

Bober 2022/23

Bebras

ACM Slovenija

Naloge za tekmovanje je izbral, prevedel, priredil in oblikoval Programski svet tekmovanja:

Alenka Kavčič (UL FRI)

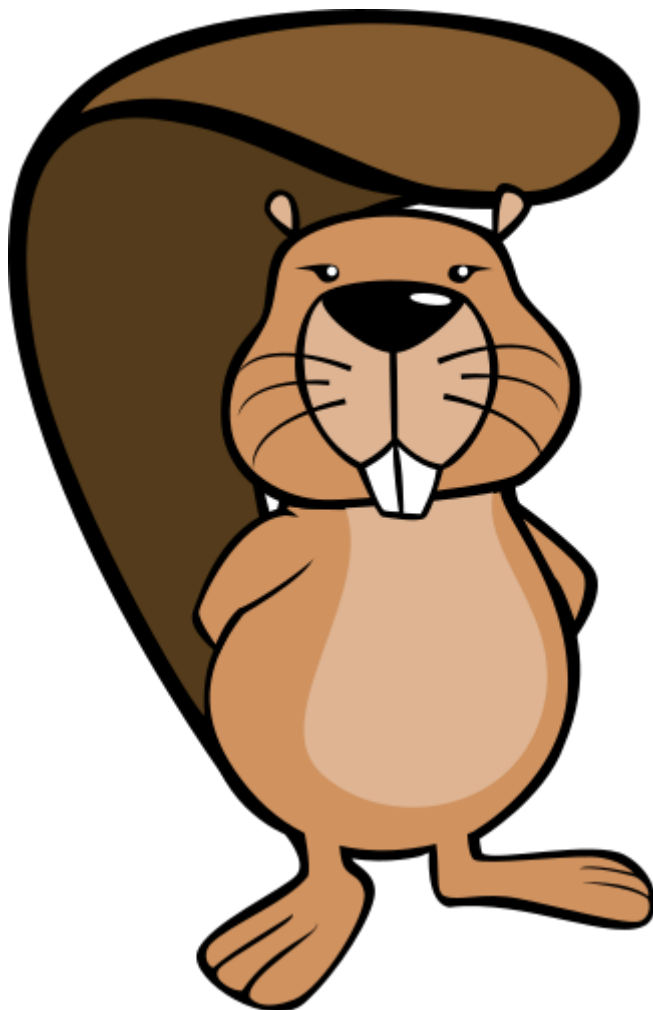
Anja Kneževič (OŠ Gradec)

Anja Koron (OŠ Branik)

Janez Demšar (UL FRI)

Nežka Rugelj (OŠ Gradec)

Špela Cerar (UL PEF)



Naloge in rešitve

ACM Slovenija



Naloge za tekmovanje je izbral, prevedel, priredil in oblikoval Programski svet tekmovanja:

Alenka Kavčič (UL FRI)

Anja Knežević (OŠ Gradec)

Anja Koron (OŠ Branik)

Janez Demšar (UL FRI)

Nežka Rugelj (OŠ Gradec)

Špela Cerar (UL PEF)

Razvoj tekmovalnega sistema:

Gašper Fele Žorž (UL FRI)

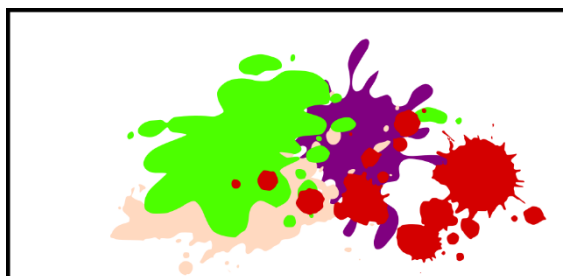
Gregor Jerše (UL FRI)

Kazalo nalog

PRI KOSILU	4	ZAKLAD	37
ABECEDA	5	SLIKE GOZDA	39
BARVANJE ROBOTOV	6	PREPROGA	42
ROBOTSKA ČEBELA 1	7	KLJUČKI	44
GOZDNA ZABAVA	8	ROBOT SMETKO 2	46
HAMBURGER	10	OGRVICE	48
OBISK BABICE	12	ROBOTSKI BOBER	51
SMUČARSKA VAS	15	TISKALNIK	52
ROJSTNODNEVNA ZABAVA	17	TOVARNA ROBOTOV	54
ROBOT SMETKO 1	19	DATOTEKE	56
KOVANCI	21	MOSTOVI	58
NAVODILA ZA GRADNJO	23	POSLUŠAJ IN HODI	60
ROBOTSKA ČEBELA 2	25	SEDEŽNI RED	62
MAJINI SOSEDI	27	SPOROČILA	64
STARODAVNE RUNE	29	LOV NA JAGODE	66
VIJAKI Z MATICO	30	PREDALČKI	68
BARVNA SESTAVLJANKA	33	PRILJUBLJEN FILM	69
ZAPOREDJE KAMENČKOV	34	PREGLED NALOG	71
ŠIFRA 8	36		



DANES SO PRI KOSILU NA ISTEM MESTU EN ZA DRUGIM JEDLI ANITA, BOJAN, CIRIL IN DEJAN, VENDAR NE V TEM VRSTNEM REDU. PRAV VSI SO S SVOJO JEDJO POPACKALI MIZO.



ANITA JE JEDLA ŠPINAČNO JUHO ().

BOJAN JE JEDEL PARADIŽNIKOVO JUHO ().

CIRIL JE JEDEL BOROVIČEVO PITO ().

DEJAN JE OD DOMA PRINESEL HUMUS ().

V KAKŠNEM VRSTNEM REDU SO KOSILI, ČE JE MIZA PO KOSILU IZGLEDALA, KOT KAŽE SLIKA?

- | | |
|-------------------------------|-------------------------------|
| A) ANITA, CIRIL, BOJAN, DEJAN | B) BOJAN, DEJAN, ANITA, CIRIL |
| C) CIRIL, DEJAN, ANITA, BOJAN | D) DEJAN, BOJAN, CIRIL, ANITA |

REŠITEV

JEDLI SO PO VRSTNEM REDU C: NAJPREJ JE MIZO UMAZAL CIRIL Z BOROVIČEVO PITO, NATO DEJAN S HUMUSOM, ZA NJIM JE ANITA MIZO UMAZALA S ŠPINAČNO JUHO, NA KONCU PA JE UMETNINO DOKONČAL BOJAN S PARADIŽNIKOVO JUHO.

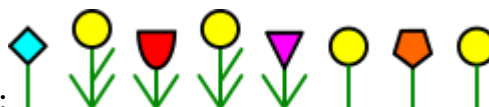
Računalniško ozadje

Packe se v našem primeru na mizo nalagajo kot sklad - packa, ki je nastala najprej, je pod vsemi ostalimi, tista, ki je nastala zadnja pa je na vseh ostalih.



BOBRI IMAJO DRUGAČNO ABECEDO KOT MI, ZATO SI PRI BRANJU NJIHOVIH ZAPISOV POMAGAMO Z NASLEDNJIMI NAMIGI:

	A	B	C	Č	D
✓	E	F	G	H	I
↘	J	K	L	M	N
↘↘	O	P	R	S	Š
↘↘↘	T	U	V	Z	Ž



POLEG BOBRIŠČA JE NASLEDNJI ZAPIS:

ZAPIŠI GA S SLOVENSKO ABECEDO. _____

REŠITEV

POSAMEZNO ČRKO SLOVENSKE ABECEDA BOBRI ZAPIŠEJO KOT KOMBINACIJO LIKA PRIKAZANEGA NAD ČRKO IN OBLIKE, KI OZNAČUJE VRSTICO, V KATERI SE NAHAJA TA ČRKA. ZGORNJI ZAPIS TAKO PREDSTAVLJA BESEDO ČOKOLADA.

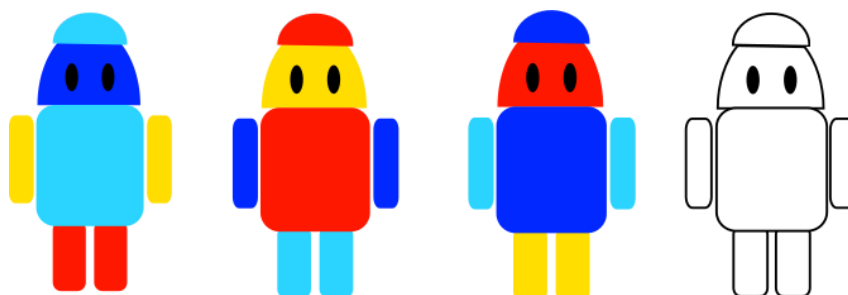
Računalniško ozadje

Tudi v računalniku so črke shranjene v drugačni obliki, kot jih razumemo mi. Črke so v računalniku shranjene v obliki zaporedja ničel in enic. Vsaki črki se določi neko število, ki mu rečemo *koda znaka*, ti znaki pa so urejeni s pomočjo tabel, ki jih imenujemo *kodne tabele*. Računalnik si besedo zapomni kot zaporedje teh kod (števil), ki pripadajo črkam besede.



ROBI IMA 4 ROBOTE, KI JIH JE POBARVAL Z RAZLIČNIMI BARVAMI: RDEČO, SVETLO MODRO, TEMNO MODRO IN RUMENO. VSAK ROBOT IMA POSAMEZEN DEL POBARVAN Z DRUGAČNO BARVO KOT VSI OSTALI.

POBARVAJ DELE ČETRTEGA ROBOTA Z USTREZNIMI BARVAMI.



REŠITEV

ČE SI OGLEDAMO POKRIVALA PRVIH TREH ROBOTOV, UGOTOVIMO, DA SO TA SVETLO MODRE, RDEČE IN TEMNO MODRE BARVE. KAR POMENI, DA BO POKRIVALO ČETRTEGA ROBOTA RUMENE BARVE, SAJ JE TO EDINA BARVA, KI ŠE NI BILA UPORABLJENA. KER SE MORAJO TUDI DRUGI DELI TELESA RAZLIKOVATI V BARVI, BODO ROKE ROBOTA RDEČE, NOGE TEMNO MODRE, TELO RUMENO IN GLAVA SVETLO MODRE BARVE.



Računalniško ozadje

Programerji morajo biti pri sestavljanju programov pozorni na različne omejitve in vzorce, ki se pojavijo v danem problemu. Tudi v tej nalogi je bilo potrebno z natančnim branjem omejitev in opazovanjem lastnosti robotov ugotoviti, katere barve bo posamezen del robota.



ROBOTSKA ČEBELA POZNA UKAZE:

NARAVNOST ↑	ČEBELA GRE ENO POLJE NAPREJ.
DESNO ↗	ČEBELA SE NA MESTU OBRNE V DESNO.
LEVO ↖	ČEBELA SE NA MESTU OBRNE V LEVO.

V GOZDU SO TRIJE PANJI, A LE RUMENI (R) PANJ JE DOM NAŠE ČEBELE.



KATERO ZAPOREDJE UKAZOV BO ČEBELO PRIPELJALO DO RUMENEGA PANJA? OBKROŽI.

A) ↑ ↗ ↖ ↑ ↗	B) ↖ ↑ ↗ ↑ ↑	C) ↑ ↗ ↑ ↖ ↑	D) ↗ ↑ ↖ ↑ ↖
--------------	--------------	--------------	--------------

Rešitev

ČEBELO BO DO RUMENEGA PANJA PRIPELJALO ZAPOREDJE B):

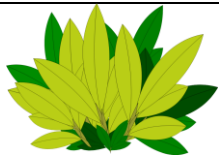


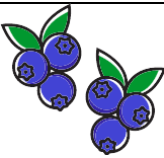





Računalniško ozadje

Programi, ki jih poganjamo na svojih računalnikih, so sestavljeni iz zaporedja ukazov, prav tako kot ti, s katerimi smo se igrali v tej nalogi, le da računalniški programi uporabljajo veliko več različnih ukazov in so veliko daljši.



PRIJATELJI BETI, LIZA IN MEDO IMAJO ZABAVO. TABELA PRIKAZUJE, KDO LAHKO POJE KATERO JED.

	 LISTJE	 RIBA	 GOBA	 BOROVNICE
 BETI	✓	✗	✗	✓
 LIZA	✗	✓	✗	✓
 MEDO	✗	✓	✓	✓

NA SLIKI VIDIŠ 9 JEDI, KI SO JIH PRIPRAVILI. KATERE JEDI BO POJEDEL VSAK OD NJIH, ČE VSAK POJE 3 JEDI? POVEŽI.



BETI

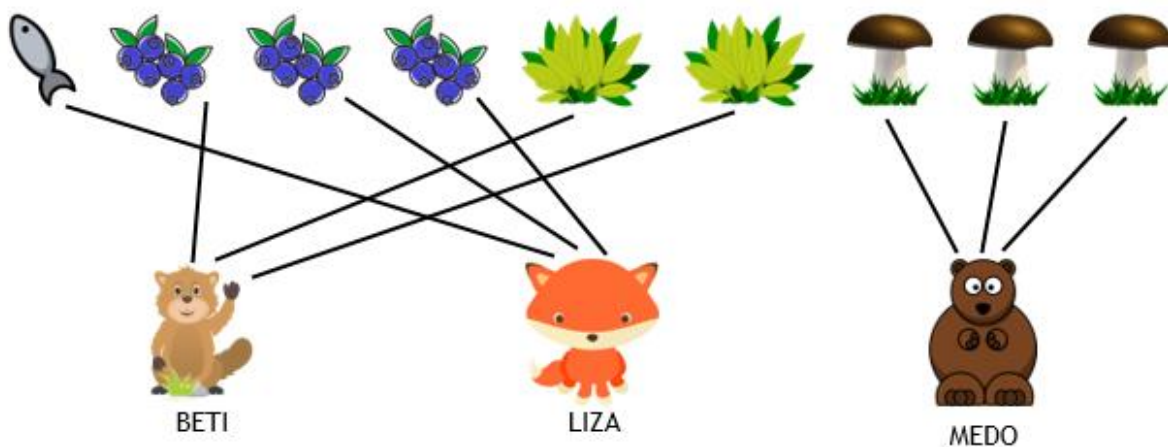


LIZA



MEDO

REŠITEV



GOBE JE SAMO MEDO, ZATO BO POJEDEL VSE TRI. KER BETI NE JE RIB, BO RIBO POJEDLA LIZA. KER LIZA NE JE LISTJA, BO POLEG RIBE POJEDLA ŠE DVE PORCIJI BOROVNIC. BETI BO POJEDLA VSE LISTJE IN BOROVNICE.

Računalniško ozadje

V tabeli naloge smo s kljukicami in križci pokazali, katero jed je posamezna žival. V računalništvu za ponazoritev, ali nekaj drži ali ne, uporabljamo logični vrednosti res (TRUE) ali ni res (FALSE).

Hamburger

3. razred



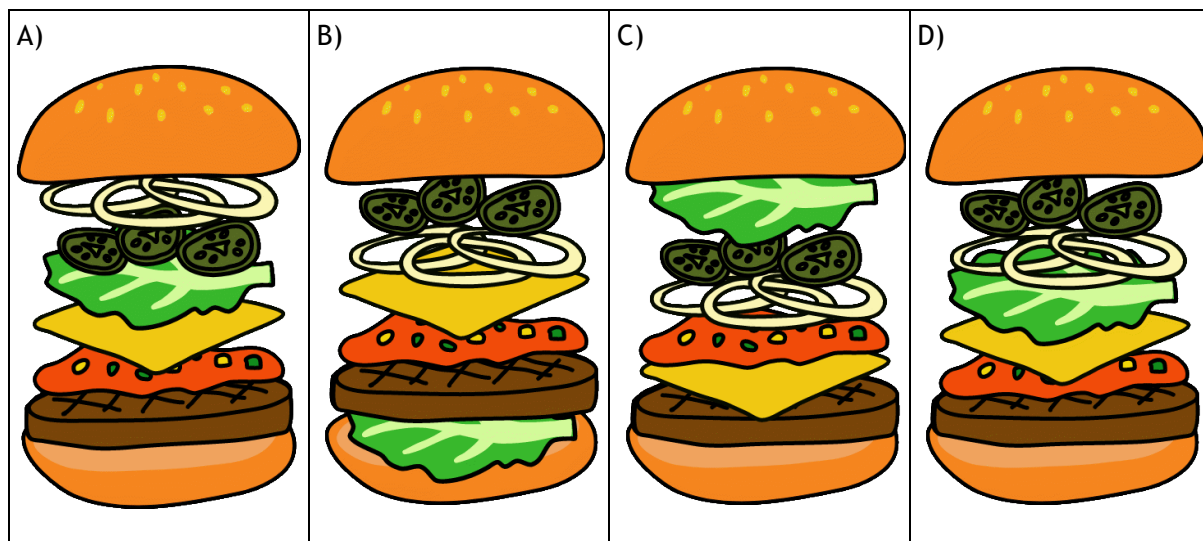
Janja pripravlja hamburgerje. Pri tem upošteva naslednja navodila:

Sestavine:

Kruh	Meso	Omaka	Kumarice	Solata	Čebula	Sir

1. Omaka mora biti na mesu.
2. Meso in sir morata biti pod kumaricami, solato in čebulo.
3. Čebula ne sme biti v stiku s kruhom.

Kateri hamburger je pripravljen po zgornjih navodilih?



