

Analiza rješenja natjecateljskih zadataka za osnovnu školu iz predmeta Informatika

Cvitanović, Marija

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:453119>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-23**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



**SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK**

Marija Cvitanović

**ANALIZA RJEŠENJA NATJECATELJSKIH
ZADATAKA ZA OSNOVNU ŠKOLU IZ PREDMETA
INFORMATIKA**

Diplomski rad

Voditelj rada:

doc. dr. sc. Goranka Nogo

Zagreb, 2019.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

*Zahvaljujem doc. dr. sc. Goranki Nogo na pomoći i strpljivosti tijekom izrade
diplomskog rada.*

*Također, zahvaljujem prijateljima! Posebno hvala Karli što je ovo putovanje učinila
ljepšim i naučila me da su brige manje kada se podijele.*

*Najviše zahvaljujem mojoj obitelji na brizi, razumijevanju i bezuvjetnoj podršci
tijekom mog obrazovanja.*

Sadržaj

Uvod	1
1 Natjecanje iz informatike u Republici Hrvatskoj	3
1.1 Algoritmi OŠ.....	4
2 Natjecateljski zadaci u petom razredu	7
2.1 Primjena algoritama	7
2.2 LOGO.....	14
3 Natjecateljski zadaci u šestom razredu	19
3.1 Primjena algoritama	19
3.2 LOGO.....	26
4 Natjecateljski zadaci u sedmom razredu	32
4.1 Primjena algoritama	32
4.2 LOGO.....	38
5 Natjecateljski zadaci u osmom razredu	42
5.1 Primjena algoritama	42
5.2 LOGO.....	47
Zaključak	54
Literatura	55
Sažetak	57
Summary	58
Životopis	59

Uvod

U ovome radu analizirat ćemo natjecateljske zadatke za osnovnu školu iz predmeta Informatika posljednjih pet godina te njihova rješenja. Također, uspoređivat ćemo potrebna znanja iz službenih propozicija s gradivom najzastupljenijih udžbenika (*SysPrint* i *Školska knjiga*) prema Katalogu obveznih udžbenika za školsku godinu 2018./2019. te s ishodima domene **Računalno razmišljanje i programiranje** (*Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*).

Kroz prvo poglavlje detaljno ćemo opisati natjecanje iz predmeta Informatika u Republici Hrvatskoj. Objasniti ćemo kako je organizirano te koje su kategorije i razine natjecanja. Posebnu pažnju posvetit ćemo kategoriji **Algoritmi za osnovnu školu: Primjena algoritama i LOGO**, čije zadatke ćemo detaljno analizirati u sljedećim poglavljima.

U drugom poglavlju proučavat ćemo zadatke s natjecanja za peti razred osnovne škole u kategorijama **Primjena algoritama i LOGO**. Osvrnut ćemo se na znanja potrebna učenicima za natjecateljske zadatke, usporediti ih s gradivom promatranih udžbenika te *Kurikulumom nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*. Za svaki zadatak, istaknut ćemo zašto smo odabrali pojedini zadatak te koja znanja su potrebna učenicima za riješiti zadatak.

Treće poglavlje bit će analiza natjecateljskih zadataka za šesti razred osnovne škole u kategorijama **Primjena algoritama i LOGO**. Na početku poglavlja, kratko ćemo usporediti potrebna učenička znanja iz službenih propozicija s temama obrađenim u udžbenicima te ishodima domene **Računalno razmišljanje i programiranje**. Također, kao i u prethodnom poglavlju, komentirat ćemo izdvojene natjecateljske zadatke te rješenja tih zadataka.

U četvrtom poglavlju proučavat ćemo zadatke s natjecanja za učenike sedmog razreda u kategoriji **Algoritmi za osnovnu školu**. Na početku poglavlja usporedit ćemo potrebna znanja za natjecateljske zadatke s gradivom redovne nastave predmeta Informatika. Zatim, odabrat ćemo ogledne zadatke s natjecanja te ih detaljno opisati.

Peto poglavlje bit će analiza zadataka s natjecanja za učenike osmog razreda u kategoriji **Algoritmi za osnovnu školu**. Kao i u prethodnim poglavljima, prvo ćemo usporediti gradivo natjecateljskih zadataka s gradivom redovne nastave, a zatim ćemo provesti detaljnu analizu zadataka koji se pojavljuju na natjecanju učenika osmog razreda.

Zbog složenosti, u ovome radu ograničili smo se na zadatke s prve dvije razine (školska i županijska razina natjecanja).

Dijelovi teksta preuzeti iz službenih dokumenata pisani su kurzivom.

1 Natjecanje iz informatike u Republici Hrvatskoj

Natjecanje iz informatike - Ulaskom u novo, treće desetljeće, nije se promijenila želja da to ostane najatraktivnije školsko natjecanje. I dalje se nastoji unaprijediti način njegovog odvijanja kako bi se uključio što veći broj naprednih, nadarenih i sposobnih mladih natjecatelja.¹

Infokup je službeno natjecanje iz informatike za učenike srednjih i osnovnih škola u Republici Hrvatskoj. Organizatori natjecanja su Ministarstvo znanosti i obrazovanja, Agencija za odgoj i obrazovanje te Hrvatski savez informatičara, a internetsko sjedište natjecanja, gdje se objavljuju sve upute i informacije vezane za natjecanje, je informatika.azoo.hr.

Na natjecanju iz informatike natjecati se mogu redoviti učenici osnovnih i srednjih škola. Svaki natjecatelj mora imati odgovornog mentora kojeg imenuje ravnatelj škole. Odgovorni mentor učeniku natjecatelju može biti isključivo odgojno-obrazovni radnik (učitelj, nastavnik ili stručni suradnik) zaposlen u školskoj ustanovi (osnovnoj školi, srednjoj školi ili učeničkom domu). Zadaće odgovornog mentora su pomagati učenicima u pripremi za natjecanje, omogućiti učenicima sudjelovanje na školskom natjecanju te surađivati s vanjskim mentorom (osoba koja nije zaposlenik škole, a priprema učenika za natjecanje u sklopu izvannastavnog programa u školi, informatičkom klubu ili nekoj drugoj instituciji).

