ukupan kapacitet protoka je (5), gde je i minimum ako posmatramo sve moguće gane. Dakle, koristeći Ford Fulkersovu teoremu došli smo do toga da je maskimalan protok ovog grafa Problem maksimalnog protoka u grafu čest je u informatici. Recimo da je najčešći primer Internet, gde želimo da pronađemo maksimalan kapacitet seta rutera kako bismo preneli onoliko podataka kolko je to moguće.

Međutim, postoje i drugi primeri povezani sa informatikom, gde se ovaj problem javlja, kao što su: tobrada fotograrije, raspored letanja aviona, mrežni kapacitet, koncept disribucije procesorske snage računara itd.

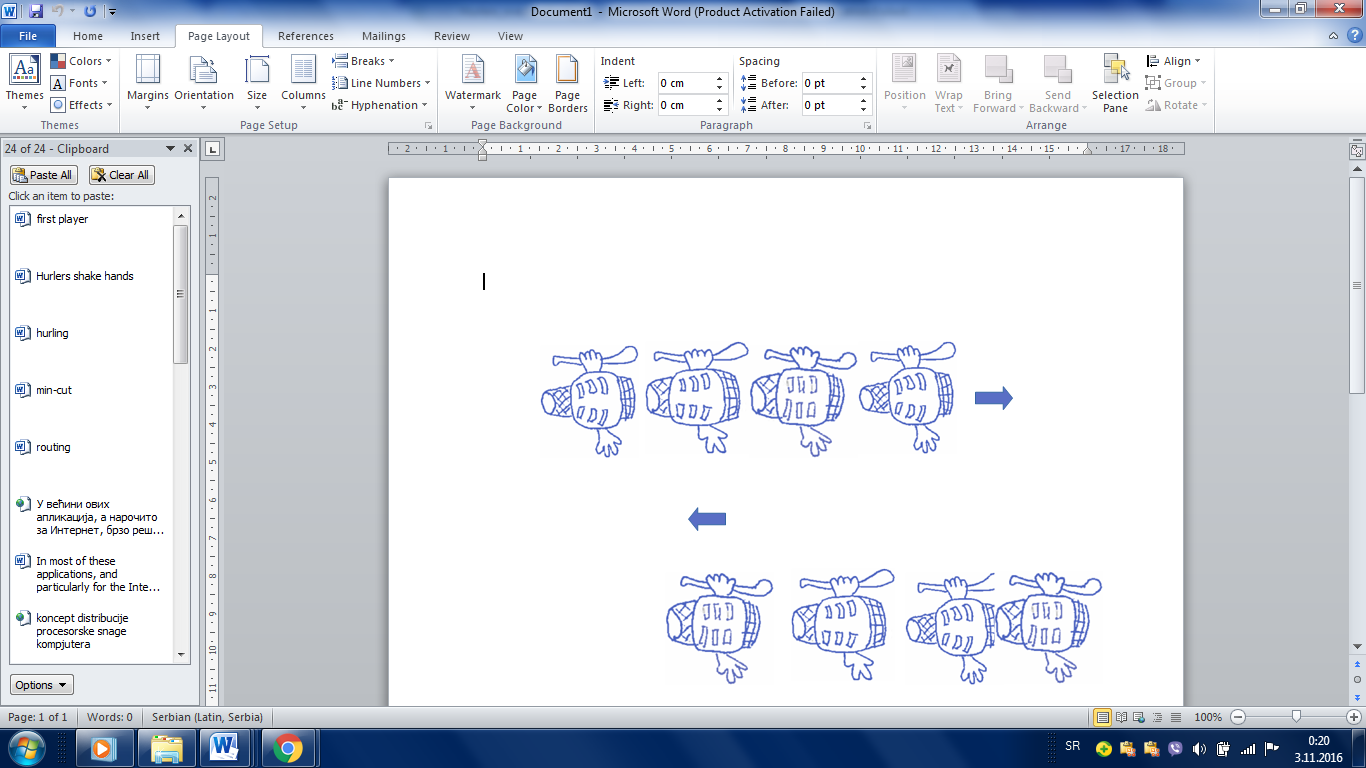
Kod većini ovih aplikacija, naročito u slučaju Interneta, brzo rešavanje ovog problema je važno jer se Internet sastoji od miliona takvih čvorova, svaki sa svojim kapacitetom, tako da je rutiranje/usmeravanje kroz rutere u ovoj mreži treba da se odradi što je moguće efikasnije.