Rešitev

Pri hamburgerju A je kruh v stiku s čebulo, zato krši pravilo 3.

Pri hamburgerju B je solata pod mesom in sirom, zato krši pravilo 2.

Pri hamburgerju C je omaka na siru, zato krši pravilo 1.

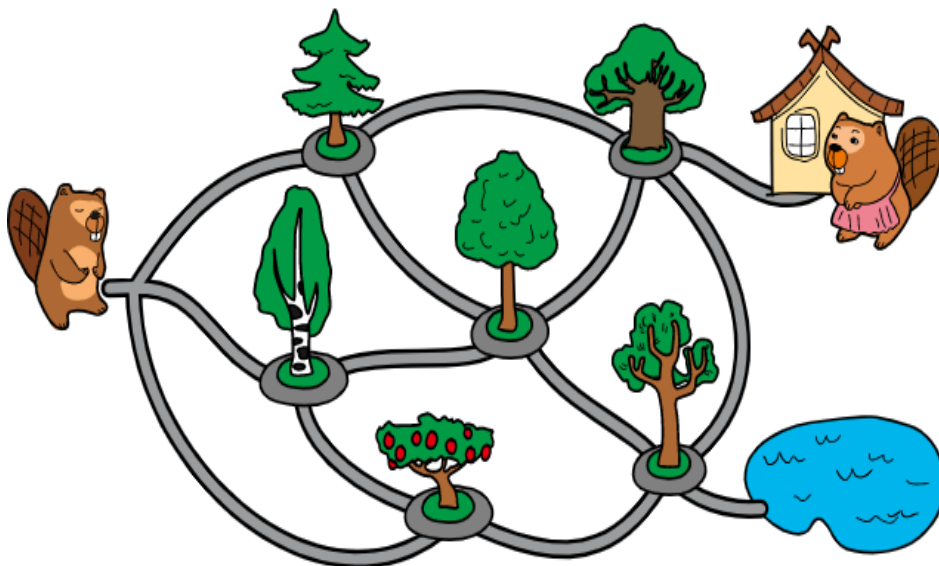
Hamburger D je sestavljen v skladu z vsemi pravili.

Računalniško ozadje

Pri reševanju te naloge smo morali preveriti, kateri hamburger upošteva vse omejitve, ki so zapisane v navodilih. Pri programiranju moramo preveriti, ali v programu upoštevamo vse omejitve, da program deluje pravilno.



Bober želi obiskati babico. Do babice lahko pride po več različnih poteh (slika).

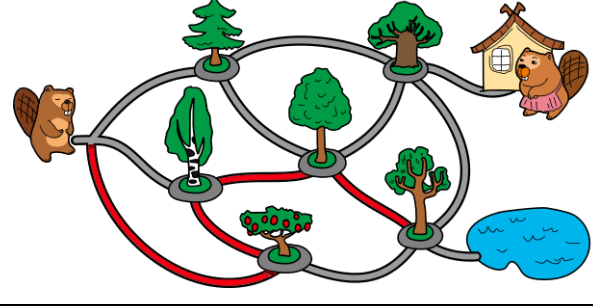

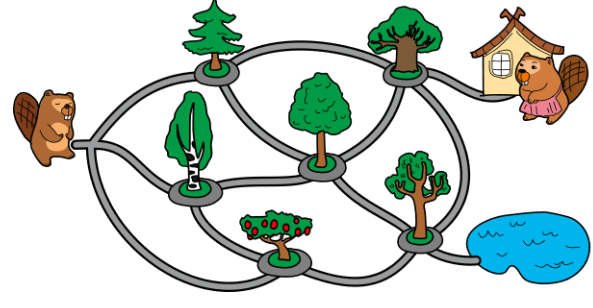
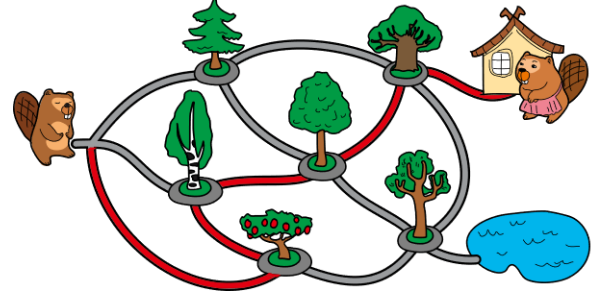


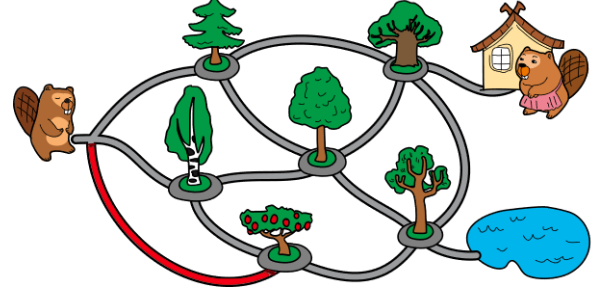


Katera od spodnjih poti pripelje bobra do babičine hiše?

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

Rešitev

Do babičine hiše vodi več poti. Od ponujenih poti je pravilna le C.







A	<p>Pot A ni ustrezna, saj vodi do jezera in ne do babičine hiše.</p> 
B	<p>Pot B je nemogoča, saj bober prvega drevesa (), ne more najprej obiskati.</p> 
C	<p>Pot C, ki je na sliki označena z rdečo, je pravilna, saj bobra pripelje do babičine hiše.</p> 
D	<p>Pri poti D bober lahko v prvem koraku pride do prvega drevesa (), vendar drugega drevesa v zaporedju ne more obiskati (), zato je pot nemogoča.</p> 

Računalniško ozadje

Načrtovanje premikov oziroma načrtovanje poti je problem, pri katerem moramo najti zaporedje veljavnih korakov, ki pripeljejo objekt od starta do cilja. Pri tej nalogi je bilo potrebno ugotoviti, kateri opis poti ustreza poti, ki bobra pripelje z izhodišča do babičine hiše.



Mali bobri Bobi, Čopi in Dane so se izgubili v smučarski vasi s petimi hišami. Vsak od njih se spomni nekaj lastnosti hiše, v kateri počitnikuje, kar smo zapisali v tabelo:

Bober	Število oken na hiši	Barva hiše	Barva vrat
Bobi	4	Zelena 	Črna 
Čopi	4	Ni rumena 	Rdeča 
Dane	2	Rumena 	Modra 

Pomagaj bobrom najti svoje počitniške hiše tako, da povežeš imena s pravo hiško.



Bobi

Čopi

Dane

Rešitev



Bobi počitnikuje v zeleni hiši s črnimi vrati.

Čopi počitnikuje v zeleni ali rdeči hiši. Ker ima štiri okna in rdeča vrata, je to torej zelena hiša z rdečimi vrati.

Dane počitnikuje v rumeni hiši z modrimi vrati.

Računalniško ozadje

V računalništvu pogosto preverjamo, kateri podatki ustrezajo določenemu pogoju. Podobno je bilo potrebno pri tej nalogi ugotoviti, katera hiša ustreza opisu, ki je podan v tabeli.

Rojstnodnevna zabava

3. in 4. razred



Bober si je za lažjo organizacijo zabave izdelal seznam opravil. Ugotovil je, da lahko nekatera opravila izvede le, če najprej opravi določena druga (spodnja tabela). Tako mora na primer bober pred izbiro lokacije najprej preveriti, koliko ljudi se bo zabave udeležilo.

OPRAVILO	Opravila, ki jih je potrebno opraviti prej
Preveri, koliko ljudi se bo udeležilo zabave.	
Kupi prigrizke.	
Določi datum.	NIČ
Oceni stroške.	
Izberi lokacijo.	

Obkroži črko pred zaporedjem, ki prikazuje vrstni red, v katerem bo bober izvedel opravila.

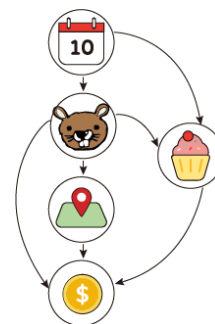
- A) → → → →
- B) → → → →
- C) → → → →
- D) → → → →

Rešitev

Pravilni odgovor je C.

Potek izvedbe opravil lahko predstavimo s sliko, kjer krogi predstavljajo posamezno opravilo, puščice pa povezave do opravil, ki sledijo le temu.

Odgovor A in B lahko takoj izločimo, saj mora bober najprej določiti datum in šele nato lahko preveri, koliko ljudi se bo udeležilo zabave. Odgovor D lahko ravno tako izločimo, saj bober ne more oceniti stroškov zabave, če pred tem ni seznanjen s številom ljudi, ki se bodo zabave udeležili.



Računalniško ozadje

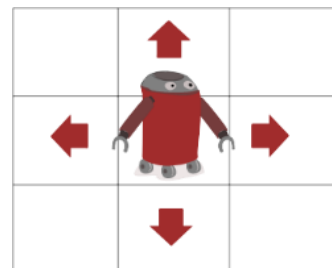
Za reševanje problemov pri programiranju pogosto probleme razdelimo na manjše podprobleme in jih rešimo. Rešitve podproblemov moramo nato, tako kot v tej nalogi, zložiti v pravilno zaporedje, da dobimo rešitev celotnega problema.

Robot Smetko 1

3. do 5. razred



Ana pospravlja po zabavi. Pomaga ji robot Smetko. Smetko se lahko premika le naprej, nazaj, levo in desno, kot kaže slika.



Smetko samodejno zazna najbližjo smet. Smetko deluje tako:

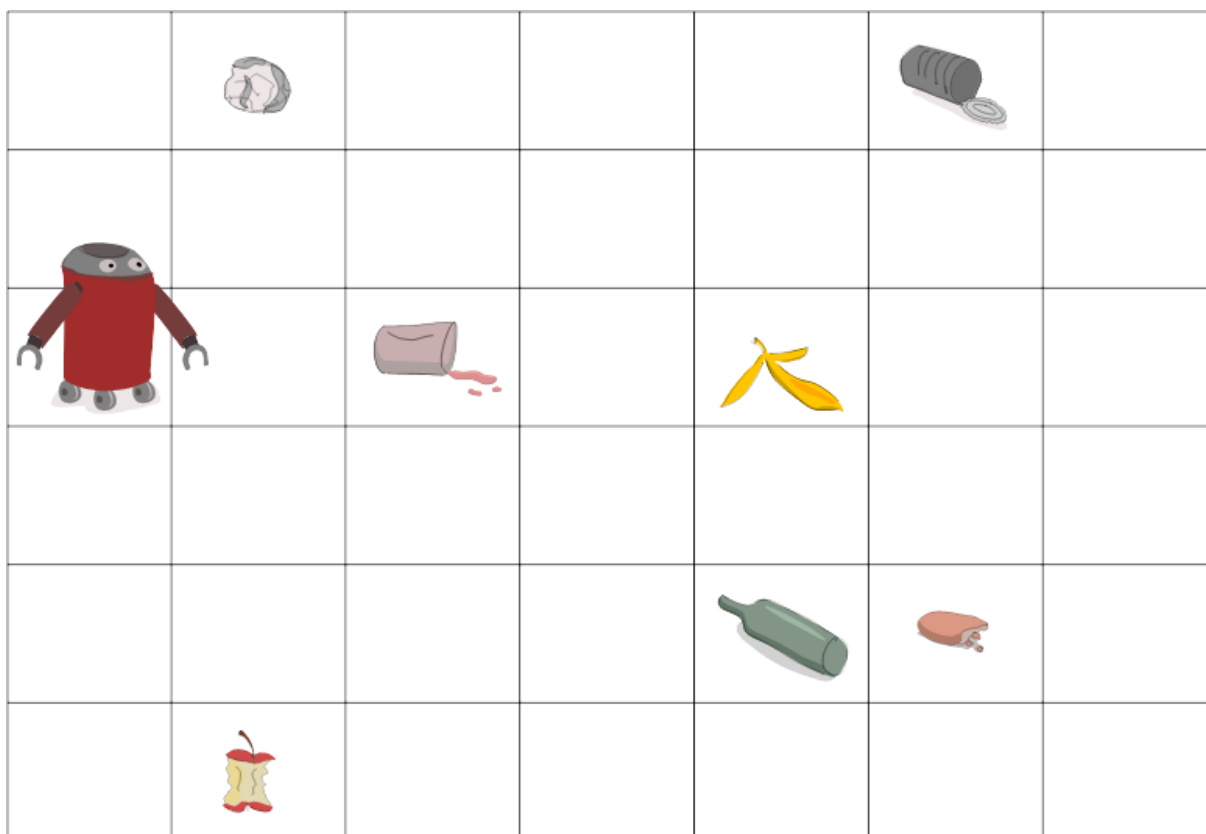
Korak 1: Zaznaj najbližjo smet.

Korak 2: Premakni se do najbližje smeti.

Korak 3: Poberi smet.

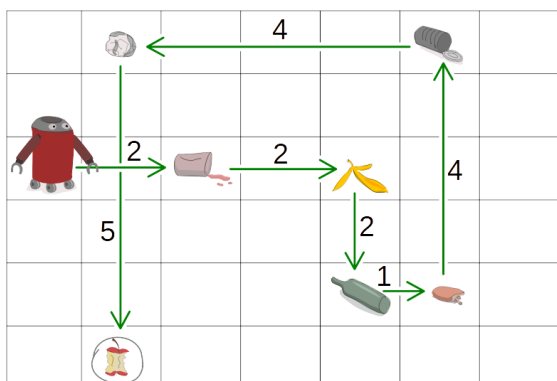
Korak 4: Če zaznaš še kakšno smet, pojdi na korak 1.

Ana postavi Smetka v razmetano sobo, kot kaže slika. Nariši pot, po kateri bo Smetko pobral vse smeti v sobi.



Rešitev

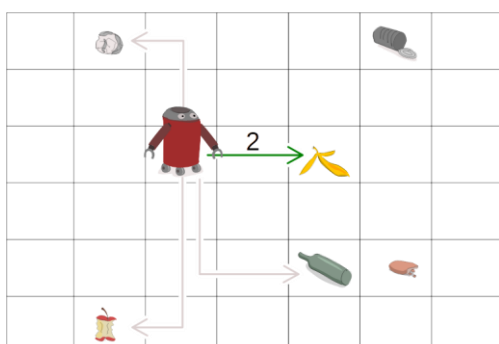
Zelene puščice na sliki prikazujejo pravilno rešitev, številke pa nam povejo, koliko premikov mora robot Smetko narediti do naslednje smeti.



Smetko v vsakem koraku poišče najbližjo smet od tam, kjer se trenutno nahaja.



Ko se premakne do smeti, jo pobere in ponovno poišče najbližjo smet z mesta, kjer se nahaja.



Ta postopek ponavlja, dokler ne pobere vseh smeti v sobi.

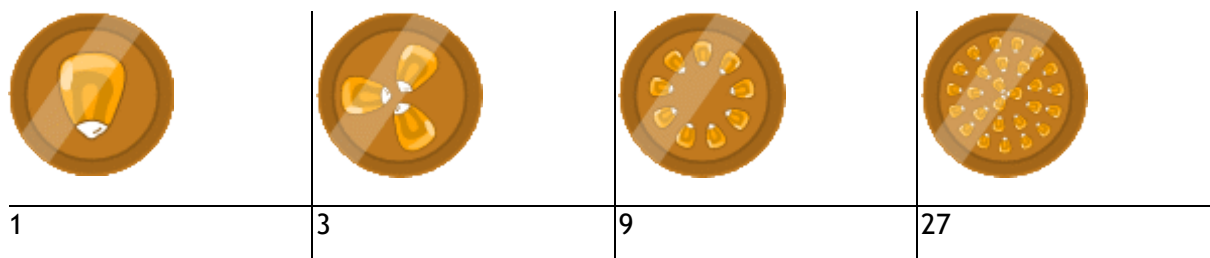
Računalniško ozadje

V računalništvu pogosto poskušamo najti najboljšo možno rešitev celotnega problema z reševanjem manjših podproblemov. Tako kot v nalogi robot Smetko vedno poišče najbližjo smet glede na to, kje se nahaja, da s tem kar se da učinkovito pospravi vse smeti v sobi.












Delija želi kupiti vrv, ki stane 21 bevrov.

Bevrski kovanci so lahko vredni 1, 3, 9 ali 27 bevrov. Danes imata Delija in prodajalec vsak le po en kovanec vsake od naštetih vrednosti.



Kako lahko Delija plača 21 bevrov za vrv?

- A) Plača z dvema kovancema za 9 bevrov   in enim za 3 bevre .
- B) Plača s kovancem za 27 bevrov  in kovancem za 3 bevre , prodajalec pa ji vrne kovanec za 9 bevrov .
- C) Plača s kovancem za 27 bevrov , prodajalec pa ji vrne kovanec za 3 bevre  in kovanec za 1 bevro .
- D) Ne more plačati točno 21 bevrov.

Rešitev

Pravilen odgovor je B, saj v tem primeru Delija plača $27 + 3 = 30$ bevrov, prodajalec pa ji vrne 9 bevrov. $30 - 9 = 21$ bevrov.

Odgovor A ni pravilen, saj Delija nima dveh kovancev za 9 bevrov.

Odgovor C ni pravilen, saj v tem primeru za vrv plača 23 bevrov.

Odgovor D ni pravilen, saj je v primeru B opisan možen način plačila točnega zneska.