Prijava za natjecanje odvija se putem internetskog sučelja na internetskom sjedištu informatika.azoo.hr te svi sudionici natjecanja (učenici, mentori, povjerenstva) trebaju imati svoj račun putem kojeg se prijavljuju u sustav. Da bi se učenik mogao prijaviti na natjecanje, potrebno je da je u sustavu već prijavljen njegov odgovorni mentor. Učenik prilikom prijave u sustavu ima obvezu navesti svog odgovornog mentora te mogućnost odabrati svog vanjskog mentora, ako to želi. Učenik, odnosno roditelj ili skrbnik maloljetnog učenika potpisuje Izjavu o davanju suglasnosti za prikupljanje, obradu i objavu osobnih podataka učenika u svrhu sudjelovanja na natjecanju, što je uvjet za pristupanje natjecanju.

Natjecatelji imaju pravo nadležnom povjerenstvu uložiti žalbu na bilo koji aspekt natjecanja u roku od 30 minuta po objavi privremenih ljestvica poredaka. Žalba se podnosi u pisanom obliku, elektronički, te mora biti upućena s elektroničke adrese u

¹ Naslovna stranica, informatika.azoo.hr

sustavu **AAI@EduHr**, a dostavlja se na adresu predsjednika/predsjednice nadležnog povjerenstva. Nadležno povjerenstvo mora odgovore na sve zaprimljene žalbe dostaviti podnositeljima prije objave ljestvice konačnog poretka. U ime nadležnog povjerenstva, odgovor na žalbu njezinom podnositelju šalje predsjednik nadležnog povjerenstva na adresu elektroničke pošte s koje je žalba upućena te je odgovor konačan.

Tri su kategorije koje testiraju različite informatičke vještine:

- Kategorija **Algoritmi**, kojoj ćemo se dodatno posvetiti u ovome radu, provjerava sposobnost rješavanja složenih problema gdje je cilj napisati kôd te pritom pokazati znanje, snalažljivost i kreativnost.
- Kategorija **Razvoj softvera** stavlja natjecatelje u uloge pravih arhitekata računalnih sustava budući da trebaju iskoristiti svoju kreativnost i izgraditi vlastiti softver.
- Za one koji ne vole složene probleme ili velike programske sustave, predviđena je kategorija **Digitalne kompetencije** (za učenike petog i šestog razreda) i **Osnove informatike** (za učenike sedmog razreda i viših razreda osnovne i srednje škole) putem koje se provjerava teoretsko znanje o računalima i informatici koje je uvijek od koristi učenicima.

Infokup natjecanje se sastoji od tri razine: **školske**, **županijske** i najviše **državne razine**. Broj učenika koji se može prijaviti nije ograničen, na školskom natjecanju sudjeluju svi prijavljeni učenici. Svaka sljedeća razina je veći je izazov od prethodne, a broj učenika koji idu na sljedeću razinu sve je manji. Najbolji učenici dobivaju poziv za sudjelovanje u finalu – na državnoj razini. Na državno natjecanje poziva se oko 170 učenika ukupno u svim kategorijama.

1.1 Algoritmi OŠ

Kao što smo već spomenuli, u ovome radu posvetit ćemo se kategoriji **Algoritmi za osnovnu školu** na školskoj i županijskoj razini. U ovoj kategoriji cilj je osmisliti algoritam (niz naredbi koji efikasno rješava neki problem) koji rješava zadani problem. Međutim, zadani problemi su često složeni, pa je potrebno kombinacijom postojećih i osmišljavanjem novih algoritama doći do efikasnog rješenja. Težina zadataka varira ovisno o kategorijama natjecanja te o razini natjecanja.

U kategoriji **Algoritmi** za osnovnu školu natjecati se mogu učenici osnovnih škola od 5. do 8. razreda i to u dvije kategorije, ovisno o programskom jeziku: **LOGO** i **Primjena algoritama OŠ**.

Po završetku natjecanja ispituje se ispravnost učeničkih rješenja zadataka po osmišljenim testnim primjerima koji nose određeni broj bodova, te se u kategoriji **LOGO** zadaci testiraju ručno, a u kategoriji **Primjena algoritama** testiranje se vrši automatiziranim sustavom – evaluatorom.

Na državno natjecanje iz osnovnih škola poziva se do 30 učenika (okvirno) za kategoriju **LOGO** i 34 učenika za kategoriju **Primjena algoritama**.

Natjecanja u kategorijama **LOGO** i **Primjena algoritama OŠ** provode se na računalima. Za svakog natjecatelja na natjecanju pripremljeno je računalo s instaliranim operacijskim sustavom (*Microsoft Windows*) i potrebnim programskim alatima. Ako se učenik želi natjecati na računalu s operacijskim sustavom *Linux*, mentor učenika treba o tome obavijestiti nadležno povjerenstvo najkasnije osam dana prije natjecanja. Ako učenik rješava zadatke na prijenosnom računalu, potrebno je osigurati dodatnu tipkovnicu i miš. Ispravan rad cijelog sustava (i hardverski i softverski aspekt) prije početka natjecanja provjeravaju članovi povjerenstva te učenik pola sata prije početka natjecanja. Natjecateljsko računalo ni na kakav način ne smije moći komunicirati s nekim drugim elektroničkim uređajima, a raspored računala u prostoriji u kojoj se natjecanje odvija mora biti takav da prepisivanje i razgovor između natjecatelja ili ostalih prisutnih osoba bude onemogućen. Natjecatelji od dodatnog pribora smiju imati samo kemijsku olovku i prazan papir dodijeljen od strane nadležnog povjerenstva.

Učenici u kategoriji **LOGO** zadatke mogu rješavati u programskom jeziku FMS Logo. Ovisno o razini natjecanja definira se:

- školska razina: učenici rješavaju tri zadatka različite težine, natjecanje traje dva sata (120 minuta)
- županijska razina: učenici rješavaju četiri zadatka različite težine, natjecanje traje dva sata (120 minuta)
- državna razina: učenici rješavaju četiri zadatka različite težine, natjecanje traje dva i pol sata (150 minuta).

Učenici u kategoriji **Primjena algoritama OŠ** zadatke mogu rješavati u jednom od navedenih programskih jezika:

- Python (v2.6 i više te v3.5 i više)
- C / C++ .

Ovisno o razini natjecanja definira se:

- školska i županijska razina: učenici rješavaju barem tri zadatka različite težine, natjecanje traje dva sata (120 minuta)

- državna razina: učenici rješavaju tri zadatka različite težine, natjecanje traje dva i pol sata (150 minuta).

Bitno je istaknuti da različite zadatke učenik može rješavati u različitim jezicima, vodeći računa o prednostima i nedostacima pojedinog jezika.