Postoje dobri algoritmi za rešavanje ovog problema, koji su efikasni čak i za velike grafove.

## 2016-IE- Pozdravljanje timova

Dabrovi su veoma dobri u irskoj igri koja se zove hurling. Kada god se igra harlinga završi, dabrovi timova stanu u dva reda, jedan iza drugog i prolaze u mimohodu pored igrača drugog tima. U prolazu se rukuju i pozdravljaju rečim “Hvala ti za igru!”

Na početku, rukuju se samo prvi igrači timova. Zatim, prvi igrači se rukuju sa drugim igračima timova (pogledaj sliku ispod). Ovo se nastavlja sve dok se svaki igrač tima ne rukuje sa svakim igračem protivničkog tima.



### Pitanje

### U igri hurling, učestvuje po 15 igrača svakog tima. Ako svaki igač potroši 1 sekundu dok se rukuje i pomeri do sledećeg protivničkog igrača, kolko sekundi će biti potrebno da se obavi rukovanje timova.

**Tačan odgovor:**

29

**Informatička pozadina**

Broj rukovanja je jednak dužni jednog reda plus dizna drugog reda, minus jedan.

Hajde da zamislimo da postoji samo po 1 igrač u svakom timu. Nakon 1 sekunde oni će završiti rukovanje. Ako zamislimo da postoje po 2 igrača u oba tima. U toku prve sekunde, prvi igrači timova će se međusobno rukovati. Tokom druge sekunde, prvi igrači će se rukovati sa drugim igračima timova i tokom treće sekunde, drugi igrači timova će se međusobno rukovati. Dakle, to je 3 sekunde.

Sa 15 igrača u svakom timu, broj sekundi koji je poteban je 15 + 15 – 1 = 29.

Ovaj zadatak može biti sagledan kao ilustracija paradigme paralelne obrade, poznate kao protočna obrada. Protočna obrada je veoma efikasan način za rad više računara zajedno na brzom rešavanju istog problema. Isto kao i kod naših igrača poslednji u svakom redu moraju da sačekaju dok se najpre prethodni rukuju.

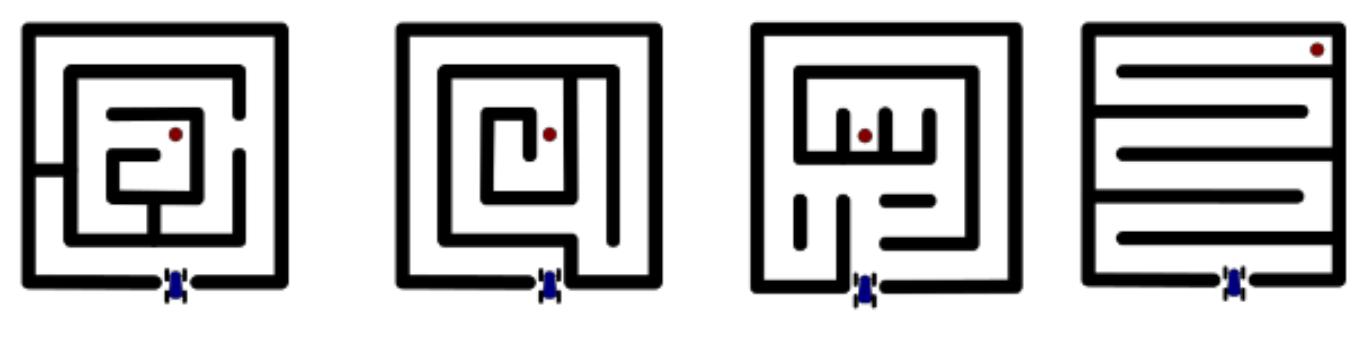
Analiza trajanja jednog algoritma je sofisticirani deo računarske nauke koji se zove složena računarska analiza. U zadatku znamo da su veličine timova fiksirane na 15, pa možemo zaključiti da je “potrebno vreme realizacije” da se izvrši algoritam rukovanja igrača 29 sekundi. Međutim, u računarskim slozenim proračunima može se zahtevati da računamo vreme trajanja algoritma nezavisno od veličine tima. Možemo zaključiti da algoritam rukovanja igača ima vreme trajanja od 2N-1 sekundi, za svaku veličinu tima N, gde je N 1 ili bilo koji veći prirodan broj.