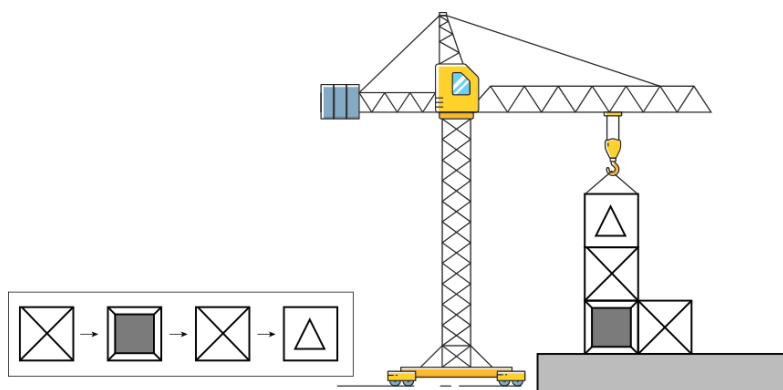
Računalniško ozadje

V računalništvu običajno uporabljamo dvojiški številki sestav, obstajajo pa tudi drugi. V tej nalogi je predstavljen primer trojiškega številskega sestava. Leta 1958 je Sergei Sobolev izdelal računalnik Setun, ki je za svoje delovanje uporabljal trojiški številski sestav.

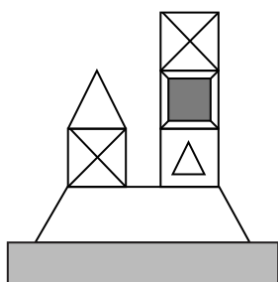


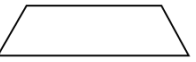





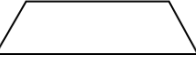





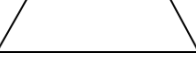





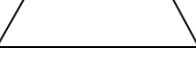





V Bobrlandu gradijo hiše iz kock po naslednjih navodilih: žerjav dvigne kocke eno po eno po predpisanem vrstnem redu in kocko odloži na temelje ali na vrh druge kocke.

Na primer, žerjav lahko, sledeč navodilom na sliki, zgradi naslednjo hišo:



Po katerih navodilih **ne more** zgraditi spodnje hiše?



- A)  →  →  →  →  → 
- B)  →  →  →  →  → 
- C)  →  →  →  →  → 
- D)  →  →  →  →  → 

Rešitev

Z navodili B ni mogoče zgraditi take hiše, saj bi lahko streho postavili le na kocko s trikotnikom ali pa na trapezno kocko, morala pa bi biti na kocki s križcem.

Računalniško ozadje

Računalniku podajamo navodila preko računalniških programov. Če navodila niso pravilno zapisana, jih računalnik ne more izvesti tako, kot smo si jih mi zamislili. V tem primeru moramo v programski kodi poiskati napako in jo popraviti. Podobno smo počeli v tej nalogi.

Robotska čebela 2

4. in 5. razred

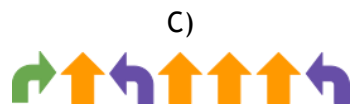
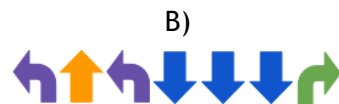
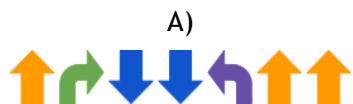


Robotska čebela pozna ukaze:

Naprej		Čebela gre eno polje naprej.
Nazaj		Čebela gre eno polje nazaj.
Desno		Čebela se na mestu obrne v desno.
Levo		Čebela se na mestu obrne v levo.

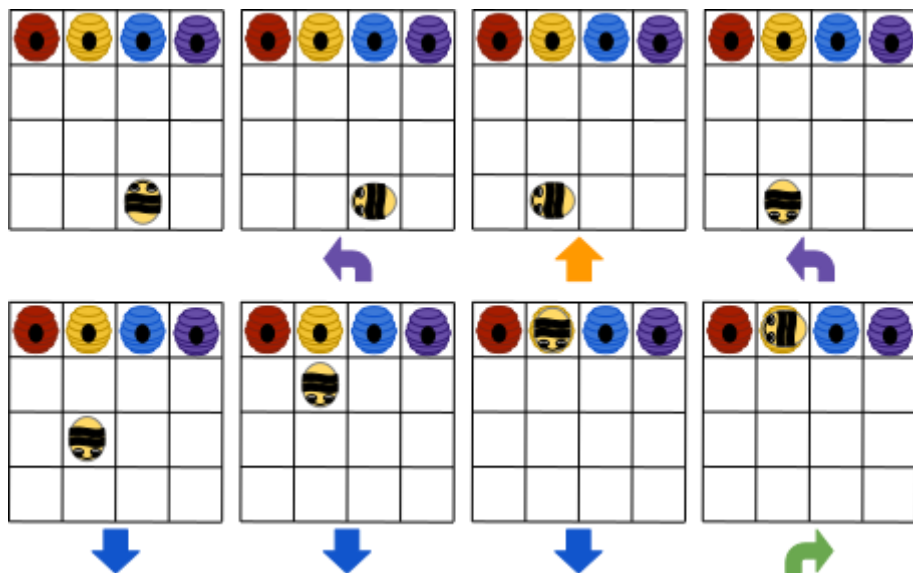
 Rdeča	 Rumena	 Modra	 Vijolična

Katero zaporedje ukazov bo robotsko čebelo pripeljalo do rumenega panja?



Rešitev

Čebelo bo do rumenega panja pripeljalo zaporedje ukazov B:



Zaporedje ukazov A bo robotsko čebelo pripeljalo do rdečega panja.

Zaporedje ukazov C bo robotsko čebelo pripeljalo do vijoličnega panja.

Zaporedje ukazov D bo robotsko čebelo pripeljalo do modrega panja.

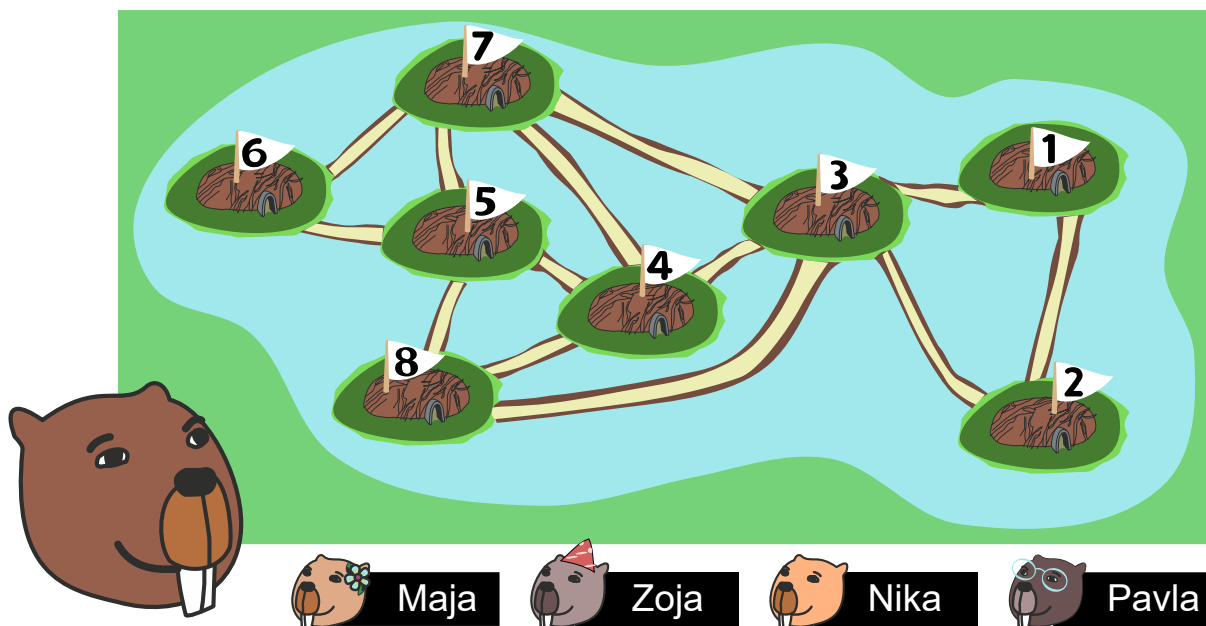
Računalniško ozadje

Programi, ki jih poganjamo na svojih računalnikih, so sestavljeni iz zaporedja ukazov, prav tako kot ti, s katerimi smo se igrali v tej nalogi, le da računalniški programi uporabljajo veliko več različnih ukazov in so veliko daljši.



Bober želi obiskati prijateljico Majo, vendar ne ve, kje živi. Na srečo ima zemljevid in nekaj podatkov. Dva bobra sta soseda, če je med njunima bobriščema zgrajena pot.

- Maja, Zoja in Pavla imajo vsaka po štiri sosede.
- Zoja in Pavla sta sosedi z Niko.
- Nika nima drugih sosedov.

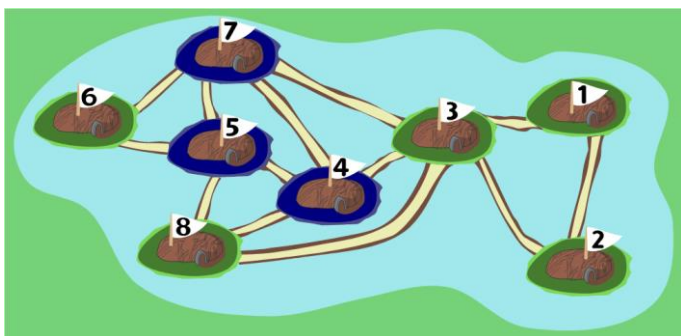


V katerem bobrišču živi Maja?

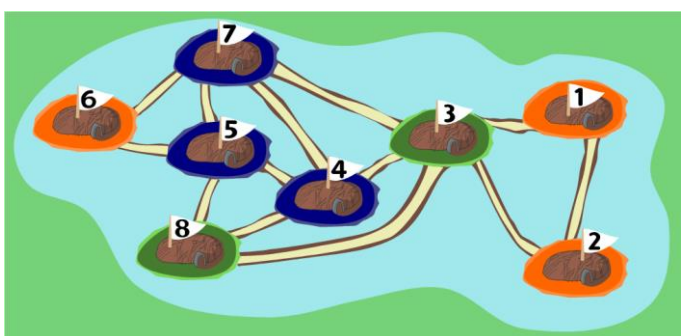
Rešitev

Maja živi v bobrišču 4.

Za rešitev te naloge je pomembno, da se osredotočimo na število poti do vsakega bobrišča. Vidimo lahko, da le do bobrišč 4, 5 in 7 vodijo po 4 poti, kar pomeni, da imajo bobri, ki živijo v njih, po 4 sosede. To hkrati pomeni, da Maja, Zoja in Pavla živijo vsaka v enem od teh treh bobrišč.



Vemo tudi, da Nika živi v bobrišču, do katerega vodita le 2 poti. Taka bobrišča so tri: 1, 2 in 6 (na sliki so označena z oranžno barvo).



Iz besedila lahko razberemo, da ima Nika za sosedi Pavlo in Zojo, kar pomeni, da Nika živi v bobrišču 6. V bobrišču 5 in 7 tako živita Pavla in Zoja. Edino preostalo bobrišče s štirimi sosedi je bobrišče 4, kar pomeni, da Maja živi v njem.

Računalniško ozadje

Bobrišča s povezavami predstavljajo graf. Računalničarji se pogosto srečujejo z grafi in jih uporabljajo za reševanje različnih nalog, na primer za postavitve internetnega omrežja.



Profesor Izzivalnik je pred svojim skrivnostnim izginotjem uspel dešifrirati starodavne rune. Edina sled za njim je njegov zvezek, v katerem so našli zapis na desni:

A	B	C	Č	D	E
F	G	H	I	J	K
L	M	N	O	P	R
S	Š	T	U	V	Z
Ž					

I =

P =

Kateri od naslednjih zapisov se prevede v BAV?

A)

B)

C)

D)

E)

Rešitev

Število paličic na levi strani predstavlja številko vrste, v kateri se nahaja posamezna črka, število paličic na desni strani pa številko stolpca, v katerem se nahaja ta črka.

Pri zgornjih primerih zapise prevedemo kot:

A)	B)	C)	D)	E)
Č P G	H P G	C A V	B A V	T A V

Pravilen je torej odgovor D.

Računalniško ozadje

Problem je primer kodiranega sporočila. Računalniki so v preteklosti igrali pomembno vlogo pri dekodiranju različnih sporočil.

Vijaki z matico

4. do 9. razred



Branko dela v tovarni na proizvodni liniji vijakov z matico.



Njegovo delo je naslednje:

- Branko stoji na koncu dolgega tekočega traku, na katerem je vrsta matic in vijakov.
- Branko vzame vsak element (ali matico ali vijak) s tekočega traku.
- Če s traku vzame matico, jo postavi v vedro, ki ga ima na tleh.
- Če s traku vzame vijak, pobere eno matico iz vedra, privije matico na vijak ter tako sestavlja del odloži v veliko škatlo.



Vendar pa gredo pri delu lahko stvari tudi narobe, in sicer v dveh primerih:

- Branko s traku vzame vijak, v vedru pa ni nobene matice.
- Na tekočem traku ni več matic in vijakov, v vedru pa so ostale še matice.

Katero zaporedje matic  in vijakov , če jih obdelujemo z leve proti desni, **ne bo** povzročilo, da gredo stvari narobe?

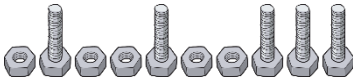
A)



B)

C)

D)


Rešitev

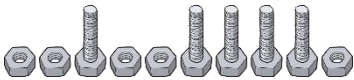
Pravilen odgovor je C. 

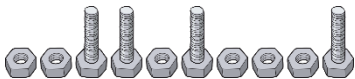
V pravilnost rešitve se lahko prepričamo tako, da spremljamo stanje vedra in tekočega traku (od leve proti desni). S črko M smo označili matice , s črko V pa vijake :

Vedro	Tekoči trak
prazno	M V M M V M M V V V
M	V M M V M M V V V
prazno	M M V M M V V V
M	M V M M V V V
M M	V M M V V V
M	M M V V V
M M	M V V V
M M M	V V V
M M	V V
M	V
prazno	prazno

Preverimo še ostale odgovore.

Pri odgovoru A  bo nastal problem pri tretjem elementu (M V V), saj za drugi vijak nimamo nobene matice v vedru.

Odgovor B  povzroči problem pri devetem elementu (M M V M M V V V V), ker v vedru zmanjka matic za peti vijak na traku (pred tem vijakom so na traku le 4 matice).

Pri odgovoru D  pa pride do problema, ko obdelamo celo zaporedje, saj nam v vedru ostaneta dve matici. Na traku imamo namreč le 4 vijake in kar 6 matic.

Računalniško ozadje

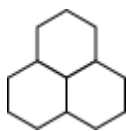
Tekoči trak si lahko predstavljamo kot spremenljivko tipa seznam, ki vsebuje podatke dveh tipov, in sicer tipa matica in tipa vijak. Vedro si prav tako lahko predstavljamo kot seznam.

Bober izvaja program, ki je napisan nekako takole: dokler seznam ni prazen, preberi vhodni podatek, če je ta podatek matica, ga dodaj v spremenljivko vedro, če je podatek vijak, preveri, ali imaš v vedru podatek matica. Če ga imaš, vzemi iz vedra podatek matica, sicer končaj program in javi napako.

Ko je seznam prazen, preveri, če je prazno tudi vedro in če ni, javi napako. Program se uspešno zaključi, če sta tako tekoči trak kot tudi vedro prazna.

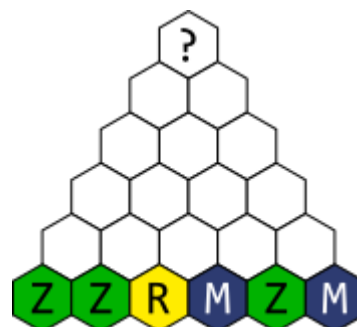
Barvna sestavljanke

4. do 9. razred, srednja šola



Samo ima šestkotne koščke sestavljanke v treh barvah. Za vsake tri koščke, ki jih sestavi skupaj, kot kaže leva slika, želi, da so ali vsi iste barve ali pa vsak druge barve.

Samo sestavi koščke v stolp na desni sliki. Barve koščkov sestavljanke v prvi vrstici so že določene. Kakšne barve je zgornji košček sestavljanke?



A)



B)



C)



D)

Možni sta dve barvi.

D)

Lahko je poljubne barve.

Rešitev

Zgornji košček sestavljanke je rumene barve. Pravilen odgovor je A.

Če poznamo barvi dveh sosednjih koščkov, poznamo tudi barvo tretjega koščka. Poglejmo na primer spodnji levi košček neznane barve. Ker sta koščka pod njim oba zelena, mora biti tudi ta košček zelen. Če pogledamo pa košček desno od njega, ugotovimo, da mora biti moder, saj sta koščka pod njim zelene in rumene barve. Tako nadaljujemo z ugotavljanjem barve koščkov od spodaj navzgor, dokler ne pridemo do vrhnjega koščka. Rešitev je prikazana na sliki.



Pri tem ni pomembno, v kakšnem vrstnem redu obravnavamo koščke v vsaki vrstici, vedno dobimo enak rezultat.