Preporučena literatura za pripremanje učenika za natjecanje je *Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*, odnosno sadržaji ishoda i aktivnosti iz domene **Računalno razmišljanje i programiranje** te svi odobreni udžbenici prema katalogu obveznih udžbenika i pripadajućih dopunskih nastavnih sredstava za osnovnu i srednju školu za školsku godinu 2018./2019. Dodatna preporučena literatura za pripremanje učenika navedena je u službenim propozicijama.

Na natjecanju u kategoriji **Algoritmi**, učenici se susreću sa zadacima koji zahtijevaju promišljanje o zadanom problemu i s načinima kako osmišljeno rješenje zapisati u odabranom programskom jeziku. Zadaci prije svega provjeravaju umijeće programiranja. U službenim propozicijama navedene su teme po razredima, koje su smjernice učenicima i njihovim mentorima tijekom priprema za natjecanja. Poznavanje svih tema nije dovoljno za dobar rezultat na natjecanju, ali uvelike može pomoći u fokusiranju priprema na određena područja i znanja.

Službena rješenja zadataka u kategoriji **Primjena algoritama** za osnovnu školu pisana su u programskom jeziku *Python*, verziji 3.5, dok su rješenja zadataka u kategoriji **LOGO** pisana u programskom jeziku FMS Logo.

2 Natjecateljski zadaci u petom razredu

2.1 Primjena algoritama

Potrebna znanja neophodna učenicima petog razreda za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **Primjena algoritama** su:

- osnovni računski operatori (+, *, -, /, //), ostatak pri dijeljenju (%)
- cjelobrojni tip podataka (*integer*, *int()*)
- naredba učitavanja (*input()*) i ispisivanja (*print()*)
- naredba odlučivanja – osnovni oblik (*if*) i *elif* oblik.

Također, teme koje se mogu pojaviti u zadacima su znakovni tip podataka (*char*), naredba ponavljanja – *for* te poneke ugrađene funkcije poput *min()*, *max()*, *abs()* i slično.

Algoritmi i metode za rješavanje natjecateljskih zadataka u petom razredu su:

- *ad hoc* - općeniti naziv za metode rješavanja koje ne zahtijevaju znanje specifičnih algoritama
- osnovni algoritmi - rad sa znamenkama broja
- jednostavne simulacije - implementacija algoritma opisanog u tekstu zadatka.

Najčešće korištena metoda rješavanje proučavanih zadataka je *ad hoc* metoda.

Proučavajući udžbenike za peti razred osnovne škole ([1] i [2]) uočili smo usklađenost s osnovama potrebnih znanja za rješavanje proučavanih zadataka. U udžbenicima naglasak je na unosu, ispisu, naredbi pridruživanja te radu s osnovnim računskim operatorima. Također, zadaci su većim dijelom usklađeni s kurikulumom (*Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*) te samom domenom **Računalno razmišljanje i programiranje**:

- *Nakon pete godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik se koristi programskim alatom za stvaranje programa u kojemu se koristi ulaznim i izlaznim vrijednostima te ponavljanjem.*
- *Nakon pete godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik stvara algoritam za rješavanje jednostavnoga zadatka, provjerava ispravnost algoritma, otkriva i popravlja greške.*

Tekstovi proučavanih zadataka su originalni te prilagođeni učenicima petog razreda, što možemo vidjeti u primjeru zadatka **Prsti** (školska razina 2019. godine):

Mali Krešimir zbraja jednoznamenkaste brojeve uz pomoć prstiju na svojim rukama. Rezultat zbrajanja takvih brojeva on pokaže prstima. Kada zbroj brojeva prelazi deset, koliko je ukupno prstiju na dvije ruke, Krešimir će pokazati svih deset. Napiši program koji za zadana dva jednoznamenkasta broja A i B ispisuje koliko je prstiju Krešimir pokazao prikazujući zbroj tih brojeva.

Zadatak **Prsti** putem pristupačnog teksta zadataka provjerava učeničke vještine unosa i ispisa cijelih brojeva, operatora zbroja te jednostavnu naredbu odluke – *if*. Zadatak nosi ukupno 40 bodova te ako učenik riješi zadatak tako da ispisuje zbroj unesenih brojeva, odnosno bez naredbe *if*, dobiva 32 boda.

Službeno rješenje zadatka:

```
A = int(input())
B = int(input())
if A + B <= 10:
    print(A + B)
else:
    print(10)
```

Sljedeći zadatak koji smo analizirali je zadatak **Mladi** sa županijske razine 2015. godine:

Svi znamo da se Mjesec okreće oko Zemlje. Jedna od posljedica tog okretanja su i mjesečeve mijene. Mijene se neprestano izmjenjuju jedna za drugom točno zadanim redoslijedom. Prvo je na nebu vidljiv mladi Mjesec. Nakon nekog vremena nastupi prva četvrt, a poslije nje Mjesec postane puni. Nakon punog slijedi zadnja četvrt, pa opet mladi Mjesec i tako u krug. Ovo ponavljanje neprestano se odvija zadnjih nekoliko eona godina. Zašto je to baš tako, možeš pronaći na Internetu ili u nekom od astronomskih udžbenika. Promotrimo mjesečeve mijene unutar jednog konkretnog mjeseca u godini. Za svaku od mijena točno znamo datum u mjesecu kada ona započinje. Napiši program koji će za zadani dan u mjesecu, odrediti i ispisati unutar koje mijene se Mjesec trenutno nalazi.

Napomena: U prirodi, mijene se izmjenjuju svakih sedam dana. U ovom zadatku nećemo pratiti taj prirodni zakon, a **eon** je znanstveni naziv za vrlo dugo vremensko razdoblje.

Tekst zadatka je zanimljiv i edukativan te poziva učenike na daljnje istraživanje mjesečevih mijena. Zadatak nosi 40 bodova te potrebno učeničko znanje za riješiti zadatak je korištenje naredbi za unos i ispis, naredbe odluke - *if* (ili *elif* oblik). Također, zadatak je lakše riješiti korištenjem logičkih operatora (operator *and*) kao u službenom

rješenju. No, proučavajući udžbenike te službene propozicije (*Prilog 3: Preporučeno znanje na pojedinoj razini natjecanja*) nismo pronašli korištenje logičkih operatora na nijednom popisu.

Službeno rješenje zadatka:

```
A = int(input())
B = int(input())
C = int(input())
D = int(input())
E = int(input())
if E >= A and E < B:
    print("MLADI")
if E >= B and E < C:
    print("PRVA")
if E >= C and E < D:
    print("PUNI")
if E >= D and E <= 31:
    print("ZADNJA")
if E < A:
    print("ZADNJA")
```

Proučavajući službeno rješenje zadatka te koristeći se saznanjima ([10], *Prilog 3: Preporučeno znanje na pojedinoj razini natjecanja*) da učenici petog razreda koriste naredbu *elif* predložili smo elegantnije te ujedno i vremenski brže rješenje.