## Lavirint

Robot se kreće kroz lavirinte pridržavajući se ovih pravila: kada god je to moguće – skreni desno, ako nije moguće – nastavi pravo, ako ne može pravo – idi levo. Pažljivo pogledajte primer sa desne strane!

Ukoliko poštuje pravila, u kojem lavirintu će uspeti da dođe do crvene takčke?

**A** **B** **C** **D**

****

**Tačan odgovor:**

U svim osim u lavirintu C. U lavirintu C robot ne može da stigne do centralnog dela, pa tako ni do crvene tačke.

**A** **B** **C** **D**

****

**Informatička pozadina**

U pitanju je jednostavan algoritam koji omogućava vožnju kroz bilo koji lavirint. Sledeći pravila, robot se nikad neće izgubiti - uvek će moći da se vrati na početnu tačku. Algoritam ne garantuje da će robot proputovati ceo lavirint.

## Tunel

Porodica dabrova je krenula u brda i usput naišla na tunel. Tunel je veoma uzak i taman, iz bezbednosnih razloga kroz njega se mogu kretati jedna ili dve osobe koje moraju imati baterijsku lampu sa sobom. Porodica kod sebe ima samo jednu baterijsku lampa.

Članovi porodice se kreću kroz tunel različitim brzinama: Milanu treba 5 minuta, njegovoj sestri Ani dva puta toliko, mami treba 20 minuta, a tati 25 minuta. Porodica želi da stigne na drugu stranu tunela za jedan sat.

Koliko je potrebno vremena porodici da se najbrže svi prebace na drugu stranu tunela?

A. 35 minuta

B. za 65 minuta

C. 60 minuta

D. Više od 65min

**Tačan odgovor:**

60 min

Porodici treba šezdeset minuta. Mama i tata moraju da pređu tunel u isto vreme i niko od njih dvoje ne bi trebalo da se vraća. Zato kroz tunel prvo mora da prođu Milan i Ana (10 minuta), jedno od njih se vaća na početak tunela (5 ili 10 minuta), zatim kroz tunel prelaze mama i tata zajdeno (25 minuta). Drugo dete koje je ostalo da čeka na drugom kraju tunela se vraća po brata ili sestru (5 ili 10 minuta), zatim brat i sestra zajdeno prolaze kroz tunel (10 minuta).

**Informatička pozadina**

Kompjuterski naučnici često traže rešenje problema uz poznata i unapred data ograničenja. Znači da se problem treba rešiti što je moguće brže i lakše. To se zove optimizacija.

## 2016-BE - Pronađi lopova

OH NE! Poznati Plavi dijamant ukraden je iz muzeja danas: lopov ga je zamenio jeftinom imitacijom zelene boje.



Izložbu Plavog dijamanta posetilo je 2000 ljudi danas. Ulazili su u sobu sa dijamantom jedan po jedan. Inspektor Dabar mora da pronađe lopova ispitivanjem nekih od posetilaca. On ima listu svih 2000 posetilaca, po redosledu kojim su ulazili u sobu. Svakom posetiocu postaviće isto pitanje: Da li je dijamant bio zelene ili plave boje kada si ga video? Savak osoba će odgovoriti istinito, osim lopova, koji će reći da je dijamant već bio zelen.

Pitanje

Inspektor Dabar je veoma pametan, primeniće strategiju ispitivanja tako da broj ispitanih posetilaca bude najmanji mogući. Koju od sledećih izjava može da kaže bez laganja?

* Mogu da garantujem da ću naći lopova ispitivanjem manje od 20 ljudi.
* Ispitivanje 20 ljudi neće biti dovoljno ( osima ko sam baš srećan) ali sigurno mogu pronaći lopova ispitivanjem manje od 200 ljudi.
* Ovo će biti težak posao: Moraću da ispitam najmanje 200 ljudi, a možda čak i 1999.
* Ne mogu da obećam ništa. Ako nemam sreće možda ćumorati da ispitam sve posetioce.

**Tačan odgovor:**

* Mogu da garantujem da ću naći lopova ispitivanjem manje od 20 ljudi.

Iznenađujuće, inspektor Dabar mora da ispita veoma malu grupu ljudi.

Broj posetioca od 1 do 2000 kako su oni ulazili u sobu.