Računalniško ozadje

V nalogi smo prišli do rešitve tako, da smo najprej ugotovili, katerega od obeh primerov (dva koščka sta iste barve ali različnih barv) moramo upoštevati. Nato smo to ponavljali, dokler nismo ugotovili vseh barv v stolpu. Tako zaporedje korakov sestavlja *algoritem*, ki ga uporabimo za rešitev problema. Če bi napisali računalniški program, ki *izvede* ta algoritem, bi bila identifikacija primera in izvedba ustrezne akcije za vsakega od primerov lahko zapisana v *pogojnem stavku*. Ponavljanje teh korakov pa bi naredili z *zanko*.

Zaporedje kamenčkov

5. do 7. razred



V tovarni kamenčkov imajo dve različni napravi. Ko na levi strani v napravo vstavimo zaporedje kamenčkov, naprava odstrani nekaj kamenčkov in na desni strani vrne ostale.

Modra naprava se imenuje »liha naprava«. Naprava odstrani vsak kamenček, ki se v zaporedju nahaja na lihem mestu:



Rjava naprava, na kateri je številka »x«, se imenuje »krajšalna naprava«. Naprava odstrani prvih »x« in zadnjih »x« kamenčkov v zaporedju ter vrne preostale. Spodnja slika prikazuje, kako deluje krajšalna naprava, če je na njej nastavljena številka 3:



Zaporedje kamenčkov pošljemo skozi liho napravo in nato skozi krajšalno napravo, na kateri je nastavljena številka 2, kot kaže slika.



Kakšno zaporedje bomo dobili na izhodu iz druge naprave?

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

Rešitev

Pravilen odgovor je B.

Kamenčki so prvotno razvrščeni v naslednjem zaporedju:



Najprej grejo skozi liho napravo, ki vrne naslednje zaporedje kamenčkov:



Ko te vstavimo v krajšalno napravo, na kateri je nastavljena številka 2, ta naprava vrne naslednje zaporedje kamenčkov:



Računalniško ozadje

Vsaka od naprav predstavlja neke vrste funkcijo, kakršne sestavljamo pri programiranju. Na primer sestavimo lahko funkcijo, ki iz seznama izbriše podatke na lihih mestih, in funkcijo, ki iz seznama odstrani prvih n in zadnjih n podatkov. Podobne funkcije lahko napišemo tudi za nize.

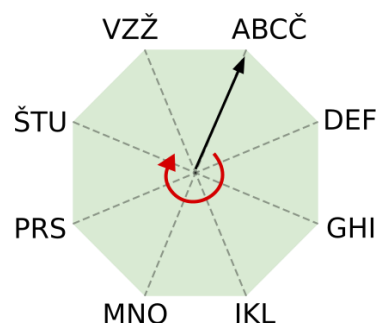


Bobri za šifriranje sporočil uporabljajo poseben pripomoček v obliki osemkotnika, ki ima na ogliščih napisane po tri ali štiri črke. Iz središča osemkotnika na skupine črk kaže kazalec, ki ga lahko vrtimo v smeri urinega kazalca.

Na začetku šifriranja novega sporočila kazalec vedno kaže na črke ABCČ.

Sporočila šifriramo na naslednji način:

- Prva številka pove, za koliko oglišč naprej od trenutnega položaja moramo v osemkotniku zavrteti kazalec.
- Druga številka pove, na katerem mestu v skupini črk je črka, ki jo iščemo.
- Zašifrirane črke med seboj ločimo z znakom -.



Na primer, sporočilo MUHA je zakodirano kot 41-23-42-61.

Kako je zašifrirano sporočilo HRANA?

- A) 22-32-31-42-41
- B) 02-43-42-12-42
- C) 22-32-53-62-23
- D) 32-22-31-42-71

Rešitev

Šifra A: 22-32-31-42-41 pomeni HRANA.

Šifra B: 02-43-42-12-42 pomeni BOBER.

Šifra C: 22-32-53-62-23 pomeni HRIBI.

Šifra D 32-22-31-42-71 pomeni KRANJ.

Računalniško ozadje

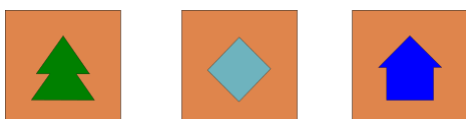
Računalniki se pogosto uporabljajo za različno kodiranje in šifriranje podatkov. Najmočnejši šifrirani algoritmi se uporabljajo za prenos zaupnih informacij med računalniki.



Marija je našla zaklenjeno škatlo z zakladom. Za odklepanje škatle mora poiskati pravilno zaporedje treh znakov. Z uporabo naslednjih namigov pomagaj Mariji odkleniti škatlo z zakladom.



1. En znak je pravilen in na pravem mestu.



2. Ni pravilnega znaka.



3. Dva znaka sta pravilna, ampak na napačnih mestih.



4. En znak je pravilen, ampak na napačnem mestu.






5. En znak je pravilen, ampak na napačnem mestu.

Katero od naslednjih zaporedij znakov bo odklenilo škatlo z zakladom?



Rešitev

Pravilen odgovor je B.



Najprej bomo izločili znake, ki ne sodijo v rešitev. V drugi vrsti izvemo, da smreka , karo  in hiška  ne sodijo v rešitev.

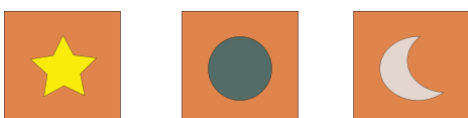
V 5. vrstici izvemo, da bo v rešitvi zvezda , ki je lahko na 1. ali 2. mestu zaporedja. Iz 3. vrstice izvemo, da zvezda ni na 2. mestu, zato je lahko na 1.



Ker vemo, da je na 1. mestu rešitve zvezda, lahko iz 1. vrste razberemo, da je v rešitvi bodisi hiška na 2. mestu ali luna na 3. mestu. Ker iz 2. vrstice vemo, da hiške v rešitvi ni, mora biti luna na 3. mestu.



Iz 3. vrste izvemo, da roza trikotnika ni v rešitvi. Iz 4. vrste izvemo, da sta lahko na praznem mestu le siv krogec  ali rdeč srček . Za 4. vrsto velja tudi, da je pravilni znak na napačnem mestu, to ne more biti srček. Pravilna rešitev je zato: zvezda, krogec, luna.

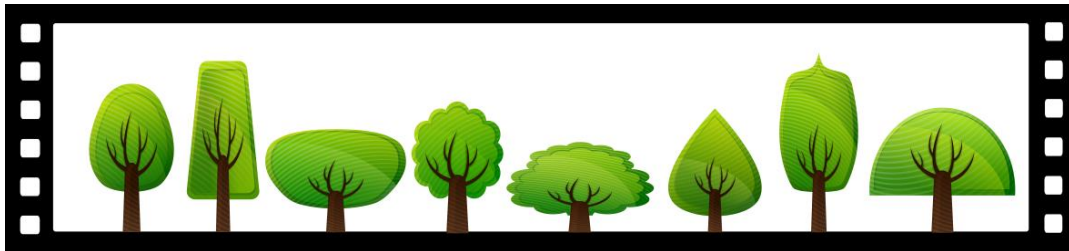


Računalniško ozadje

Morda se vam zdi, da je naloga bolj logična kot računalniška, vendar je tudi logika zelo pomembna za računalništvo.



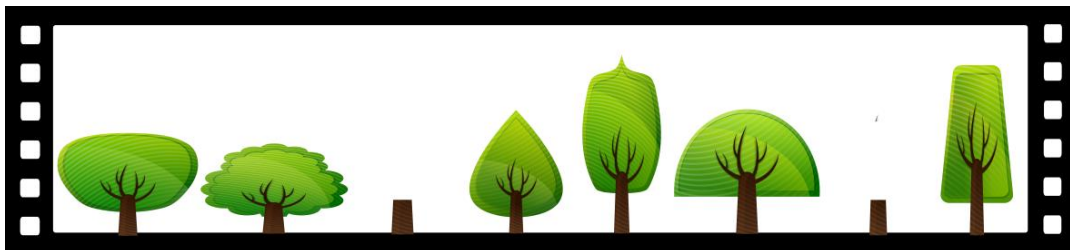
Darija je v gozdu posnela 360 stopinjno sliko osmih dreves.



Čez nekaj dni se je vrnila na isto mesto v gozdu in ponovno posnela sliko. Ugotovila je, da so medtem dve drevesi požagali.

Katero sliko je drugič posnela Darija?

A)



B)



C)



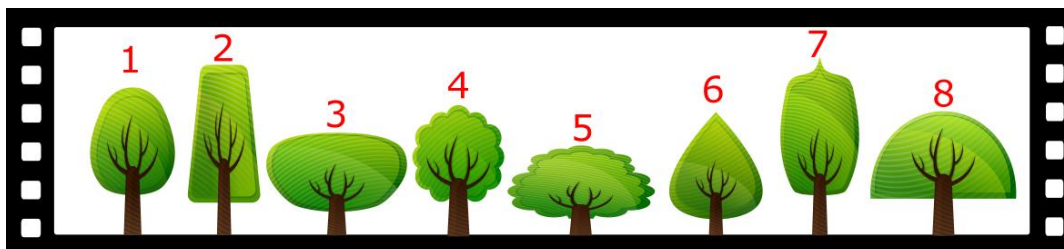
D)



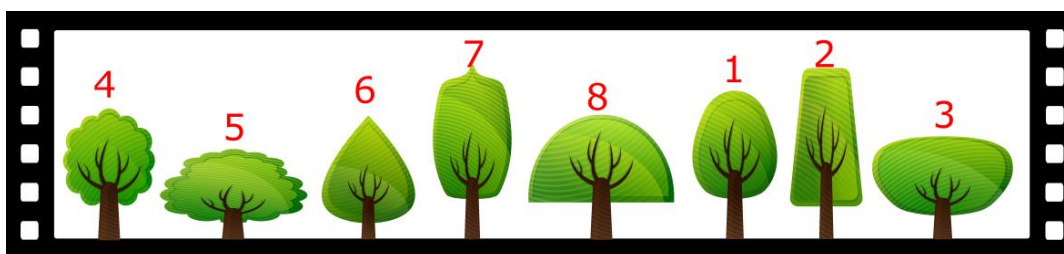
Rešitev

Pravilen odgovor je C.

Drevesa na prvi sliki oštevilčimo in dobimo:



Če nekoliko spremenimo zorni kot in posnamemo drugo sliko, bodo drevesa še vedno v istem vrstnem redu, le nekoliko zamaknjena. Tudi drevesa na drugi sliki oštevilčimo tako, da ima vsako drevo isto številko na obeh slikah:

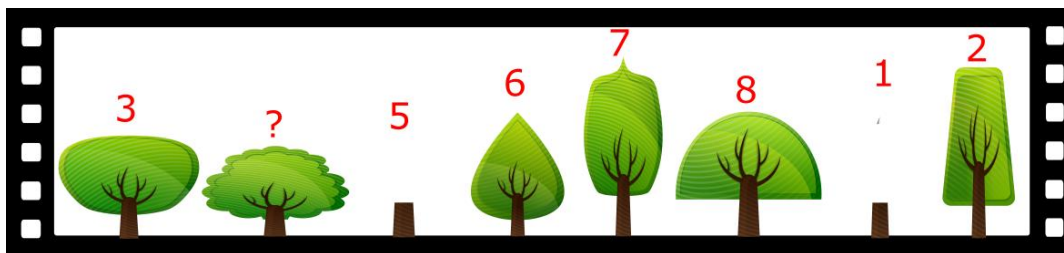


Tako lahko na vsaki sliki v podanih možnih odgovorih drevesa oštevilčimo in preverimo, če so drevesa še vedno v pravem zaporedju.

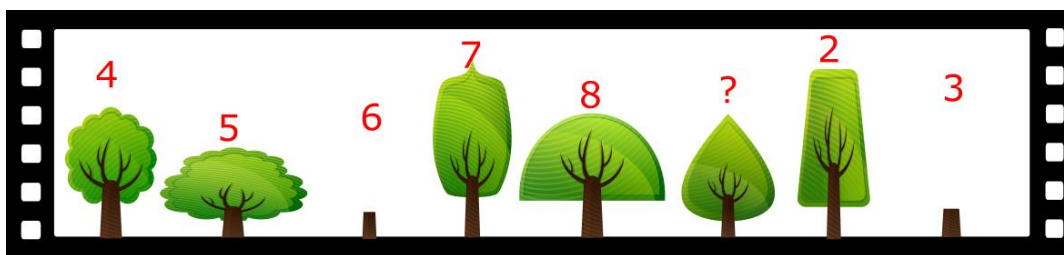
Slika pri odgovoru C se ujema z našo drugo oštevilčeno sliko, manjkata pa drevesi 4 in 3, ki sta bili posekani. Vsa ostala drevesa so v pravilnem vrstnem redu.



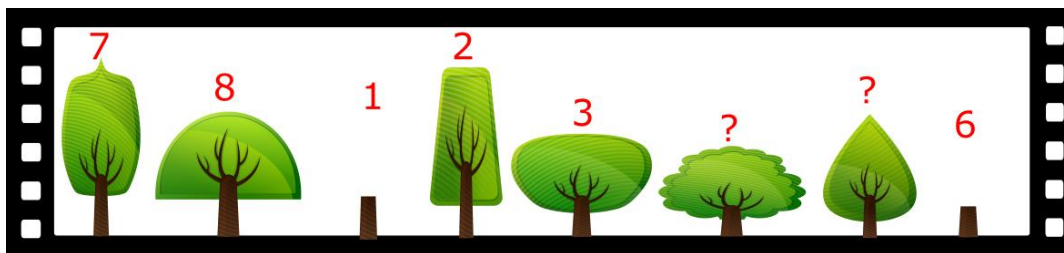
Slika pri odgovoru A ima posekani drevesi 1 in 5. Vendar je drevo 4 druge oblike kot na prvi posneti sliki, kar je narobe. Zato te slike Darija ni mogla posneti.



Slika pri odgovoru B ima posekani drevesi 6 in 3. A je drevo 1 drugačne oblike kot na prvi posneti sliki, kar je narobe. Tudi te slike Darija ni mogla posneti.



Slika pri odgovoru D pa ima posekani drevesi 1 in 6. Vendar sta tu drevesi 4 in 5 drugačne oblike kot na prvi posneti sliki. Tudi to je narobe in te slike Darija ni mogla posneti.



Računalniško ozadje

Naloga zahteva iskanje vzorca, ki mu sledijo drevesa, in ta vzorec uporabiti na možnih odgovorih. Zmožnost prepoznavanja vzorca nam omogoča, da najdemo napake v zaporedju. V tej nalogi je torej pomembno zaporedje dreves. Ko smo pregledovali možne odgovore, smo iskali razlike v zaporedju dreves, torej napake. Proces iskanja napak/hroščev (pri programiranju) imenujemo razhroščevanje. Kadar določene informacije manjkajo (kot so posekana drevesa v naši nalogi), je iskanje napak težje.