Prijedlog rješenja:

```
A = int(input())
B = int(input())
C = int(input())
D = int(input())
E = int(input())
if E >= A and E < B:
    print("MLADI")
elif E < C:
    print("PRVA")
elif E < D:
    print("PUNI")
else:
    print("ZADNJA")
```

Analizirajući zadatke, zainteresirao nas je tekst zadatka **Izbori** (županijska razina 2015. godine) koji mlade ljude uvodi u koncept izbora te samim time uključuje ih u svijet politike. Također, u test primjer autor zadatka stavio je stvarne rezultate s predsjedničkih izbora održanih netom prije.

Zadatak **Izbori**

U Republici Hrvatskoj nedavno su održani predsjednički izbori. U prvome krugu izbora bilo je četvero kandidata koje ćemo abecedno označiti slovima A, B, C i D. Dvoje kandidata koji su u prvome krugu osvojili najviše glasova prošli su u drugi krug izbora, a onaj tko je u drugome krugu osvojio više glasova, pobijedio je na izborima. Tvoj je zadatak, za zadane brojeve osvojenih glasova kandidata u prvome i drugome krugu izbora, pronaći pobjednika izbora.

Zadatak **Leo** iz 2016. godine sa županijske razine logički je zadatak kojega učenici mogu povezati sa stvarnim životnim događajem, a pritom zahtijeva poznavanje naredbi za unos i ispis te naredbu *if*.

Zadatak **Leo**

Leo i Kiki odmaraju se u najboljem svjetskom hotelu. Hotel ima 100 katova, najniži kat (onaj najbliži zemlji) ima oznaku 1, a onaj najviši ima oznaku 100. Između tih katova putuje se liftom (dizalom). Jednog dana, Leo i Kiki susreli su se na X-tom katu i zajedno ušli u lift. Lea će lift odvesti na L-ti kat, a Kikija na K-ti kat. Lift, od trenutnog kata na kome se nalazi, najprije vozi do kata koji mu je bliži, tj. do kojeg treba proći manje katova. Ako su katovi do kojih treba voziti jednako udaljeni, uvijek će najprije odvesti Lea do njegovog kata. Odredi i ispiši odgovore na sljedeća pitanja:

- 1. Koliko katova su Leo i Kiki zajedno putovali u liftu?*
- 2. Kolika je ukupna udaljenost (izražena u katovima) koju je lift prešao tijekom vožnje u kojoj je i Lea i Kikija odvezao do njihovih željenih katova?*
- 3. Je li lift, od X-tog kata, najprije krenuo prema gore („G“) ili prema dolje („D“)?*

Uz prethodno navedena temeljna znanja, zadatak je jednostavnije riješiti korištenjem ugrađene funkcije *abs()* za određivanje udaljenosti izražene u katovima kao u službenom rješenju. No, učenici iz predmeta Matematika uče apsolutnu vrijednost u šestom razredu.

Učenici kako bi točno riješili zadatak, trebaju odrediti bliži kat X-tom katu te ukupnu udaljenost koju je lift prešao tijekom vožnje tako da udaljenosti od X-tog do bližeg kata nadodaju udaljenost od početno bližeg kata do početno daljeg kata u odnosu na X-ti kat. Treće pitanje je najteže te učenici trebaju ovisno o položaju katova L i K u odnosu na kat X, procijeniti je li lift krenuo prema gore ili dolje.

Služeno rješenje zadatka:

```
X = int(input())
L = int(input())
K = int(input())
if abs(X - L) <= abs(X - K):
    zajedno = abs(X - L)
else:
    zajedno = abs(X - K)
print(zajedno)
ukupno = abs(L - K) + zajedno
print(ukupno)
if abs(X - L) <= abs(X - K): # L je bliži ili jednako blizu
X-u od K
    if L > X: # L je više od X
        print("G")
    else: # L je niže od X
        print("D")
else: # K je bliži
    if K > X: # K je više od X
        print("G")
    else: # L je niže od X
        print("D")
```

U prvom retku ispisa treba biti ispisan prirodan broj, odgovor na prvo pitanje iz zadatka, analogno za drugo i treće pitanje treba ispisati u odgovarajućem retku. Ako učenik želi preskočiti odgovor na neko pitanje, tada umjesto točnog odgovora treba ispisati riječ INFORMATIKA.

Zadatak nosi 70 bodova te ga je ukupno riješilo dvoje učenika od 113 kojih ga je rješavalo na natjecanju.

Zadatak koji simpatičnim i primjerenim tekstom za učenike petog razreda provjerava različita učenika znanja je zadatak **Latinum** sa županijske razine 2018. godine:

Zadatak **Latinum**

Latinum je plemeniti metal koji Ferengiji, vrsta iz Star trek serijala, koristi kao sredstvo plaćanja. Postoje tri oblika latinuma: poluga, traka i listić. Pri tome vrijede sljedeći odnosi:

- 1 poluga vrijedi kao 20 traka
- 1 traka vrijedi kao 100 listića.

Pitanje #1: Quark, jedan od Ferengija, odlučio je prodati svoj bar na svemirskoj stanici DS9. On misli da njegov bar vrijedi P poluga, T traka i L listića. Kolika je vrijednost njegovog bara izražena u listićima?

Pitanje #2: Rom i Zek žele kupiti Quarkov bar. Svaki od njih je rekao koliko nudi poluga, traka i listića latinuma za kupnju bara. Ponude su različite vrijednosti. Quark će prodati bar onome koji je ponudio veću ukupnu vrijednost latinuma. Kome je Quark prodao svoj bar?

Pitanje #3: Quark je gubitniku dao još jednu šansu. Ako poveća ponuđeni broj poluga tako da ukupno nadvisi pobjednikovu ponudu, prodat će mu bar. Koliko još najmanje poluga latinuma gubitnik treba ponuditi tako da njegova nova ponuda po vrijednosti bude veća od originalne pobjednikove ponude?

Napiši program koji će na osnovi ulaznih podataka ispisati odgovore na postavljena pitanja.