Inspektor Dabar prvo pita posetioca sa brojem 1000 koje boje je bio dijamant kada ga je on video

Ako on odgovori plave boje, lopov mora da je bio nakon broja 1000, sa brojem između 1001 i 2000

Ako odgovori zeleno, lopov mora da je bio pre broja 1000, sa brojem između 1 i 1000 ( imajte na umu da lopov može biti i posetilac sa brojem 1000)

U ba slučaja, broj posetilaca koji potencijalno mogu biti lopovi se smanjuje sa 2000 na 1000. Drugim rečima, prepolovljen je.

Zatim inspektor postavlja pitanje „srednjoj“ osobi na preostaloj listi. ( Broj 1500 u prvom slučaju, broj 500 u drugom). To mu ponovo omogućava da smanji listu osumljičenih na pola.

Nastavljajući na isti način, moćiće da umanji broj osumljičenih na 500, 250, 63, 32, 16, 8, 4 i na kraju na 2. Kada ostanu 2 osumljičena, osdtaviće pitanje prvom. Ako on odgovori ,,Zeleno“, on ej lopov, u suprotnom lopov je 2.

Inspektor će pronaći lopova ispitivanjem samo 11 osoba.

**Inforamtička pozadina**

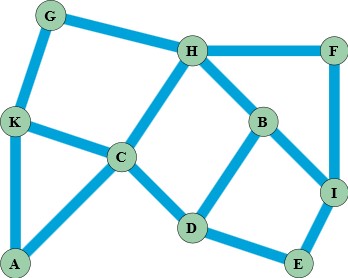
Binarna pretraga.

U našem slučaju koristimo ,,Binarnu pretragu“ da pronađemo prvi ,,Zeleni“ element u listi plavih i zelenih elemenata, gde ,,Plavi“ elementi dolaze prvi. Činjenica da lopov uvek laže je zanimljiv obrt.

Problem je nerešiv ukliko ne znamo koje će boje lopov koristiti u svom odgovoru.

## 2016-CH- Zdravstveni centar





Dabrovi žive na raskrsnicama kanala, označenih slovima. Grad će izgraditi tri zdravstvena centra. Zdravstvene centre žele da postave tako da dabrovi plivaju najviše kroz jedan kanal do najbližeg, bez obzira na kojoj su raskrsnici.

**Pitanje**

Na kojim raskrsnicama će izgraditi zdravstvene centre?

Odgovor

1. EHK
2. EHF
3. ADH
4. CGB

**Tačan odgovor:**

Postoji više mogućih odgovora, jedno od tačnih je E, H i K.

Dabrovi koji žive na raskrsnicama D, E i I mogu da koriste Zdravstveni centar na raskrsnici E

Dabrovi koji žive na raskrsnicama B, C, F, G i H mogu da koriste Zdravstveni centar na raskrsnici H

Dabrovi koji žive na raskrsnicama A, C, G i K mogu da koriste Zdravstveni centar na raskrsnici K

Druga moguća rešenja koja nisu ponuđena su AEH, CGI, CHI, CIK, DFK.