Preproga

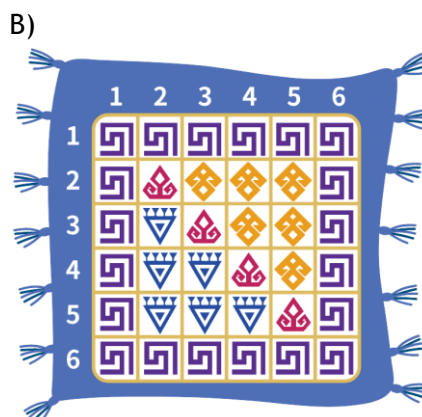
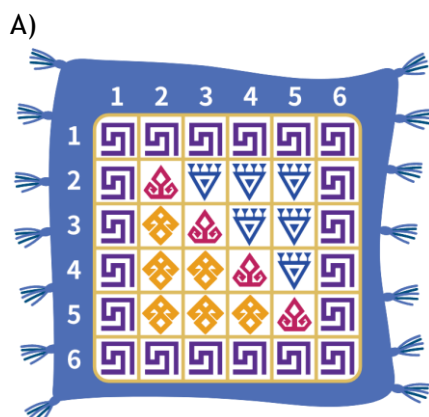
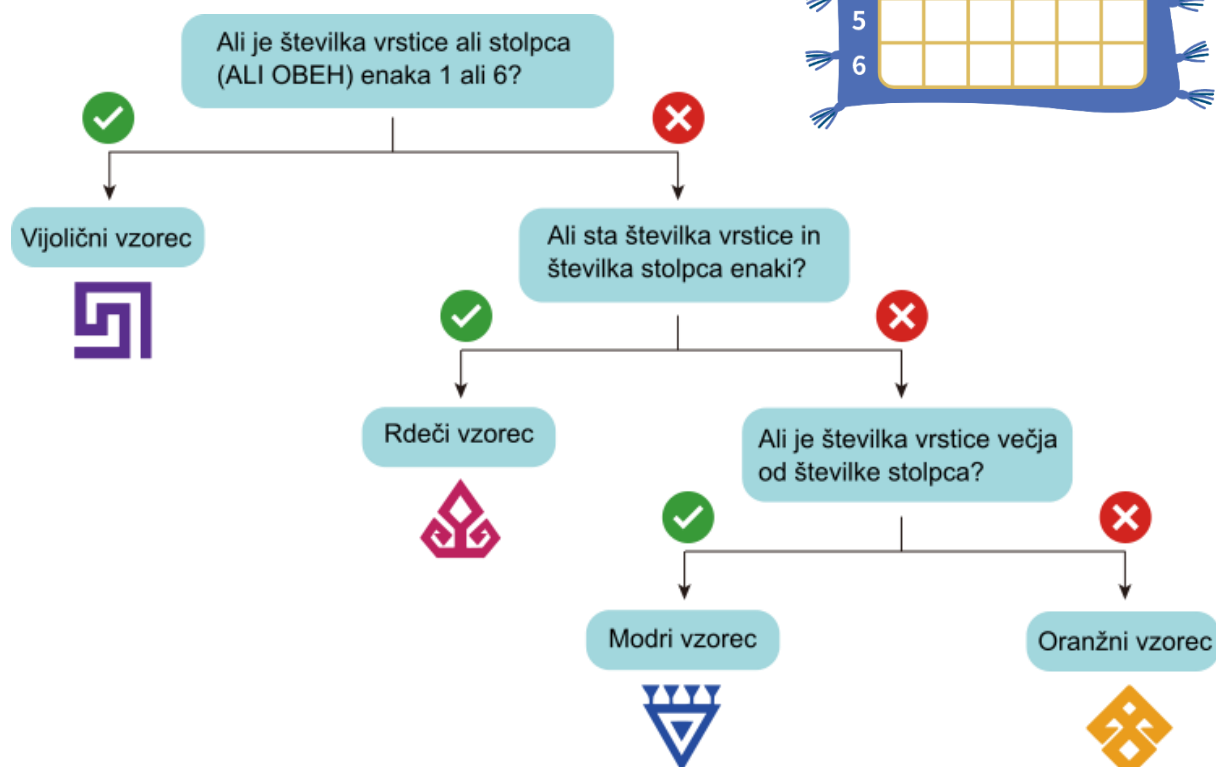
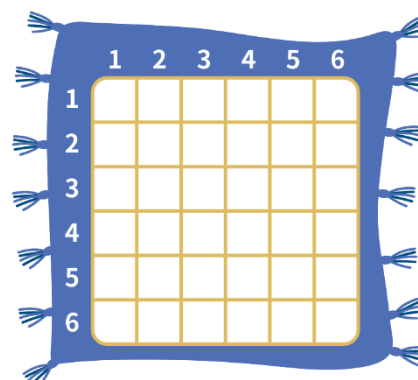
6. in 7. razred



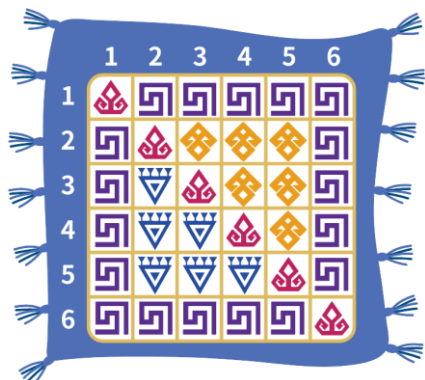
Hale tke preproge. Izdeluje kvadratno preprogo s 6 vrsticami in 6 stolpci.

Hale v vsak kvadrateg na preprogi izveze simbol. Pri tem sledi naslednjim navodilom:

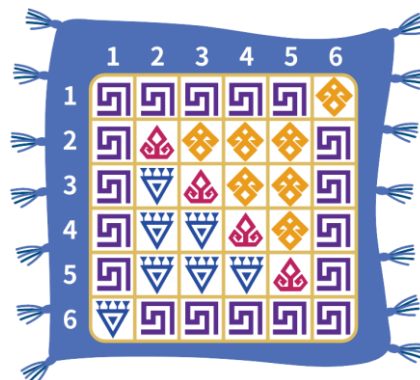
Kako bo izgledala preproga?



C)



D)



Rešitev

V prvi vejitvi izvemo, da so vsi zunanji kvadrati na preprogi zapolnjeni z vijoličnim vzorcem, zato sta možni rešitvi le A in B.

Po diagonali od 2. do 5. vrstice so kvadrati zapolnjeni z rdečim vzorcem, kar velja tako za A kot tudi za B.

Za preostale kvadratke velja, da če je številka vrstice večja od številke stolpca, Hale nanje všije modri vzorec, sicer pa oranžnega. To drži za preprogo B.

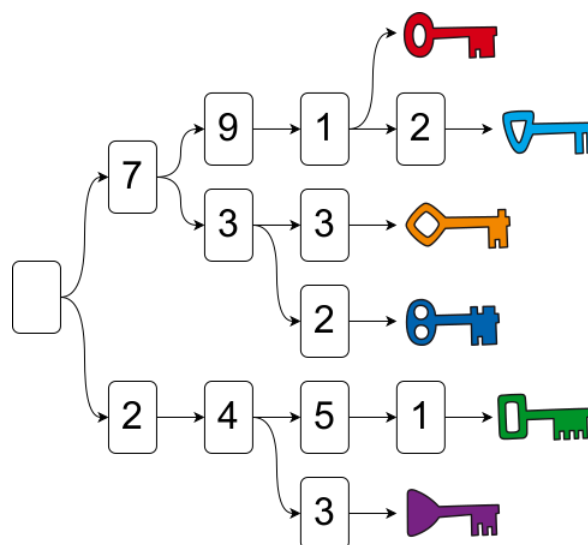
Računalniško ozadje

Pri reševanju računalniških problemov si marsikdaj pomagamo z odločitvenimi drevesi, kakršno je prikazano v tej nalogi. Pri vsaki odločitvi pride do vejitve.



Ekipe tekmovalcev je odšla na bazen na tekmo podvodnega hokeja. Vsak je svoje stvari spravil v garderobno omarico, si zapisal številko omarice na listek papirja in ga skupaj s ključkom od omarice predal v hrambo trenerju. Trener si je številke garderobnih omaric in pripadajoče ključke zabeležil, kot kaže slika.

Številko garderobne omarice je razdelil na številke, med njimi narisal povezavo in jih povezal s pripadajočim ključkom omarice. Na primer, za omarico številka 243 je zapisal številke 2, 4 in 3 po vrsti, jih povezal med seboj in še z vijoličnim ključem (glej spodnji del slike).



Po tekmi so ugotovili, da je voda poškropila listke s številkami omaric in razmazala nekatere številke.

Če vemo, da so ključki za garderobne omarice s številkami 73X, 24XX in 7X1X urejeni po vrsti, kateri od naslednjih kompletov ključkov je pravi za te omarice?

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

Rešitev

Pravilen odgovor je A.

Številka tretje omarice ima štiri številke in se začne s 7. Torej je zanjo lahko le ključek . Podobno za drugo omarico ugotovimo, da je zanjo lahko le ključek , saj se številka začne z 2 in ima 4 številke. Ključek od prve omarice pa je lahko ali (733) ali (732), saj se oba začneta s 73.

Torej je pravilen odgovor A: . Prav tako bi bil pravilen odgovor tudi , vendar ta ni naveden med možnimi odgovori.

Odgovor B je napačen, ker ustreza trimestni številki omarice. Odgovor C je tudi napačen, saj se številka omarice s ključkom začne s številko 2 in ne s 7. Odgovor D pa je napačen, ker ima številka omarice s ključkom drugo številko enako 9 namesto 3.

Računalniško ozadje

Graf, ki ga je sestavil trener v tej nalogi, predstavlja *številsko drevo* ali *trie* (ime izhaja iz angleške besede *retrieval*, kar pomeni pridobitev oz. povrnitev). To je drevesna podatkovna struktura, ki hrani pare ključ-vrednost, podobno kot razpršilna tabela. Številsko drevo predstavimo kot usmerjen graf, kjer je vsako vozlišče (razen začetnega) označeno s črko. Vrednost dobimo tako, da sledimo povezavam v grafu črko po črko, enako kot v primeru v nalogi.

Številsko drevo omogoča iskanje z uporabo prefiksa: pri iskanju podamo le začetek besede in najdemo vse besede, ki se ujemajo s podanim začetkom. Tako iskanje je zelo uporabno pri nalogah samodejnega dokončanja.

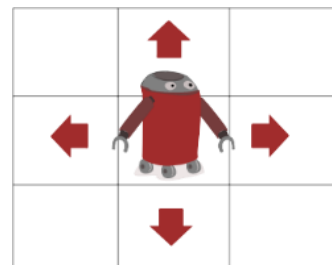
Robot Smetko 2

6. do 9. razred



Ana pospravlja po zabavi. Pomaga ji robot Smetko. Smetko se lahko premika le naprej, nazaj, levo in desno, kot kaže slika.

Premik z ene ploščice na drugo šteje kot en premik. Smetko samodejno zazna najbližjo smet. Smetko deluje tako:



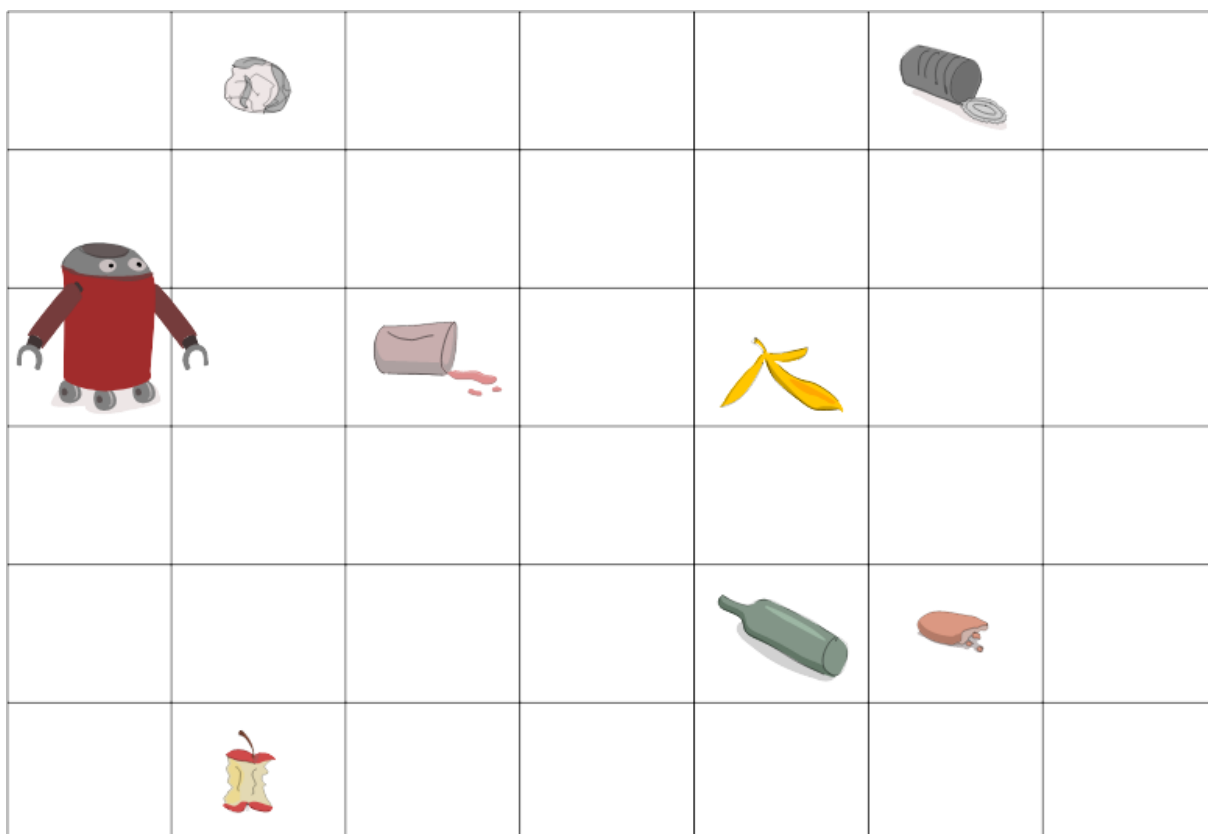
Korak 1: Zaznaj najbližjo smet.

Korak 2: Premakni se do najbližje smeti.

Korak 3: Poberi smet.

Korak 4: Če še nisi pobral vseh smeti, pojdi na korak 1.

Ana postavi Smetka v razmetano sobo, kot kaže slika.

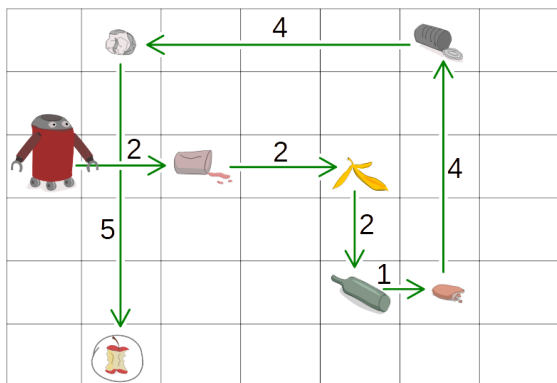


Katero smet bo robot Smetko pobral nazadnje?

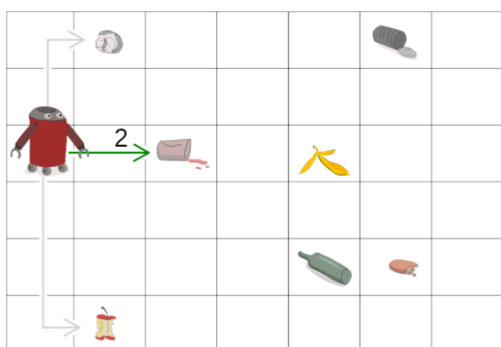


Rešitev

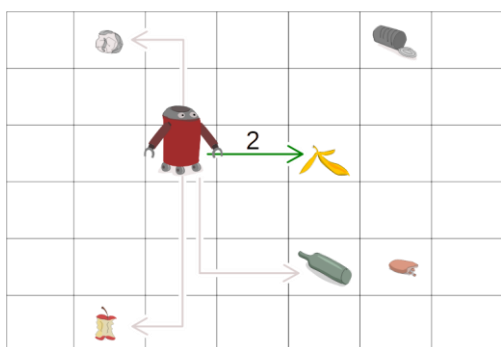
Robot Smetko bo nazadnje pobral jabolčni ogrizek. Njegova pot čiščenja je prikazana na sliki: zelene puščice prikazujejo premike robota, številke pa nam povejo, koliko premikov mora narediti do naslednje smeti.



Smetko v vsakem koraku poišče najbližjo smet od tam, kjer se trenutno nahaja.



Ko se premakne do smeti, jo pobere in ponovno poišče najbližjo smet z mesta, kjer se nahaja.



Ta postopek ponavlja, dokler ne pobere vseh smeti v sobi.

Računalniško ozadje

V računalništvu pogosto poskušamo najti najboljšo možno rešitev celotnega problema z reševanjem manjših podproblemov. Tako kot v nalogi robot Smetko vedno poišče najbližjo smet glede na to, kje se nahaja, da s tem kar se da učinkovito pospravi vse smeti v sobi.



Monika in Veronika imata vsaka svojo ogrlico iz perl.

Monika



Veronika

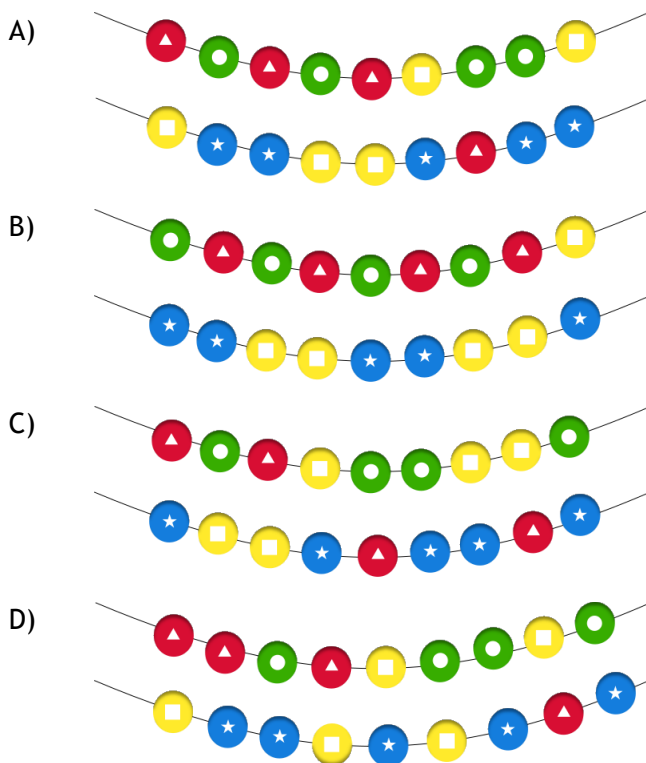


Odločita se, da bosta ogrlici spremenili na naslednji način:

1. Vzemi eno perlo z desne strani svoje ogrlice.
2. Če je ta perla rumena s kvadratom ali rdeča s trikotnikom, jo dodaj na levo stran druge ogrlice. Sicer pa jo dodaj na levo stran svoje ogrlice.

To ponavljata izmenično: vedno je najprej na potezi Monika, nato Veronika. Menjavanje se konča, ko je vsaka od njiju dala drugi že vsaj tri perle.

Kakšni sta ogrlici Monike in Veronike po izvedenem postopku izmenjave perl?