Zadatak **Latinum** je zadatak u kojemu učenici prvenstveno trebaju odrediti „formulu“ za izračunati vrijednost latinuma izraženu u listićima. Zatim, u drugom pitanju učenici trebaju ponude izraziti kao broj ponuđenih listića te ih usporediti (ili mogu uspoređivati po polugama, trakama i listićima). Najteži dio zadatka je treće pitanje u kojem trebaju odrediti razliku vrijednosti dobivenih u drugom pitanju iskazanu u polugama latinuma te uočiti da na dobivenu vrijednost trebaju nadodati jedan kako bi nadvisili pobjedničku ponudu.

Kao pomoć pri rješavanju zadatka, učenici imaju testne primjere i njihov ispis te opis testnog primjera.

Zadatak nosi ukupno 90 bodova, a točno riješeno prvo pitanje donosi 15 bodova, drugo 30 bodova te treće pitanje nosi 45 bodova.

Službeno rješenje:

```
X = int(input())
Y = int(input())
Z = int(input())
RP = int(input())
RT = int(input())
RL = int(input())
ZP = int(input())
ZT = int(input())
ZL = int(input())
# pitanje #1
print(X * 20 * 100 + Y * 100 + Z)
```

```
# pitanje #2
rom = RP * 20 * 100 + RT * 100 + RL
zek = ZP * 20 * 100 + ZT * 100 + ZL
if rom > zek:
    print("ROM")
else:
    print("ZEK")
# pitanje #3
print (abs(rom - zek) // 2000 +1)
```

2.2 LOGO

Potrebna znanja nužna učenicima petog razreda za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **LOGO** su:

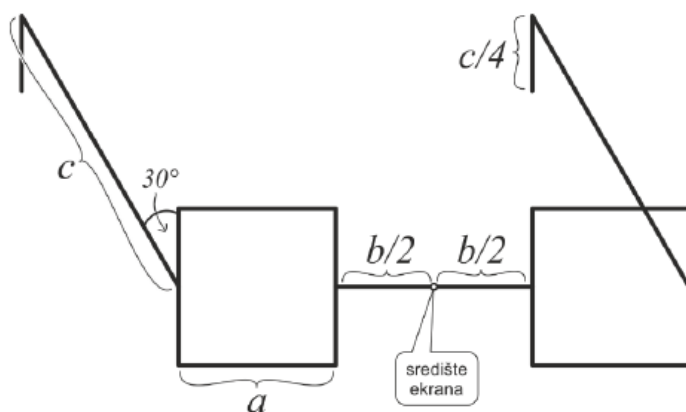
- procedure
- osnovne naredbe za kretanje kornjače (naprijed, nazad, lijevo, desno, podizanje i spuštanje olovke, brisanje ekrana, vraćanje na početak)
- crtanje mnogokuta
- osnovne operacije za kontrolu toka programa (REPEAT, IF)
- osnovne matematičke operacije
- crtanje kružnice
- sve naredbe za kontrolu toka programa (FOR, WHILE...)
- povezivanje logičkih uvjeta (AND, OR)
- osnove rada s koordinatnom grafikom (pamćenje i pomak na poziciju).

Navedene teme su dijelom usklađene s udžbenicima za peti razred ([1] i [2]), osim što se u udžbenicima ne spominje crtanje kružnice, logički operatori i osnove rada s koordinatnom grafikom. Također, neusklađenost smo pronašli s gradivom predmeta Matematika za peti razred. Naime, učenici iz predmeta Matematika mnogokute uče u sedmom razredu osnovne škole (osim trokuta i paralelograma) kao i pravokutni koordinatni sustav.

Jedan od konceptualno tipičnih zadataka na natjecanju petog razreda u kategoriji **LOGO** je zadatak **Naočale** sa županijske razine 2016. godine.

Zadatak **Naočale**

Napišite proceduru NAOCALE :a :b :c koja briše ekran i crta naočale kao na skici desno. Okvir naočala se sastoji od dva kvadrata duljine stranice :a povezanih spojnicom duljine :b. Na okvir su pod kutom od 30° pričvršćene dvije drške kao na skici. Drške su dugačke :c, a njihovi završetci :c/4. Napomena: Parametri :a, :b i :c su cijeli brojevi veći ili jednaki 0 i to takvi da lik ne prelazi granice ekrana. Pozicija lika na ekranu je bitna, a središte ekrana naznačeno je na skici.



Zadatak **Naočale** provjerava učenikovo korištenje osnovnih naredbi za kretanje kornjače: naprijed, nazad, lijevo, desno, podizanje i spuštanje olovke. Također učenik treba izbrisati ekran te ako učenik zaboravi napisati naredbu CS gubi 1 bod.

Potrebno je znati nacrtati kvadrat korištenjem naredbe REPEAT (ne nužno) te poznavati kutove uz usporedne pravce, što učenici petog razreda još nisu učili iz predmeta Matematika. Također, u zadatku su neke duljine stranica zadane razlomcima, koje učenici petog razreda još ne poznaju. U zadatku je zanimljivo što je zadano središte ekrana, odnosno točka u kojoj učenik treba krenuti crtati naočale te vraćati se korištenjem naredbe HOME, također ne nužno.

Za 12 bodova od ukupno 30 dovoljno je nacrtati samo okvir naočala, odnosno naočale bez držaka.

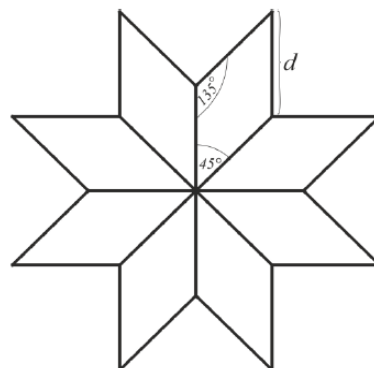
Službeno rješenje zadatka:

```
to naocale :a :b :c
  cs
  lt 90 fd :b/2 rt 90 bk :a/2
  repeat 4 [fd :a lt 90]
  lt 90 fd :a rt 90 fd :a/2 lt 30 fd :c rt 30 bk :c/4
  pu home pd
  rt 90 fd :b/2 lt 90 bk :a/2
  repeat 4 [fd :a rt 90]
  rt 90 fd :a lt 90 fd :a/2 lt 30 fd :c rt 30 bk :c/4
end
```

Zadatak **Uzorak** sa školskog natjecanja 2018. godine je zadatak kojeg možemo povezati s gradivom predmeta Matematika za peti razred za čije rješavanje učenici trebaju znati što je romb i koja su mu obilježja. Osim toga, zadatak **Uzorak** provjerava učenikovo znanje petlje FOR za crtanje redaka i stupaca mreže rombova.