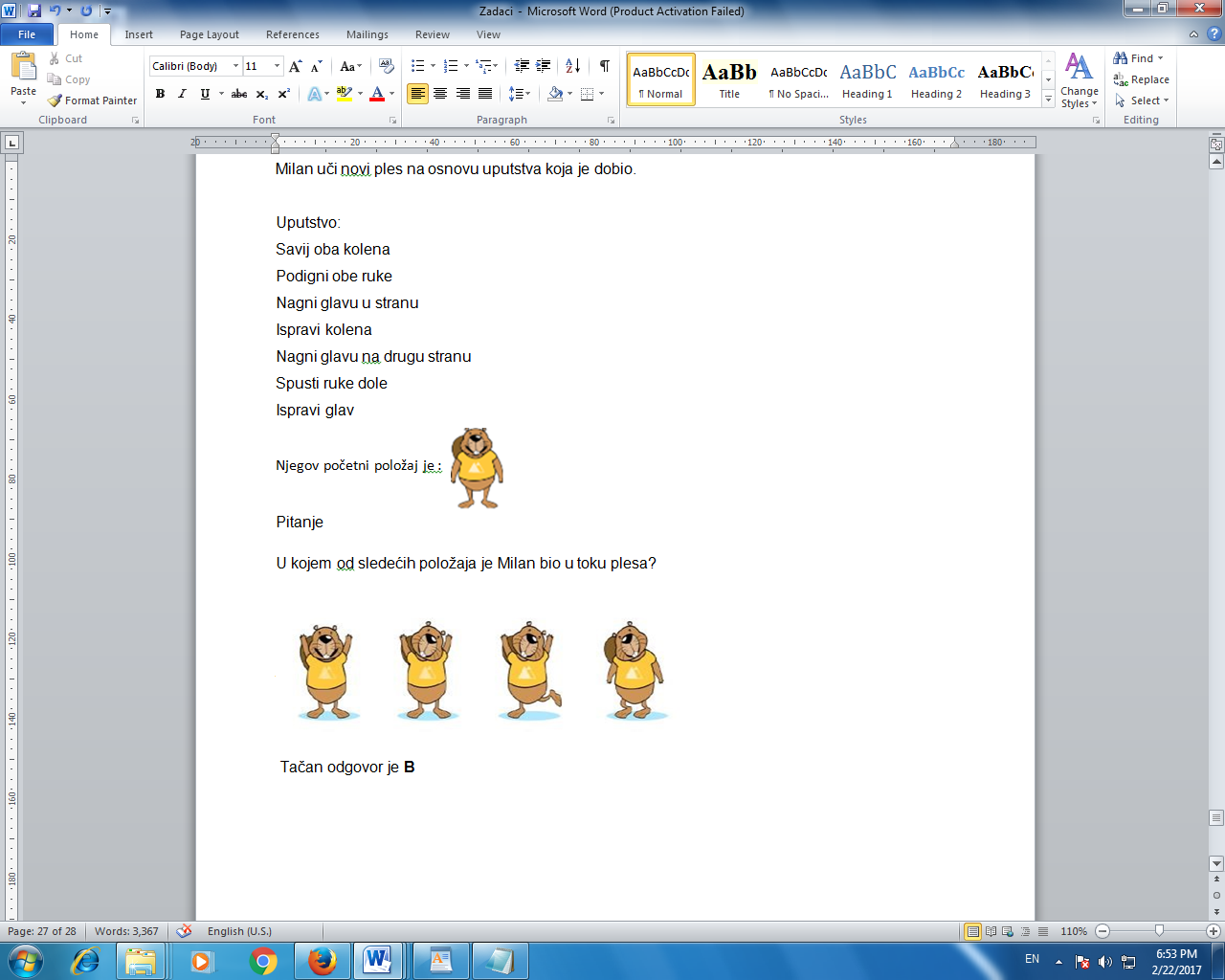
**Informatička pozadina**

Teorija grafova.

Problem, kojije opisan zadatkom je jedan od poznatijih informatičkih problema za koje informatički eksperti još uvek nisu pronašli rešenje. Ako bi broj dabrova bio mnogo veći, bilo bi potrebno i mnogo više domova zdravlja. Čak i najbržim računarima bi trebalo mnogo vremena da pronađu najoptimalnija rešenja pri čemu se programi kreiraju tako da se testiraju sve moguće kombinacije da bi se došlo do onih koje ispunjavaju sve uslove. U ovako jednostavnim primerima, metoda testiranja svih kombinacija je efikasna, ali u složenijim pojavljuju se problemi.

## 

## 2016-SK- Igranje



## Rođendanska torta

Dabar Pejča puni jedanaest godina. Njegova mama je kupila jedanaest svećica različitih boja ali je slučajno izgubila šest svećica.



Pejčina mama zna kako da sa samo pet svećica prikaže broj jedanaest na slavljeničkoj torti, prema sledećim pravilima:

• Prva svećica sa desne strane predstavlja broj jedan

• Svaka sledeća svećica predstavlja duplo veću vrednost od prethodne

• Računaju se samo svećice koje gore

• Ukoliko je nekoliko svećica upaljeno, sabiraju se vrednosti

• Svećice su postavljene po sledećem rasporedu, gledano sa desne na levu stranu: plava, crvena, zelena, žuta i roze.

Evo primera kako Pejčina mama prikazuje određene brojeve:



Pitanje:

Koje svećice će Pejčina mama zapaliti da bi prikazala broj jedanaest?

**Tačan odgovor:**

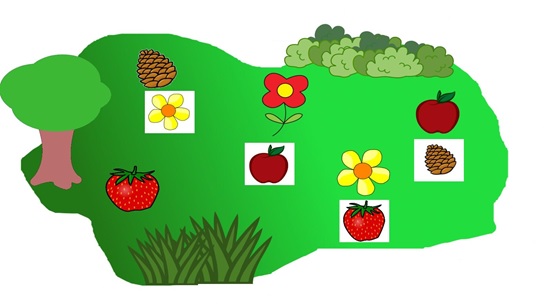
Plavu-Crvenu-Žutu

## Tajni recept

Dabrica Ceca se priprema za festival hrane. Ona bi želela da ponese Hrskavi kolač, ali njen kuvar je na odmoru. Ceca je odlučila da sama pripremi kolač ali jedino čega se seća je da je važno dodati 5 sastojaka određenim redosledom.