Rešitev

Pravilen odgovor je A.

Odgovor B ni pravilen, ker si, glede na naveden postopek, izmenjata le po tri perle. Ogrlici pri tem odgovoru pa imata izmenjane po štiri perle.

Odgovor C tudi ni pravilen, ker se koraki še niso končali. Izmenjani sta le po dve perli (čeprav smo z desne strani vsake ogrlice vzeli po tri perle).

Pri odgovoru D so izmenjave po tri perle, vendar vrstni red perl ni pravilen. Pri izmenjavi perl pri vsaki ogrlici odstranimo po pet perl (da dobimo tri perle, ki jih izmenjamo), torej morajo leve štiri perle na vsaki od prvotnih ogrlic ostati nedotaknjene in v prvotnem vrstnem redu. Pri Monikini ogrlici skrajno desna perla (zelena s krogom) nove ogrlice ni na pravem mestu, pri Veronikini ogrlici pa manjka modra perla z zvezdico na skrajni desni strani nove ogrlice. Poleg tega tudi premeščene perle na levi strani niso v pravilnem zaporedju.

Pravilnost odgovora A lahko preverimo tako, da sledimo postopku, ki sta ga izvedli Monika in Veronika, korak za korakom. Zaporedju korakov od 2 do 4, ki jih ponavljamo, bomo rekli iteracija. Spreminjanje obeh ogrlic v vsaki iteraciji je prikazano na sliki, kjer vsaka vrstica predstavlja eno ogrlico.

ZAČETNO STANJE

Monika

Ru	Z	Z	Ru	Ru	Z	Z	Ru	Ru
M	R	M	M	R	M	R	M	R

Veronika

1. ITERACIJA

Monika

R	Ru	Z	Z	Ru	Ru	Z	Z	Ru	
Ru	M	R	M	M	R	M	R	M	

Veronika

2. ITERACIJA

Monika

		R	Ru	Z	Z	Ru	Ru	Z	Z		
M	Ru	Ru	M	R	M	M	R	M	R		

Veronika

3. ITERACIJA

Monika

R	Z	R	Ru	Z	Z	Ru	Ru	Z			
M	Ru	Ru	M	R	M	M	R	M			

Veronika

4. ITERACIJA

Monika

Z	R	Z	R	Ru	Z	Z	Ru	Ru				
M	M	Ru	Ru	M	R	M	M	R				

Veronika

5. ITERACIJA

Monika

R	Z	R	Z	R	Ru	Z	Z	Ru				
Ru	M	M	Ru	Ru	M	R	M	M				

Veronika

1. iteracija: Obe perli na desni strani ogrlic sta rumene ali rdeče barve, zato sta izmenjani in dodani na levo stran druge ogrlice. Tako sta ogrlici izmenjali po eno perlo.

2. iteracija: Ker ima Monika na desni strani ogrlice rumeno perlo, jo doda na levo stran Veronikine ogrlice. Veronika pa ima na desni strani modro perlo (ki je ne izmenja), zato jo prestavi na levo stran svoje ogrlice. Do sedaj je Monika dala dve perli, Veronika pa eno.

3. iteracija: Naslednja perla na Monikini ogrlici je zelena, zato jo prestavi na levo stran svoje ogrlice. Veronikina naslednja perla pa je rdeča, zato jo doda na levo stran Monikine ogrlice. Tako dva do sedaj obe dali po dve perli.

4. iteracija: Monika ima na desni strani zeleno perlo, zato jo prestavi na levo stran svoje ogrlice. Veronika pa modro perlo z desne strani svoje ogrlice prestavi na levo stran. V tej iteraciji nista izmenjali perl, zato ni spremembe: vsaka od njiju je dala dve perli.

5. iteracija: Naslednji perli na desni strani ogrlic sta rumena in rdeča, zato se izmenjata in dodata na levo stran druge ogrlice. S tem sta tako Monika kot Veronika dali tri perle.

Ker sta druga drugi dali tri perle, postopek zaključita. Njuni ogrlici izgledata tako, kot prikazuje vrstica pri 5. iteraciji na sliki. To ustreza sliki pri odgovoru A.

Računalniško ozadje

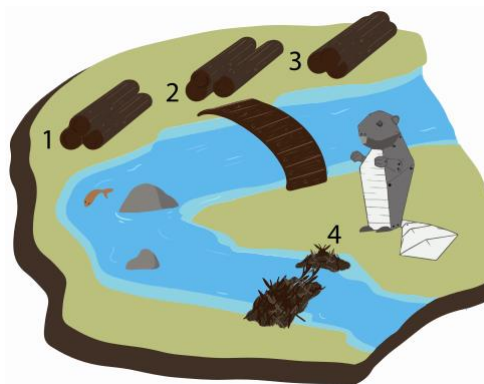
Tudi v tej nalogi smo uporabili FIFO (First In First Out ali prvi noter prvi ven) vrsto. Pri vrsti elemente vedno jemljemo z enega konca in dodajamo na drugi konec. Vrsta je zelo uporabna podatkovna struktura, saj samodejno skrbi za ohranjanje vrstnega reda elementov, ki jih dodamo v vrsto.



Bober Bobo je izdelal robota, da bi namesto njega gradil jezove. Robot lahko naenkrat prenaša le en hlod, pozna pa tri različne ukaze:

- Dx - premakni se do x, kjer je x lahko 1, 2, 3 ali 4;
- P - poberi;
- S - spusti.

Če na primer želi Bobo, da robot vzame hlod z mesta 1 in ga postavi na jez (mesto 4), mora robotu sporočiti naslednje zaporedje ukazov: **D1 P D4 S**.



Bobo želi, da robot vzame po en hlod z vsakega kupa (mesta 1, 2 in 3) ter te hlode postavi v jez, ki ga gradi (mesto 4). Katero je pravilno zaporedje ukazov za to nalogo?

- A) D1 P D4 S D1 P D4 S D1 P D4 S
- B) D1 P D4 S D2 S D4 P D3 S D4 P
- C) D3 P D4 S D2 P D4 S D1 P D4 S
- D) D1 P D4 S D2 P D4 P D3 P D4 S

Rešitev

Pravilen odgovor je C.

Na ukaze pri odgovoru A bo robot pobral vse tri hlode z mesta 1 in nobenega z mest 2 in 3, saj zaporedje ukazov ne vključuje ukazov D2 in D3.

Pri odgovoru B robot na mestu 2 poskuša spustiti hlod, namesto da bi ga tam pobral. Poleg tega pobere hlod na jez in ga spusti na mesto 3 (napačni ukazi so označeni krepko: **D1 P D4 S D2 S D4 P D3 S D4 P**).

Zaporedje ukazov pri odgovoru D pa je napačno, ker robot ne spusti hloda na mestu 4, temveč poskuša tam pobrati še en hlod (napačni ukazi so označeni krepko: **D1 P D4 S D2 P D4 P D3 P D4 S**).

Pravilno je edino zaporedje ukazov pri odgovoru C, kjer robot po vrsti, najprej z mesta 3, nato z mesta 2 in na koncu še z mesta 1, pobere po en hlod in ga prenese na jez.

Računalniško ozadje

Gre za nalogo iz programiranja oziroma sledenja izvajanju programa. To pogosto počnemo, ko iščemo napako v svojem programu ali pa želimo razumeti delovanje programa, ki ga je napisal kdo drug.



Trije sodelavci si delijo isti omrežni tiskalnik. Tiskalnik vsak prejeti dokument vedno natisne v celoti, preden začne tiskati drugi prejeti dokument. Če ima tiskalnik v vrsti že druge dokumente, začne naslednji dokument, ki je na vrsti, tiskati takoj, ko zaključi tiskanje predhodnega dokumenta.

Če na primer ob 10. uri pošljemo tiskalniku dokument, katerega tiskanje traja 2 minuti, začne tiskalnik tiskati ta dokument ob 10.00 in ob 10.02 zaključi tiskanje. Če ob 10.01 (torej med tiskanjem prvega dokumenta) pošljemo tiskalniku drugi dokument, ga bo začel tiskati takoj, ko zaključi s tiskanjem prvega dokumenta, to je ob 10.02.

Danes morajo v pisarni natisniti veliko strani dokumentov, sodelavci pa jih pošiljajo v tiskanje, kot je navedeno:

Sodelavec 1

Št. naročila	Čas	Trajanje tiskanja
11	10.00	2 min
12	10.03	1 min
13	10.10	5 min

Sodelavec 2

Št. naročila	Čas	Trajanje tiskanja
21	10.01	1 min
22	10.04	2 min
23	10.12	1 min

Sodelavec 3

Št. naročila	Čas	Trajanje tiskanja
31	10.07	3 min
32	10.08	5 min
33	10.13	1 min

V kakšnem vrstnem redu bo tiskalnik natisnil dokumente in kdaj bo zaključil s tiskanjem?

- A) 11 12 13 21 22 23 31 32 33, s tiskanjem bo zaključil ob 10.22
- B) 11 21 12 22 31 32 13 23 33, s tiskanjem bo zaključil ob 10.21
- C) 11 12 13 21 22 23 31 32 33, s tiskanjem bo zaključil ob 10.21
- D) 11 21 12 22 31 32 13 23 33, s tiskanjem bo zaključil ob 10.22

Rešitev

Pravilen odgovor je D. Dokumente bo tiskalnik natisnil v naslednjem zaporedju: 11, 21, 12, 22, 31, 32, 13, 23 in 33, s tiskanjem bo zaključil ob 10.22.

Do rešitve pridemo tako, da pogledamo vrstni red, v katerem bo tiskalnik prejel dokumente. Torej vse dokumente uredimo po času pošiljanja tiskalniku (drugi stolpec) in dobimo naslednje zaporedje:

Št. naročila	Čas pošiljanja	Začetek tiskanja	Trajanje	Konec tiskanja
11	10.00	10.00	2 min	10.02
21	10.01	10.02	1 min	10.03
12	10.03	10.03	1 min	10.04
22	10.04	10.04	2 min	10.06
31	10.07	10.07	3 min	10.10
32	10.08	10.10	5 min	10.15
13	10.10	10.15	5 min	10.20
23	10.12	10.20	1 min	10.21
33	10.13	10.21	1 min	10.22

Iz tabele lahko razberemo, da bo tiskalnik zaključil tiskanje ob 10.22.

Računalniško ozadje



Tiskalnik za hranjenje prispelih dokumentov za tiskanje uporablja podatkovno strukturo, ki ji rečemo *vrsta*. Nove dokumente vedno dodajamo na eni strani vrste, medtem ko dokumente brišemo na drugi strani vrste. Enako deluje tudi vrsta na blagajni, kjer se novi kupci vedno postavijo na konec vrste, blagajničarka pa najprej postreže kupca, ki je prvi v vrsti.





Vrsta deluje po principu »kdor prvi pride« oziroma »prvi noter, prvi ven«, za kar se uporablja angleška kratica FIFO (First In First Out). Zato vrsto uporabimo v primerih, ko želimo podatke obdelovati v takem zaporedju, kot jih dobimo. V računalništvu se podatkovna struktura vrsta zelo pogosto uporablja, na primer za pošiljanje sporočil, pretok informacij, v računalniških omrežjih ...



V tovarni izdelujejo robote za opravljanje gospodinjskih opravil. Vsak robot je sestavljen iz glave, telesa in nog. Za robota so na voljo 3 različne glave, telo v 4 oblikah in noge v 3 oblikah.

Stroj samodejno sestavi robota tako, da vzame po eno glavo, eno telo in ene noge, po vrsti, kot prikazujejo spodnje slike:

Glava	 modra	 zelená	 oranžna	(ponovi)
-------	--	---	--	----------

Telo	 krog	 kvadrat	 trikotnik	 petkotnik	(ponovi)
------	---	--	--	---	----------

Noge	 dve nogi	 štiri noge	 ena noga	(ponovi)
------	--	--	--	----------

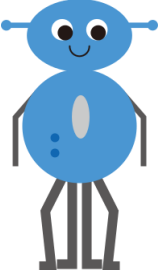
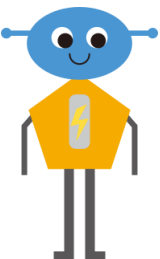
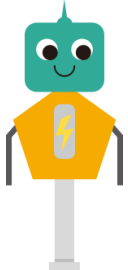

Vsak dan začne stroj delati na enak način: prvega robota izdelá tako, da vzame modro glavo, okroglo telo in dve nogi.

Včeraj ponoči pa je prišlo v tovarni do izpada elektrike, zato se je stroj za izdelavo robotov pokvaril in je začel izdelovati robote na drugačen način. Prvega robota še vedno izdelá pravilno, za naslednje pa uporabi nov vzorec. Namesto da bi jemal dele robota po vrsti, naredi naslednje: pri glavi in pri nogah vedno preskoči en del, pri telesu pa preskoči dva dela.



Drugi robot, ki ga je izdelal tako pokvarjen stroj, je prikazan na desni.

Ko so delavci v tovarni ugotovili, da je stroj pokvarjen, so ga ugasnili, ko je izdelal desetega robota. Kako je izgledal 10. izdelani robot?

A)	B)	C)	D)
			

Rešitev

Pravilen odgovor je B. Deseti izdelani robot ima modro glavo, petkotno telo in dve nogi.

Prvega robota je stroj izdelal na pravi način. Ima modro glavo, okroglo telo in dve nogi. Za vse naslednje robote pa se vzorec delov spremeni.

Glava: Stroj izbira vsako drugo glavo s seznama. Tako je novi vzorec: modra -> oranžna -> zelena (nato se ponovi). Torej ima deseti robot modro glavo.

Telo: Stroj izbira vsako tretje telo s seznama. Tako je novi vzorec: krog -> petkotnik -> trikotnik -> kvadrat (nato se ponovi). Torej ima deseti robot petkotno telo.

Noge: Stroj izbira vsake druge noge s seznama. Tako je novi vzorec: dve nogi -> ena noga -> štiri noge (nato se ponovi). Torej ima deseti robot dve nogi.

Računalniško ozadje

Pri računalniškem razmišljanju sta razpoznavna vzorcev in algoritem dva zelo pomembna koncepta. Oba smo uporabili v tej nalogi.

Razpoznavna vzorcev je metoda, ki se uporablja pri analizi podatkov, kjer iščemo in prikažemo podobnosti med objekti ali nabori podatkov. To nam omogoča, da lažje analiziramo trende ali napovemo prihodnje podatke.

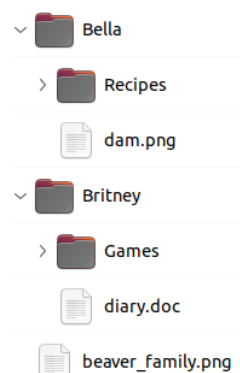
Algoritem je množica pravil in navodil, s katerimi povemo, kako naj bi bil nek proces izveden.



Bella vidi naslednje datoteke na svojem računalniku v ukazni vrstici (levo) in preko grafičnega uporabniškega vmesnika (desno):

```

/Britney
/Britney/Games
/Britney/Games/dam_sweeper.app
/Britney/Games/happy_beavers.app
/Britney/diary.doc
/beaver_family.png
/Bella
/Bella/dam.png
/Bella/Recipes
/Bella/Recipes/cake.txt
    
```



Benjamin pa na svojem računalniku vidi naslednje datoteke:

```

/Bruce
/Bruce/Music
/Bruce/Music/beaver_song.mp3
/Beth
/Beth/Pictures
/Beth/Pictures/birthday.png
/Beth/Pictures/beach.jpg
/Beth/addresses.xls
/readme.txt
/Benjamin
/Benjamin/homework1.doc
    
```

Kateri od naslednjih pogledov na datoteke v grafičnem vmesniku ustreza datotekam na Benjaminovemu računalniku?

	<p>A)</p>	<p>B)</p>	<p>C)</p>	<p>D)</p>
--	-----------	-----------	-----------	-----------

Rešitev

Pravilen odgovor je C. Sliki pri odgovorih A in B prikazujeta mapo Pictures na napačnem mestu, slika pri odgovoru D pa na napačnem mestu prikazuje mapo Music.

Računalniško ozadje

Ljudje shranjujemo datoteke na računalniku v hierarhične strukture. To pomeni, da datoteke postavimo v mape, kjer je vsaka datoteka v natanko eni mapi. Tudi vsaka mapa je v natanko eni mapi (izjema je le najbolj vrhnja mapa, ki ji pravimo korenski direktorij). Vsaka mapa pa lahko vsebuje več datotek in/ali drugih map.

Ideja takega načina shranjevanja datotek v mape je zelo stara in izhaja še iz časov pred računalniki, ko so na podoben način shranjevali papirnate dokumente.

Direktojska struktura z datotekami in mapami je primer drevesne strukture, ki se navadno uporablja za organizacijo hierarhičnih podatkov.

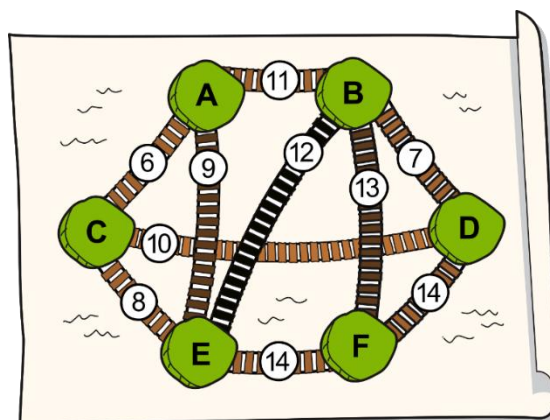
Poleg direktojske strukture pa za organizacijo datotek lahko uporabljamo tudi druge načine, kot je na primer označevanje datotek, kjer vsaki datoteki določimo eno ali več oznak.