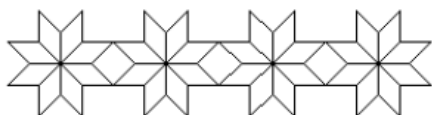
Zadatak **Uzorak**

Kako bi pokazao Marici da cvjetovi ljepše izgledaju kada se crtaju uz pomoć nekih drugih geometrijskih likova, Ivica je odlučio nacrtati mrežu cvjetova. Svaki će pojedini cvijet nacrtati koristeći osam jednakih rombova stranice duljine d kao na skici. Napišite proceduru **UZORAK** $:n :m :d$ koja će nacrtati Ivičinu mrežu od n redova i m stupaca cvjetova. Susjedni se cvjetovi trebaju dodirivati u vrhovima rombova kao što je prikazano u primjerima test podataka.

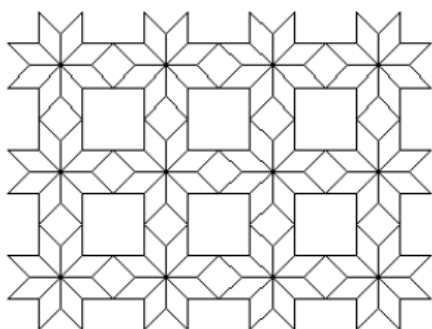


Primjer test podataka:

CS UZORAK 1 4 20



CS UZORAK 3 4 20



Pri rješavanju zadataka učenici trebaju najprije trebaju nacrtati jedan cvijet kojemu su latice cvijeta rombovi sa stranicama duljine d , šiljastim kutom od 45° i tupim kutom od 135° . Kako bi dobili cvijet učenici osam puta trebaju ponoviti crtanje takvih rombova, pritom se okrećući za 45° kako bi rombovi bili pravilno raspoređeni.

Zadatak nosi ukupno 50 bodova, a za nacrtani cvijet učenici dobivaju 15 bodova. Za osvajanje dodatnih 25 bodova, potrebno je m puta ponoviti crtanje cvijeta pomoću petlje FOR. Pritom je važno pravilno se pozicionirati prije crtanja sljedećeg cvijeta u nizu, kako bi se susjedni cvjetovi dodirivali u vrhovima rombova. Za osvajanje preostalih 10 bodova, potrebno je n puta ponoviti crtanje tog reda cvjetova, također pazeći na pravilno pozicioniranje, ne ostavljanje traga prilikom pozicioniranja.

Službeno rješenje zadatka:

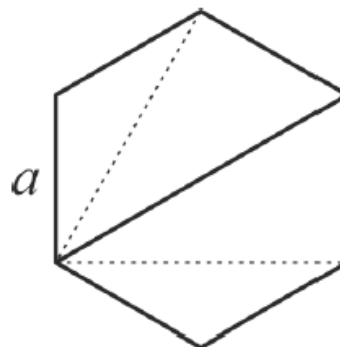
```
to jedan :d
  repeat 8[fd :d rt 45 fd :d rt 135 fd :d rt 45 fd :d]
  pu rt 90 fd :d rt 45 fd :d lt 45 fd :d lt 45 fd :d lt 45
pd
end
to uzorak :n :m :d
  for[i 1 :n 1][
    for[j 1 :m 1][jedan :d]
    repeat :m[pu rt 45 bk :d rt 45 bk :d rt 45 bk :d lt 45
bk :d lt 90 pd]
    pu fd :d lt 45 fd :d rt 90 fd :d lt 45 fd :d pd ]
  end
```

Zadatak **Dijagonale** sa županijske razine 2018. godine nam je zanimljiv jer ga možemo povezati s gradivom predmeta Matematika. Učenici rješavajući zadatak uočavaju pravilnosti vezane za dužine dijagonala pravilnih mnogokuta. Također, bitno je naglasiti da za rješavanje ovog zadatka potrebno koristiti koordinatnu grafiku, koja se pojavljuje u natjecateljskim zadacima već od petog razreda, dok se pravokutni koordinatni sustav, kao što smo već napomenuli, obrađuje u sedmom razredu iz predmeta Matematika.

Zadatak **Dijagonale**

*Mala Lana svoje slobodno vrijeme provodi proučavajući pravilne mnogokute i njihove dijagonale. Od svih dijagonala najviše joj se sviđaju one najdulje, pa je odlučila nacrtati pravilni mnogokut zajedno sa svim njegovim najduljim dijagonalama. Napišite proceduru **DIJAGONALE** :n :a koja crta pravilan :n-terokut duljine stranice :a i sve njegove najdulje dijagonale.*

Na skici je masno otisnuta najdulja dijagonala koja izlazi iz jednog vrha nekog šesterokuta.



Kako bi učenici riješili zadatak **Dijagonale** prvo trebaju odrediti kako pronaći najdulju dijagonalu (ili dijagonale) koja izlazi iz nekog vrha pravilnog mnogokuta. Učenici trebaju uočiti da s obzirom na to da dijagonala spaja dva nesusjedna vrha u pravilnom mnogokutu, da će najdulja dijagonala biti ona koja spaja vrh s njemu najudaljenijim vrhom što učenicima petog razreda može predstavljati problem jer su od mnogokuta do tada iz predmeta Matematika učili trokute i paralelograme te njihove dijagonale.

Također, učenici trebaju uočiti da kod pravilnog mnogokuta s parnim brojem vrhova, najveći broj vrhova kroz koje moraju proći da dođu do najudaljenijeg vrha će biti jednak $n/2$, a kod pravilnog mnogokuta s neparnim brojem vrhova, najudaljeniji vrh neće biti jedinstven te rješavanje zadatka mogu podijeliti na slučaj kad je n paran i kad je n neparan.

Prilikom crtanja dijagonala učenici se, koristeći se koordinatnom grafikom, trebaju pozicionirati u vrh pravilnog mnogokuta, spremati poziciju vrha u neku varijablu, koristiti naredbu SETPOS za pomak na poziciju, pazeći da pritom ostavljaju trag te koristi naredbu SETH te tako osigurati da su se pravilno usmjerili pri povratku na trenutni vrh.

U ovom zadatku, pomoću `int n/2` možemo objediniti oba slučaja i za parni i neparni n te tako u potpunosti riješiti zadatak.

Pri rješavanju zadatka učenicima mogu pomoći različiti primjeri test podataka.

Služeno rješenje zadatka:

```
to dijagonale :n :d
  repeat :n [
    localmake "x pos localmake "h heading
    repeat (int :n/2) [fd :d rt 360/:n]
    setpos :x seth :h
    fd :d rt 360/:n ]
end
```

Crtnanje jednakostraničnog trokuta donosi 6 bodova, dok crtnanje pravilnog mnogokuta gdje je n manji ili jednak pet donosi 18 bodova od ukupno 60 bodova.