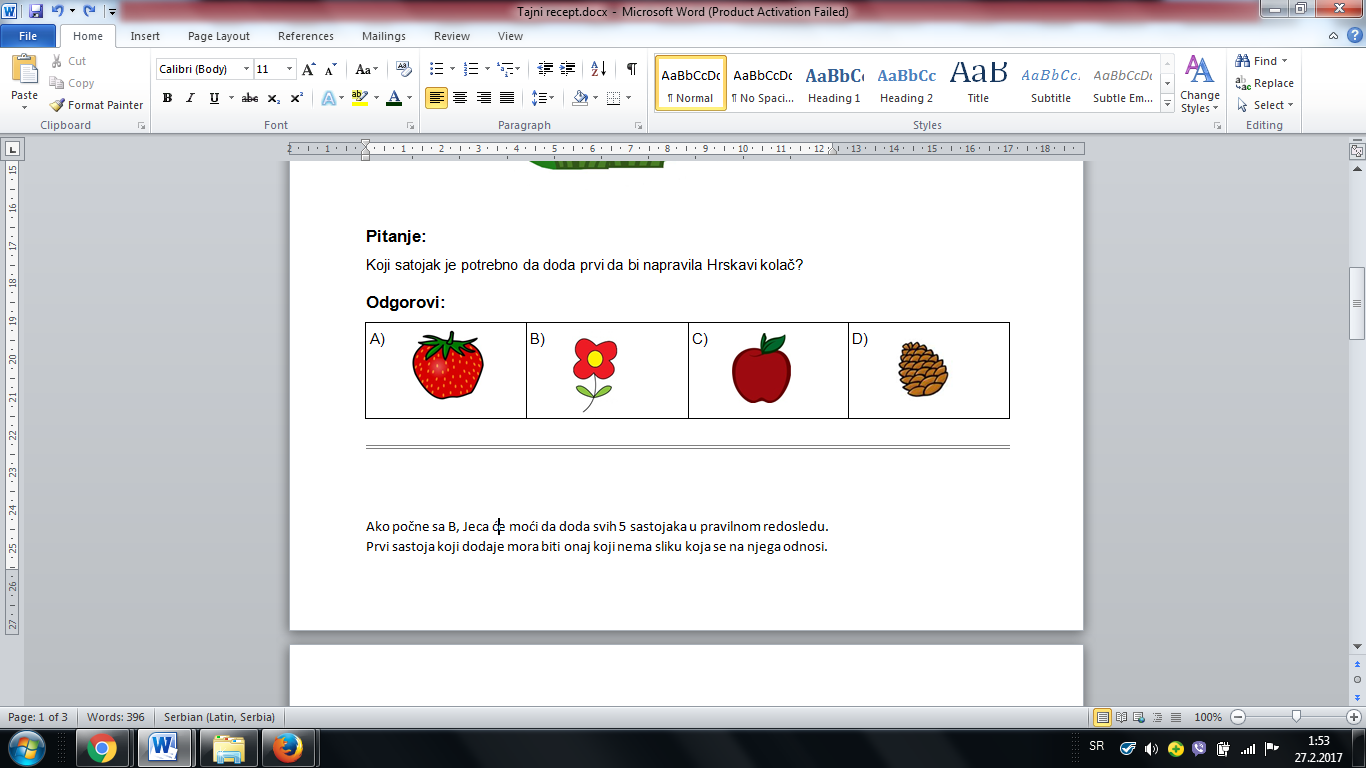
Kada je otišla u baštu, primetila je da pored svakog sastojka stoji parče papira koje prikazuje sliku sastojka koji treba da se doda sledeći.

Postoji samo jedan sastojak pored koga ne stoji papir. Bašta izgleda kao na slici ispod.



Pitanje:

Koji sastojak se prvi dodaje da bi se napravio Hrskavi kolač?



**Tačan odgovor:**



Ako počne sa B, Jeca će moći da doda svih 5 sastojaka u pravilnom redosledu. Prvi sastoja koji dodaje mora biti onaj koji nema sliku koja se na njega odnosi.