Skupnost, ki živi na šestih otokih, želi povezati te otoke z več visečimi mostovi. Naredili so načrt (na sliki), kako bi lahko povezali otoke z mostovi. Mostovi se ne sekajo; most lahko teče pod oz. nad drugimi mostovi. Na načrtu je označena tudi cena za gradnjo vsakega mosta (številka na mostu).

V skupnosti želijo povezati vse otoke tako, da bodo preko enega ali več mostov lahko potovali z vsakega otoka na kateri koli drug otok. Hkrati pa želijo zgraditi mostove s čim manj stroški.

Koliko stane najcenejši način povezave vseh otokov?

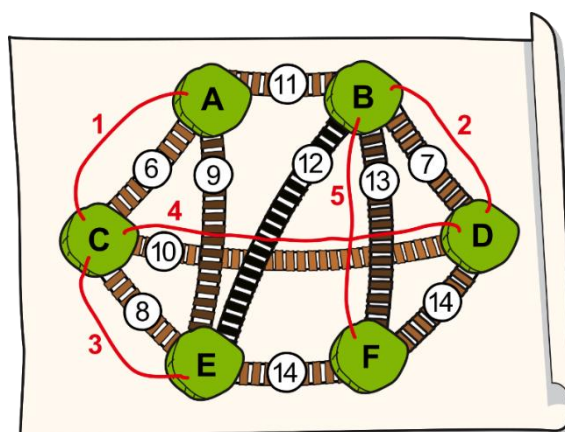


Rešitev

Vse otoke lahko povežejo s 5 mostovi, katerih skupna cena je 44.

Do rešitve lahko pridemo z uporabo enostavnega in intuitivnega algoritma: mostove gradimo po vrsti od najcenejšega do najdražjega, pred gradnjo vsakega pa preverimo, če ta most res potrebujemo.

1. Začnemo z najcenejšim mostom, to je most A - C, ki stane 6.
2. Nato izberemo drugi najcenejši most, to je most B - D, ki stane 7.
3. Tretji najcenejši je most C - E s ceno 8.
4. Naslednji najcenejši most je A - E, ki stane 9. Vendar pa bi s postavitvijo tega mostu dobili krožno pot AECA, kar pomeni, da tega mosta niti ne potrebujemo, saj lahko z otoka A dosežemo otok E tudi preko otoka C.
5. Četrty most, ki ga postavimo, je C - D s ceno 10.
6. Naslednji najcenejši most je A - B s ceno 11. Vendar bi tudi ta most omogočil krožno pot ABDCA, torej ga ne potrebujemo, saj z otoka A lahko dosežemo otok B preko otokov C in D.
7. Podobno velja tudi za most B - E s ceno 12, pri katerem dobimo krožno pot BECDB.
8. Peti most, ki ga moramo postaviti, je B- F s ceno 13.



Sedaj imamo povezanih vseh šest otokov in ker smo na vsakem koraku izbrali najcenejši most, je tudi skupna cena petih povezovalnih mostov najnižja: $6 + 7 + 8 + 10 + 13 = 44$.

Računalniško ozadje

Podani problem je problem iskanja minimalnega vpetega drevesa za podan graf. Vpeto drevo je podgraf danega grafa, ki ima ista vozlišča, a le nekatere povezave in je drevo.

Za rešitev problema smo uporabili Kruskalov algoritem, ki vedno najde minimalno vpeto drevo za povezan utežen graf. Algoritem temelji na dodajanju posameznih povezav z najmanjšo utežjo, pri tem pa se izogibamo ciklom z že dodanimi povezavami.

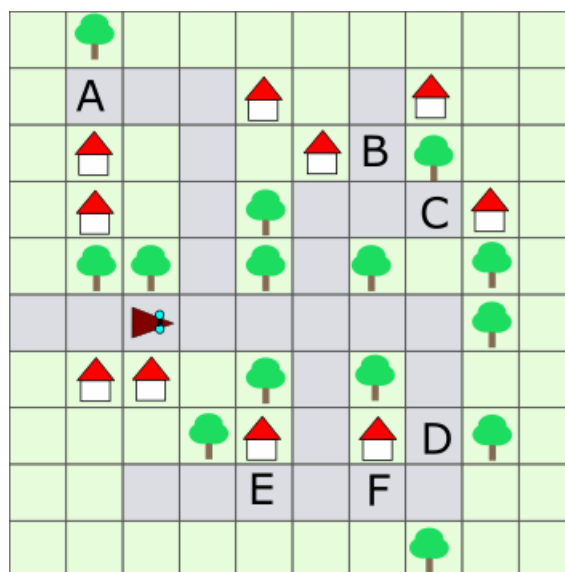


Tina je slepa in za hojo po ulicah mesta uporablja govoreča očala. Ta imajo kamere in inteligen ten sistem za prepoznavo objektov. Lahko razpoznajo štiri vrste objektov: hišo, drevo, cesto in zelenico (štirje različni kvadrati na zemljevidu).

Ko Tina stopi na nov kvadr atek na cesti, ji govoreča očala povejo, kaj je na njeni levi, kaj je pred njo in kaj je na njeni desni, v tem vrstnem redu. Ko je Tina na kvadr atku, kot prikazuje zemljevid, ji očala povejo: »drevo, cesta, hiša«.

Tina začne svojo pot na kvadr atku, kot je prikazano na sliki (obrnjena je proti desni strani zemljevida), in posluša svoja očala. Očala ji govorijo naslednje (začenši z začetnim kvadr atkom):

- drevo, cesta, hiša;
- cesta, cesta, zelenica;
- drevo, cesta, drevo;
- cesta, cesta, cesta;
- drevo, cesta, drevo;
- drevo, hiša, cesta;
- cesta, cesta, drevo;
- hiša, cesta, drevo.



Na koncu Tina prispe do enega od kvadr atkov, ki so označeni s črko. Do katerega?

Rešitev

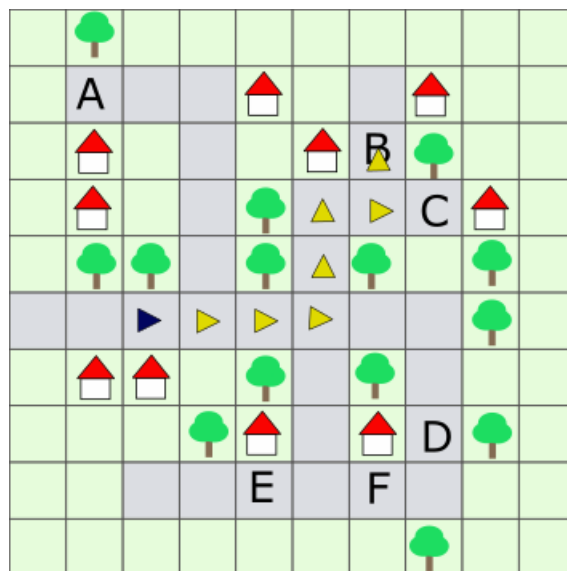
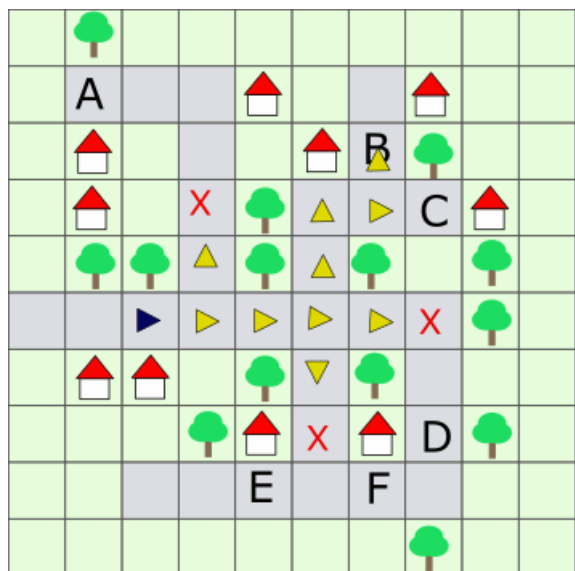
Tina je prišla do kvadr atka B.

Rešitev lahko poiščemo na različne načine. Lahko sledimo Tinini poti od začetka in hodimo po poti, ki se ujema z informacijami, ki jih dobimo od govorečih očal. Očala ne povejo, kdaj Tina zavije levo ali desno na cesti. Zato v primeru, ko se na neki lokaciji naslednji opis objektov, ki jih podajo govoreča očala, ujema z objekti na več kot le enem sosednjem kvadr atku (naslednji korak poti), moramo preveriti obe alternativni. Na primer, ko se Tina prestavi za tri kvadr atke (njena okolica je cesta, cesta, cesta), je naslednje kvadr atek, na katerega se prestavi (opis drevo, cesta, drevo) lahko en kvadr atek naravnost naprej ali pa kvadr atek na levi (obrne se levo in gre en kvadr atek naprej). Preveriti moramo obe alternativni, če zanju veljajo tudi nadaljnji opisi.

Do rešitve lahko pridemo tudi tako, da preverimo poti do vseh možnih destinacij (kvadr atkov s črko) in za vsako pot sproti preverjamo, če se opisi na poti ujemajo z opisi, prejetimi od govorečih očal. Tako na primer preverimo pot do kvadr atka A in ugotovimo, da bi po tretjem

kvadratku (z opisom drevo, cesta, drevo) moral slediti opis zelenica, cesta, drevo, medtem ko so očala Tini sporočila opis cesta, cesta, cesta. To pomeni, da pot do destinacije A ni tista, po kateri je šla Tina.

Spodnja leva slika prikazuje vse možne poti, po katerih bi lahko šla Tina. X označuje, da po tej poti ni mogla nadaljevati (opis se ne ujema).



Še hitreje pa lahko najdemo rešitev, če najprej preverimo vse ciljne kvadratke (torej kvadratke s črkami). Le kvadratka B in D ustrežata opisu hiša, cesta, drevo. Vendar noben od sosednjih kvadratkov kvadratka D ne ustreza opisu cesta, cesta, drevo, ki je predzadnji opis. Zato je lahko Tinin ciljni kvadrata le B. Zgornja desna slika prikazuje Tinino pot.

Računalniško ozadje

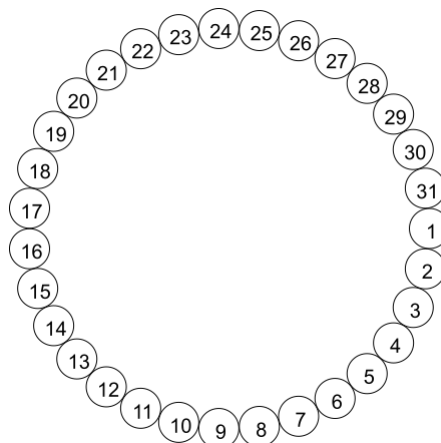
V nalogi smo uporabili dnevniški zapis sporočil govorečih očal, da smo uspešno rekonstruirali Tinino pot. V splošnem bi sicer lahko mislili, da le ta dnevniški zapis nima dovolj informacij, da bi lahko ugotovili, kam je Tina prišla, saj ne hrani smeri Tininih premikov oz. njenih zavojev levo ali desno. Vendar pa nam dodatno znanje, ki ga imamo o okolici, v kateri se Tina premika, omogoča, da vseeno lahko izsledimo Tinino destinacijo. Vidimo lahko, da so osebni podatki lahko zelo občutljivi, še posebej pa to velja za prostorske ali geografske podatke.



V razredu je 31 stolov. Postavljeni so v krogu in oštevilčeni po vrsti od 1 do 31, kot prikazuje desna slika.

Otroci dobijo naslednja navodila:

- v razred vstopajte posamično,
- sedi na stol, katerega številka je enaka tvojemu dnevu rojstva,
- če je ta stol že zaseden, pa se pomakni v smeri urinega kazalca do prvega prostega stola in se usedi nanj.



Primer: Greta in Silva sta dvojčici, rojeni 20. aprila, Aleš je rojen 21. januarja in Zvonko 22. septembra. Če v razred vstopijo najprej Greta, nato pa še Silva, Zvonko in Aleš, potem Greta sedi na stolu s številko 20, Silva na stolu 21, Zvonko na stolu 22 in Aleš na stolu 23. Če pa v razred vstopi najprej Silva, nato pa še Aleš, Zvonko in Greta, sedi Silva na stolu s številko 20, Aleš na 21, Zvonko na 22 in Greta na 23.

V razred vstopi šest otrok in sedijo na stolih, kot kaže tabela.

Ime	Rojstni dan	Sedi na stolu
Ana	11. maj	13
Boštjan	12. februar	12
Ciril	14. september	14
Darja	11. avgust	11
Ema	13. april	15
Franci	12. julij	16

Katera od naslednjih trditev **ne** more biti pravilna?

- A) Ciril se je usedel prvi.
- B) Franci se je usedel zadnji.
- C) Ema se je usedla pred Ano.
- D) Boštjan se je usedel pred Darjo.

Rešitev

Pravilni odgovor je C.

Ciril sedi na stolu z enako številko, kot je njegov dan rojstva. Zato je možno, da se je usedel prvi. Torej je odgovor A lahko pravilna trditev.

Franci sedi na stolu 16, zato so bili ob njegovem prihodu stoli 12 do 15 že zasedeni. Darja sedi na stolu 11, zato je morala priti pred Ano, ki ima isti dan rojstva. Torej je tudi odgovor B pravilna trditev.

Če bi se Ema usedla pred Ano, bi bil stol 13 še prost in se ji ne bi bilo potrebno prestaviti do stola 15. Torej odgovor C ne navaja pravilne trditve.

Boštjan in Darja sedita na stolu s številko dneva rojstva, torej ne moremo vedeti, kateri od njiju je vstopil prvi. Torej je tudi odgovor D lahko pravilna trditev.

Računalniško ozadje

Naloga ilustrira *zgoščevanje* (angl. *hashing*), s katerim veliko množico možnih vhodnih vrednosti (imena otrok) preslikamo v manjšo nespremenljivo množico izhodnih vrednosti (številka stola). Za preslikavo smo uporabili zgoščevalno funkcijo, v našem primeru je dan rojstva. Ker pa je vhodna množica vrednosti veliko večja od izhodne množice, se lahko zgodi, da se dve (ali pa tudi več) vhodni vrednosti preslikata v isto izhodno vrednost. Za obravnavo takih primerov moramo predvideti ustrezen mehanizem, v našem primeru smo dobljeno izhodno vrednost povečali za ena.



Bombom in Lala v svoji restavraciji komunicirata s pomočjo dveh signalnih svetilk, prva je v obliki lista (🍃), druga v obliki rožice (🌸). Vsaki jedi na meniju sta določila enolično kodo, sestavljeno iz teh dveh oblik. Kode kaže tabela na desni. Bombom posreduje naročila hrane Lali tako, da prižge in ugasne svetilki v določenem zaporedju, ki ustreza kodi jedi. Lala pa nato pripravi ustrezno jed.

Jed	Koda
burger	🍃🍃
ocvrta riž	🍃🍃🌸

Ker Lala začne pripravljati jed takoj, ko razpozna kodo, sta naletela na problem: ko Bombom pri oddajanju kode za ocvrta riž dvakrat posveti s svetilko v obliki lista, Lala prepozna naročilo za burger in začne z njegovo pripravo. Tako nikoli ne pripravi ocvrtega riža. Da bi rešila ta problem, sta spremenila kode jedi. Nove kode za vse jedi na meniju so podane v tabeli.

Jed	Koda
burger	🍃🍃
ocvrta riž	🍃🌸🍃🌸
sendvič	🌸🍃
pica	🍃🌸🌸
toast	🌸🌸🌸

Nekega dne se odločita, da bosta na meni dodala še pomfrit. Katero kodo lahko uporabita za pomfrit?

- A) 🌸🌸🍃
- B) 🍃🍃🌸
- C) 🍃🌸🍃
- D) 🌸🌸🌸🍃
- E) 🍃🌸🌸🌸

Rešitev

Lala začne pripravljati napačno jed, kadar je cela koda te jedi enaka začetku kode neke druge jedi. Zato se nobena koda jedi ne sme začeti s kodo druge jedi.

Odgovor B ni pravilen, saj je koda za burger (🍃🍃) enaka začetku kode pri odgovoru B (🍃🍃🌸). Podobno velja tudi pri kodi za toast (🌸🌸🌸) in odgovoru D (🌸🌸🌸🍃) ter pri kodi za pico (🍃🌸🌸) in odgovoru E (🍃🌸🌸🌸). Vendar tudi odgovor C ne more biti pravi, ker je njegova koda (🍃🌸🍃) identična začetku (nove) kode za ocvrta riž (🍃🌸🍃🌸).

Pravilni odgovor je lahko le A, saj celotna koda pri odgovoru A ni enaka začetku nobene druge kode jedi in tudi celotna koda nobene druge jedi ni enaka začetku kode pri odgovoru A.