3 Natjecateljski zadaci u šestom razredu

3.1 Primjena algoritama

Potrebna znanja učenicima šestog razreda za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **Primjena algoritama** su:

- naredba ponavljanja - *while*
- niz znakova (string)
- jednodimenzionalni niz (polje, lista).

Teme koje se također mogu pojaviti u zadacima su realni tip podataka (*real, double*) te sve što se moglo pojaviti u natjecateljskim zadacima za peti razred.

Algoritmi i metode za rješavanje natjecateljskih zadataka u šestom razredu analogni su kao kod petog razreda uz dodatne:

- osnovni algoritmi - traženje minimuma/maksimuma elemenata u nizu, sortiranje.

Proučavajući udžbenike za šesti razred osnovne škole ([3] i [4]) uočili smo usklađenost gradiva udžbenika s naredbama ponavljanja, te neusklađenost gradiva udžbenika s temama kao što su nizovi znakova te jednodimenzionalni niz. U udžbenicima šestog razreda naglasak je na logičkim operatorima, naredbi *if* i *elif* te petljama *for* i *while*. Zadaci, kao i gradivo, su djelomično usklađeni s kurikulumom (*Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*) te samom domenom **Računalno razmišljanje i programiranje**:

- *Nakon šeste godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik stvara, prati i preuređuje programe koji sadrže strukture grananja i uvjetnoga ponavljanja te predviđa ponašanje jednostavnih algoritama koji mogu biti prikazani dijagramom, riječima govornoga jezika ili programskim jezikom.*
- *Nakon šeste godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik razmatra i rješava složeniji problem rastavljajući ga na niz potproblema.*

Tekstovi zadataka su složeniji nego tekstovi zadataka za peti razred, ali primjereni učenicima šestog razreda. Jedan od tekstualno zanimljivih i poučnih zadataka je zadatak **Rub** sa županijske razine 2015. godine.

Zadatak **Rub**

Klara je otvorila program za obradu teksta. Ovaj put nije se odlučila za MS Word jer postoje jednako dobri, a besplatni softveri kao što su LibreOffice ili OpenOffice, prikladni za besplatan i moćan operativni sustav kao što je Linux Mint ili Ubuntu.

U programu je nacrtala **tablicu** od **R** redaka i **S** stupaca. Potom je sva polja koja se nalaze **na rubu tablice** obojila tamnom bojom. Slika prikazuje obojenu tablicu 3 x 4:

Tvoj je zadatak izračunati koliko je polja Klara obojila.

U uvodnom dijelu zadatka se ukazuje učenicima na besplatne softvere za obradu teksta kao što su LibreOffice ili OpenOffice koji su prikladni za besplatan operativni sustav kao što je Linux Mint ili Ubuntu, te osvještava učenike kako MS Office paket nije besplatan.

Zadatak **Rub** nosi ukupno 40 bodova. Jedan od načina za odrediti broj rubnih polja je da učenici prvotno odrede broj svih polja ($R * S$) te od ukupnog broja polja oduzmu broj polja koja nisu na rubu, nego u unutrašnjosti ($(R - 2) * (S - 2)$).

Službeno rješenje zadatka:

```
r = int(input())
s = int(input())
if r < 3 or s < 3:
    print(r * s)
else:
    print(r * s - (r - 2) * (s - 2))
```

Proučavajući zadatak, uočili smo da se sami račun može provesti i jednostavnije. Naime, broj rubnih polja jednak je dvostrukom broju redaka i stupaca bez 4 rubna polja koja se ponavljaju pri brojanju i redaka i stupaca.

Prijedlog rješenja:

```
r = int(input())
s = int(input())
```

```
print(2*(r + s - 2))
```

Sljedeći zadatak koji smo odabrali za analizu je zadatak **Stella** sa županijske razine 2016. godine zbog matematičkog problema koji je primjeren učenicima šestog razreda.

Zadatak **Stella**

Stella mora vježbati oduzimanje brojeva. Slučajno odabere prirodan broj N (koji ima paran broj znamenki) te ga podijeli po sredini kako bi dobila dva nova broja s jednakim brojem znamenki. Na kraju oduzme manji od većeg tako dobivenog broja. Npr.

$$N = 4283 \longrightarrow 42 \mid 83 \longrightarrow 83 - 42 = 41$$

Napiši program koji će za zadani prirodan broj ispisati razliku dobivenih brojeva.

Za riješiti ovaj zadatak učenicima su potrebna znanja, osim naredbi za unos i ispis, naredba odlučivanja te algoritam određivanja znamenki broja.

U sekciji *Ulazni podaci* definira se da je prirodan broj N veći od 10, manji od 999999 te broj znamenki u broju N je paran. Stoga je autor zadatka u službenom rješenju kôd podijelio u tri dijela ovisno o tome kojem intervalu N pripada, potom odredio lijevi broj (cjelobrojnim dijeljenjem) i desni broj (ostatak pri dijeljenju) te konačno traženu razliku iz teksta zadatka.

Službeno rješenje zadatka:

```
N = int(input())
if N <= 99:
    l = N // 10
    d = N % 10
    if l <= d: # može s naredbom odlučivanja
        razlika = d - l
    else:
        razlika = l - d
# razlika = abs(l - d) # a može i pomoću apsolutne
vrijednosti
if N >= 1000 and N <= 9999:
    l = N // 100
    d = N % 100
    razlika = abs(l - d)
if N >= 100000 and N <= 999999:
    l = N // 1000
    d = N % 1000
```

```
    razlika = abs(1 - d)
print(razlika)
```

Promatrajući zadatak te službeno rješenje, odlučili smo napisati program, koji će vrijediti za svaki prirodan broj, koji će prvo prebrojiti broj znamenki unesenog prirodnog broja te ovisno o broju znamenki odrediti pripadnu razliku.

Prijedlog rješenja:

```
N = int(input())
M = N # kopija prirodnog broja N koja nam služi za
određivanje broja znamenki
broj_znamenki = 0
k = 1
while M > 0:
    M //= 10
    broj_znamenki += 1
for i in range(br//2): # određivanje broja s kojim ćemo
dijeliti prirodan broj N
    k *= 10
print(abs((N // k) - (N % k)))
```

Zadatak nosi ukupno 40 bodova te je prosječna riješenost zadatka 16,95 % .