Ako izabere jagode, ne može da nastavi dalje jer ne postoji papir sa slikom sledećeg sastojka. Jabuke nisu pravi izbor, jer ako počne sa jabukama moraće da preskoči crveni cvet.

Šišarka nije dobar izbor, jer ako započne sa njom moraće da preskoči crveni cveti i jabuke.

**Informatička pozadina**

Struktura podataka koja se ovde koristi zove povezane liste u kojima ne može biti proizvoljan broj stavki. Povezane liste su linearna kolekcija elemenata podatak koji se sastoji od jedne stave i referentne tačke (Pointer) koja pokazuje sledeću stavku. Prva tačka kod povezanih lista veoma je važna, jer lisa počinje odatle i to je jedina takča koja se odnosi na celu listu.

Recept za Hrskavi kolač je povezana lista. Sastojci su elementi, a svaki list papira je pokazivač sledećeg elementa koji je na redu. Drugim rečima, biljke su podaci, a papiri su pokazivači

Prva komponenta je sastojak na koji se ne odnosi ni jedna papir, ali koji ima papir pored sebe. Ove stavke sadrže samo jednu referencu, tako da se zna samo sledeći element u redu, ali ne postoji trag o prethodnom.

Korist od povezanih listi je da se stavke različitog tipa i veličine mogu čuvati zajedno, kao i voće i cveće u ovom primeru.

Ova struktura podataka takođe može biti i nelinearna. Moguće je ukazivati na istu stavku sa različitih povezanih listi, a takođe je moguće da jedna stavka ukazuje na drugu stavku u drugoj povezanoj listi .

## 

## Ključevi

Dabar Siniša je stigao do svog stana i kada je izvadio svežanj sa ključevima oni su se rasuli po podu. Samo jedan ključ otvara bravu. Pomozite Siniši da otključa stan.



Pitanje: Koji ključ otključava bravu?

|  |  |
| --- | --- |
| A) | B) |
| C) | D) |

**Tačan odgovor:**





Odgovor pod C) je tačan, ovaj ključ odgovara otvoru.



Samo ova dva ključa imaju isti oblik kao i otvor u bravi, ključevi pod C) i A). Ali samo jedan od njih je odgovarajući. Nakon što okrenemo ključ pod A), vidimo da on nije odgovarajućii (slika ispod).



**Informatička pozadina**

### Da bi se rešio ovaj zadatak, morate da zamislite koja procedura mora se izvršiti kako bi se testiralo da li je izabrani ključ odgovara otvoru brave. Nije dovoljno samo da proverimo oblik. Potrebno je rotirati ključ oko vertikalne ose, a zatim oko anteroposterior ose (napred-nazad).

Pri programiranju ruke robota, potrebno je zamisliti kako se ona rotirata tako da može da postavi objekat u željeni položaj. Pokret hvataljke robota je programiran u 6 dimenzija,  
3 za pozicioniranje na x, y i z koordinatama, i tri za rotiranje u x, y, z koordinatnom sistemu.

U ovom zadatku, kako bi robot otvorio bravu pomoću odgovarajućeg ključa, prvo mora da ga rotira oko vertikalne ose, a zatim rotira oko sagitalne ose (desna na levo) i konačno da se kreće duž sagitalne ose.

## 

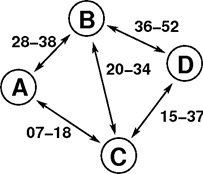
## Planiranje putovanja

Mapa i strelice na njoj pokazuju kada i kako vozovi putuju između 4 grada А, B, C i D.

Dva broja zapisana pored strelice označavaju minute u tekućem satu kada će vozovi poći i stići iz jedne u drugu stanicu. Raspored se ponavlja na svakih sat vremena. Na primer, vozovi iz grada A ka gradu B polaze u 8.28, 9.28, 10.28 itd, a dolaze u grad B nakon 10 minuta tj. u 8.38, 9.38, 10.38 itd... Isto tako i vozovi iz grada B ka gradu A polaze u 8.28, 9.28, 10.28 itd..., a dolaze u grad A nakon 10 minuta tj. u 8.38, 9.38, 10.38 itd...

Pitanje:

Tačno vreme je 8.45 i dabrica Ceca je na železničkoj stanici u gradu A. Ona želi da putuje u grad D. Kada najranije može stići do grada D?



**Tačan odgovor:**

9.52