Računalniško ozadje

Lala in Bombom sta uporabila pravilo kodiranja, kjer nobena koda ni enaka začetku neke druge kode. Kode, ki imajo tako lastnost, imenujemo *prefiksne kode*. V računalništvu jih uporabljamo za stiskanje podatkov.

Brez stiskanja bi besedo lahko zapisali tako, da vsako črko zakodiramo z bitnim zaporedjem (tj. zaporedjem ničel in enic) enake dolžine. Na primer, pri uporabi kodnega nabora ASCII, ki za zapis znaka uporabi 7 bitov, bi za zapis imena EVA (3 črke) porabili $3 \times 7 = 21$ bitov.

Če pa uporabimo prefiksno kodiranje, določimo črkam, ki se pogosteje uporabljajo, krajše kode, bolj redkim črkam pa daljše kode. Najpogosteje se v besedah pojavlja črka E, zato jo zakodiramo z 1. Črka A se tudi pojavlja pogosto, a nekoliko redkeje kot E. Zakodiramo jo lahko z dvema bitoma kot 01. Črka V pa se redko pojavlja, zato zanjo lahko uporabimo najdaljšo kodo: 001. Pri določanju kod črkam smo pazili, da so kode prefiksne. Tako lahko ime EVA zakodiramo kot 100101, za kar potrebujemo le 6 bitov. Ker je koda prefiksna, lahko zakodirano zaporedje brez težav enolično dekodiramo, tj. pretvorimo nazaj v besedilo.

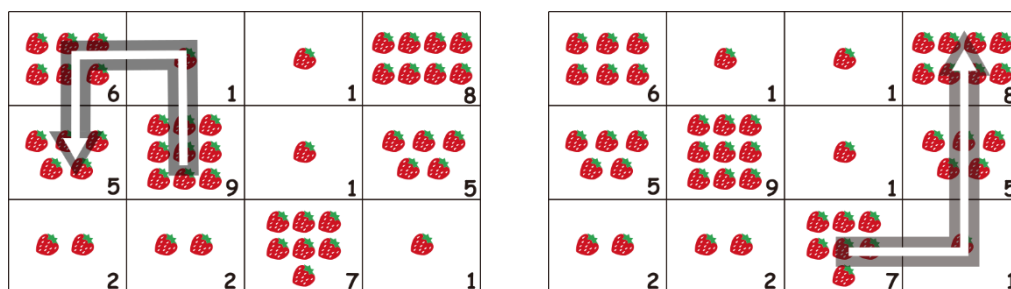
Lov na jagode

srednja šola



V igri Lov na jagode postavimo različno število jagod v vsako polje mreže. Igralec začne na katerem koli polju in naredi tri korake. Pri vsakem koraku se lahko premakne gor, dol, levo ali desno na sosednje polje. Na taki izbrani poti igralec obiše štiri polja in poje vse jagode, ki so na teh poljih.

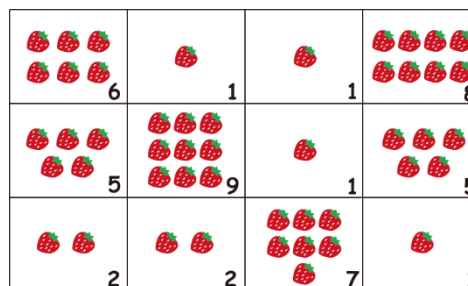
Dva primera poti na isti mreži z jagodami kažeta sliki:



Na poti, prikazani na levi sliki, bo igralec lahko pojedel $9 + 1 + 6 + 5 = 21$ jagod.

Na poti, prikazani na desni sliki, bo igralec prav tako pojedel $7 + 1 + 5 + 8 = 21$ jagod.

Največ koliko jagod lahko poje igralec na mreži z jagodami na desni?



Rešitev



















Igralec lahko poje največ 23 jagod.

Čeprav je 51 možnih poti po dani mreži (glej spodnjo tabelo), ni potrebno, da izračunamo vsoto pobranih jagod za vseh 51 poti. Vemo, da lahko vsak igralec doseže le 4 zaporedna sosednja polja. Ta štiri polja lahko sestavljajo naslednje oblike:

oblika	kvadrat	črta	oblika L	oblika L vodoravno	oblika Z	oblika Z navpično
pot						
možnosti	6	3	12	16	8	6

Preverimo navedene poti za vse možnosti, ki vsebujejo polje z 9 ali polje z 8 jagodami (vključno s potmi, ki vsebujejo obe polji). Ker kombinacije poti z vključenimi polji z 9 ali 8 jagodami vključujejo tudi polja s 5, 6 ali 7 jagodami, bomo tako preverili vsa polja z večjim številom jagod. Potem lahko poiščemo največje število jagod na eni poti:

Pot določene oblike z največjim številom jagod

	vsebuje polje 9 jagod	vsebuje polje 8 jagod	večje število
kvadrat 	 $9 + 5 + 6 + 1 = 21$	 $8 + 5 + 1 + 1 = 15$	21
črta 	 $5 + 9 + 1 + 5 = 20$	 $6 + 1 + 1 + 8 = 16$	20
oblika L 	 $7 + 2 + 9 + 1 = 19$	 $8 + 5 + 1 + 7 = 21$	21
oblika L vodoravno 	 $9 + 1 + 5 + 8 = 23$	 $9 + 1 + 5 + 8 = 23$	23
oblika Z 	 $5 + 9 + 2 + 7 = 23$	 $8 + 1 + 1 + 9 = 19$	23
oblika Z navpično 	 $6 + 5 + 9 + 2 = 22$	 $8 + 5 + 1 + 7 = 21$	22

Ugotovimo lahko tudi, da je največja vsota jagod na štirih poljih, če ne upoštevamo polj z 9 in 8 jagodami, lahko največ $7 + 6 + 5 + 5 = 23$. Torej tudi če v pot ne vključimo polj z 9 in 8 jagodami, števila 23 ne moremo preseči. Torej lahko igralec pobere največ 23 jagod.

Računalniško ozadje

Veliko problemov v računalništvu je takih, da iščemo največjo ali najmanjšo vrednost, kot je na primer opraviti največ dela, porabiti najmanj denarja ali pa – kot v naši nalogi – pobrati največ jagod. Ker imamo pri takih problemih več možnosti, kako opraviti delo, porabiti denar ali pobrati jagode, je izziv med vsemi možnostmi poiskati tiste z največjo ali najmanjšo vrednostjo. Temu pravimo optimizacijski problem.

Optimalno rešitev lahko poiščemo na več načinov. Eden od njih je, da poiščemo vse možne rešitve, za vsako od njih izračunamo vrednost in nato poiščemo tiste, ki imajo najboljšo vrednost. Takemu pristopu rečemo *iskanje z grobo silo* (angl. *brute force search* ali *exhaustive search*). Vendar je tak pristop zelo počasen in zahteva veliko računanja, če imamo veliko možnih rešitev. Zato je boljši pristop, da najprej analiziramo problem in poiščemo omejitve, ki nam zmanjšajo število možnih rešitev, ter tako hitreje najdemo optimalno rešitev. Za nekatere probleme v računalništvu obstajajo poznane metode za učinkovito iskanje optimalne rešitve. Za druge probleme pa je znano, da jih ne moremo rešiti učinkovito.



Imamo škatlice s petimi oštevilčenimi predalčki (rdeče številke). V vsakem predalčku je listek s številko med 1 in 5.

	1	2	3	4	5
X	3	5	2	4	1

Navodilo (X 1) pomeni: vzemi številko iz prvega predalčka. Dobili bomo številko 3.

Navodila so lahko tudi bolj zapletena. Vzemimo navodilo (X (X (X 1))). Vzamemo številko iz prvega predalčka. Dobimo 3. Zato vzamemo številko iz tretjega predalčka, dobimo 2. Zato vzamemo številko iz drugega predalčka. Dobimo 5.

Še bolj zabavno je, če imamo več škatlic s predalčki.

Kaj dobimo, če sledimo navodilu (A (C (B 4)))?

Vzamemo številko iz četrtega predalčka B; dobimo 3.

Zato vzamemo tretjo številko iz C; dobimo 4. Zato vzamemo četrto številko iz A; dobimo 1.

A	3	2	4	1	5
---	---	---	---	---	---

B	5	4	1	3	2
---	---	---	---	---	---

Kaj dobimo, če sledimo navodilu (A (B (C 3)))?

C	2	5	4	3	1
---	---	---	---	---	---

Rešitev

Dobimo 4.

(A (B (C 3))) pomeni, da najprej vzamemo tretjo številko iz predalčka C, dobimo 4. Potem vzamemo četrto številko iz predalčka B in dobimo 3. Zato vzamemo tretjo številko iz predalčka A in dobimo 4. To je tudi naša rešitev.

Računalniško ozadje

Škatlice v nalogi predstavljajo sezname z imeni A, B in C. Oštevilčeni predalčki predstavljajo indeksirane celice v seznamu. Listi predstavljajo podatek, ki je shranjen na posameznem mestu v seznamu. V tej nalogi so bili predalčki označeni s številkami od 1 do 5. V programskem jeziku Python bi bile celice oštevilčene z indeksi od 0 do 4.



Skupina prijateljev se odloča, katerega od sedmih filmov si bodo ogledali v kinu. Vsak od njih je vsak film ocenil kot dober (✓), srednje dober (•) ali slab (✗). Ocene kaže tabela.

	1	2	3	4	5	6	7
Alja	✓	✓	✓	✓	✓	✓	✓
Nataša	•	✓	✓	•	•	✓	✓
Nik	✗	✗	✗	•	✗	✗	✗
Gregor	✗	•	•	•	✗	•	✗
Edo	✓	•	✗	✗	•	✓	✓
Cvetka	•	✗	•	✗	✓	•	•

Priljubljen film je tisti, kateremu so vsi prijatelji podelili svojo najboljšo oceno. Na primer, film številka 6 ni priljubljen film, saj je Nikova najboljša ocena dana filmu 4 (in tudi Cvetka je najboljšo oceno dala filmu 5).

Trenutno noben film ni priljubljen. Najmanj koliko ocen filmov moramo spremeniti, da bomo dobili priljubljen film?

Rešitev

Spremeniti moramo dve oceni.

Ni možno, da bi priljubljen film dobili s spremembo le ene ocene, saj za vsak film velja, da sta vsaj dva od prijateljev boljše ocenila nek drug film.

Film Število prijateljev, ki so boljše ocenili nek drug film

- 1 4: Nataša, Nik, Gregor in Cvetka
- 2 3: Nik, Edo in Cvetka
- 3 3: Nik, Edo in Cvetka
- 4 3: Nataša, Edo in Cvetka
- 5 4: Nataša, Nik, Gregor in Edo
- 6 2: Nik in Cvetka
- 7 3: Nik, Gregor in Cvetka

Če Nik in Cvetka spremenita vsak po eno svojo oceno, lahko film številka 6 postane priljubljen film. Nik in Cvetka morata izboljšati svoji oceni filma 6. Druga možnost pa je, da znižata ocene tistih filmov, ki sta jih ocenila boljše kot film 6: Nik zniža oceno filmu 4, Cvetka pa zniža oceno filmu 5.

Računalniško ozadje

Kako smo se lotili rešitve problema? En način je, da za vsak film po vrsti in za vsakega od prijateljev po vrsti pregledamo, ali obstaja boljša ocena za kakšen drug film. Tako na koncu dobimo tabelo, kot smo jo sestavili pri rešitvi naloge. Iz nje lahko razberemo, kateri prijatelji morajo spremeniti svoje ocene, da dobimo priljubljen film s kar najmanj spremembami ocen. Tako uspešno rešimo problem.

Vendar, ali je algoritem tudi učinkovit? Ali bi lahko problem rešili hitreje? V splošnem lahko imamo F filmov in P prijateljev. Pregledati moramo vseh $F \times P$ elementov tabele in za vsakega od njih moramo pregledati $F - 1$ ostalih ocen iste osebe. Tako moramo skupaj pogledati $F \times (F - 1) \times P$ ocen. Vendar pa za ugotavljanje, ali ocena neke osebe za določen film predstavlja problem, zadostuje, da poiščemo najboljšo dano oceno te osebe. Če je ocena filma nižja od najboljše dane ocene te osebe, ta film ne more biti priljubljen film. Torej lahko najprej za vsakega prijatelja preverimo, kakšna je njegova najboljša dana ocena (tako da pregledamo vseh $F \times P$ ocen), nato pa preverimo vseh $F \times P$ ocen, ali so nižje od najboljše dane ocene osebe. Tako moramo skupaj preveriti $2 \times F \times P$ ocen. Če je $F = 7$ in $P = 6$, kot v našem primeru, to znese 84 preverb, medtem ko je prvotni algoritem zahteval $7 \times 6 \times 6 = 252$ preverb. Ta drugi algoritem tudi uspešno reši problem, a na bolj učinkovit način kot prvi algoritem.

V računalništvu ni pomembno le uspešno reševanje problemov, ampak tudi kar najbolj učinkovito. Hitrejša strojna oprema omogoča hitrejšo reševanje problemov, vendar ne pomaga veliko, če za rešitev problema ne obstaja tudi učinkovit algoritem.

Pregled nalog

Šolsko tekmovanje

			2.	3.	4.	5.	6.	7.	8.	9.	SŠ
Pri kosilu	Slovenija		•								
Abeceda	Slovenija		•	•							
Barvanje robotov	Savdska Arabija		•	•							
Robotska čebela 1	Slovaška		•	•							
Gozdna zabava	Irska		•	•	•	•					
Hamburger	Južna Koreja			•							
Obisk babice	Litva			•							
Smučarska vas	Slovaška			•							
Rojstnodnevna zabava	Tajvan			•	•						
Robot Smetko 1	Kitajska			•	•	•					
Kovanci	Madžarska				•	•					
Navodila za gradnjo	Vietnam				•	•					
Robotska čebela 2	Slovaška				•	•					
Majini sosedi	Ciper				•	•	•	•			
Starodavne rune	Nova Zelandija				•	•	•	•			
Vijaki z matico	Kanada				•	•	•	•	•	•	
Barvna sestavljanke	Vietnam				•	•	•	•	•	•	•
Zaporedje kamenčkov	Nizozemska					•	•	•			
Šifra 8	Slovaška					•	•	•			
Zaklad	S. Makedonija					•	•	•	•	•	
Slike gozda	Argentina					•	•	•	•	•	•
Preproga	Turčija						•	•	•	•	
Ključki	Pakistan						•	•	•	•	•

			2.	3.	4.	5.	6.	7.	8.	9.	SŠ
Robot Smetko 2	Kitajska						•	•	•	•	
Ogrlice	Makedonija						•	•	•	•	•
Robotski bober	Črna gora						•	•	•	•	•
Tiskalnik	Romunija						•	•	•	•	•
Tovarna robotov	Avstrija						•	•	•	•	•
Datoteke	Finska								•	•	•
Mostovi	Litva								•	•	•
Poslušaj in hodi	Nemčija								•	•	•
Sedežni red	Indija								•	•	•
Sporočila	Poljska								•	•	•
Lov na Jagode	Tajvan										•
Predalčki	Avstrija										•
Priljubljen film	Nemčija										•

Avtorji nalog

Gulgun Afacan, Turčija
Daumilas Ardickas, Litva
Liam Baumann, Avstrija
Marta Burzańska, Poljska
Špela Cerar, Slovenija
David Clark, Nova Zelandija
Christian Datzko, Madžarska
Susanne Datzko, Švica
Janez Demšar, Slovenija
Adam Grodeck, Avstralija
Yasemin Gulbahar, Turčija
Husnul Hakim, Indonezija
Andrea Hrušecká, Slovaška
Thomas Ioannou, Ciper
Mile Jovanov, Severna Makedonija
Ungyeol Jung, Južna Koreja
Taina Lehtimäki, Irska
Marielle Léonard, Francija
Yilin Li, Kitajska
Judith Lin, Tajvan
Alenka Kavčič, Slovenija
Anja Koron, Slovenija
Anja Knežević, Slovenija
Monika Maneva, Severna Makedonija
Anna Patricia Miravite, Filipini
Tom Naughton, Irska
Veronika Ognjanovska, Severna Makedonija
Margot Phillipps, Nova Zelandija
Chris Roffey, Velika Britanija
Nežka Rugelj, Slovenija
Emil Stankov, Severna Makedonija
Veronika Stefanovska, Severna Makedonija
Monika Tomcsányiová, Slovaška
Willem van der Vegt, Nizozemska
Rechilda Villame, Filipini
Xiao Yang, Kitajska
Yi Shan Yeh, Tajvan
Vu Van Luan, Vietnam
Wang Yu, Kitajska
Chen Zhengrong, Kitajska
Andrea Hrušecká, Slovaška
Chris Roffey, Velika Britanija
Leonardo Cavalcante, Brazilija
Mathias Hiron, Francija
Marvin G. Hall, Jamajka
Kristina Slišurić, Hrvaška
Evgenia Sendova, Bolgarija
En-hsuan Lee, Tajvan
Judith Lin, Tajvan
Yeh Yi Shan, Tajvan
Justina Dauksaite, ZDA
Valentina Dagienė, Litva
Vaidotas Kinčius, Litva
Linda Björk Bergsveinsdóttir, Islandija
Laila Alharthi, Savdska Arabija
Yong Mao, Kitajska
Graeme Buckie, Avstralija