Jedan od poučnih zadataka na natjecanju je zadatak **Zvijezda** (županijska razina 2018. godine) koji učenicima ukazuje na klasifikaciju zvijezda prema površinskim temperaturama. Također, zanimljivo je što u opisu prvog test podatka ukazano da je riječ o stvarnom podatku površinske temperature zvijezde Sunce.

Zadatak **Zvijezda**

Zvijezda je nebesko tijelo koje isijava svjetlost i toplinu. Naše Sunce je zvijezda.

*Astronomi zvijezde dijele u skupine tj. klasificiraju ih prema površinskoj temperaturi **T** kojom isijavaju. Temperatura se iskazuje u stupnjevima Celzijusovim (°C). Postoji 7 skupina označenih velikim slovima engleske abecede: O, B, A, F, G, K i M. Pri tome vrijede sljedeći odnosi:*

Oznaka	Temperatura
O	površinska temperatura ≥ 29727 °C
B	9727 °C \leq površinska temperatura < 29727 °C
A	7227 °C \leq površinska temperatura < 9727 °C
F	5727 °C \leq površinska temperatura < 7227 °C
G	4927 °C \leq površinska temperatura < 5727 °C
K	3427 °C \leq površinska temperatura < 4927 °C
M	2127 °C \leq površinska temperatura < 3427 °C

Napiši program koji će za zadanu površinsku temperaturu T neke zvijezde, ispisati oznaku skupine u koju će je astronomi klasificirati.

Za riješiti zadatak **Zvijezda** potrebno je znati koristiti naredbe za unos i ispis te složene naredbe odlučivanja.

Službeno rješenje zadatka:

```
T = int(input())
if T >= 29727:
    print("O")
elif T >= 9727:
    print("B")
elif T >= 7227:
    print("A")
elif T >= 5727:
    print("F")
elif T >= 4927:
    print("G")
elif T >= 3427:
    print("K")
elif T >= 2127:
    print("M")
```

Jedan od zadataka koji simulira igru te je samim time potencijalno privlačan učenicima je zadatak **Simpsoni** sa školske razine 2019. godine:

Kang i Konos, dok ne smišljaju planove za osvajanje Zemlje, igraju igru s igraćom kockicom. To je kockica kod koje je svaki broj od jedan do šest napisan na jednoj od njenih strana. Igra traje N rundi, a u svakoj rundi oba igrača po jednom bace kockicu.

Tko će u prvoj rundi prvi bacati kockicu tj. imati **prednost bacanja**, određuje se prije početka igre, a tijekom rundi se može mijenjati. Opišimo pravila igre.

1. Prvo jednom bacimo kockicu da bi odredili prednost bacanja u prvoj rundi. Ako je na kockici pao **paran broj**, tada prednost bacanja dobiva Kang, a inače će je dobiti Konos.

2. U svakoj od N rundi igre svaki igrač po jednom baci kockicu pri čemu prvi baca onaj koji ima prednost bacanja.

- pobjednik runde je igrač koji je na kockici dobio strogo veći broj. Za pobjedu u rundi igrač dobije zbroj bodova palih na kockicama. Igrač s prednošću bacanja se u ovom slučaju ne mijenja;
- ako su oba igrača dobila isti broj, tada nema pobjednika niti tko dobija bodove, a prednost bacanja se mijenja i prelazi na drugog igrača.

Na osnovi zadanih podataka, odgovori na sljedeća pitanja:

- tko je u prvoj rundi prvi bacao kockicu?
- koliko je rundi osvojio Kang, a koliko Konos?
- koliko je bodova na kraju igre imao Kang, a koliko Konos?

Za uspješno riješen zadatak **Simpsoni**, učenicima je potrebno znanje naredbi za ponavljanje i odlučivanje.

Učenici početno trebaju odrediti prednost prvog bacanja jednostavnom provjerom je li prvi učitani broj paran ili neparan, odnosno je li djeljiv s dva ili nije.

Za određivanje broja osvojenih rundi i broja dobivenih bodova učenici trebaju zaključiti da je ključno praćenje igrača na potezu. Igrač na potezu mijenja se svaki put kada su brojevi dobiveni na kockicama jednaki. Ovisno o tome tko je na potezu i odnosa prvog i drugog dobivenog broja učenici trebaju povećati broj osvojenih rundi i dobivenih bodova.

Službeno rješenje zadatka:

```
P = int(input())
N = int(input())
# prednost prvog bacanja, paran-Kang-0, neparan-Konos-1
if P % 2 == 0:
    prednost = 0
    print('KANG')
else:
    prednost = 1
```

```

    print('KONOS')
kangR = kangB = konosR = konosB = 0
for i in range(N):
    K1, K2 = map(int, input().split())
    if K1 > K2:
        if prednost == 0:
            kangR += 1
            kangB += K1 + K2
        else:
            konosR += 1
            konosB += K1 + K2
    elif K2 > K1:
        if prednost == 1:
            kangR += 1
            kangB += K1 + K2
        else:
            konosR += 1
            konosB += K1 + K2
    else:
        if prednost == 0:
            prednost = 1
        else:
            prednost = 0
print(kangR, konosR)
print(kangB, konosB)

```

Ako je učenik napisao program koji simulira navedenu igru bez mijenjanja prednosti bacanja kockice, dobiva 54 boda od ukupno 90 bodova.

3.2 LOGO

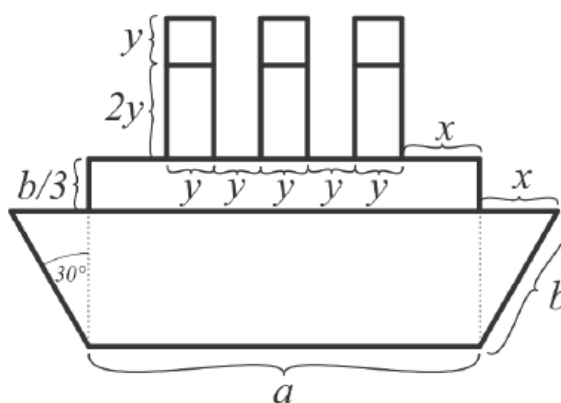
Potrebna znanja neophodna učenicima šestog razreda, osim navedenih u poglavlju 2.2. za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **LOGO** su:

- funkcije (OP)
- ispis teksta na ekran (SHOW, PR)
- ispunjavanje omeđenog dijela ekrana nekom bojom (FILL)
- osnove rada s listama (dubine 1) i riječima.

Navedena znanja su jako dobro usklađena s gradivom promatranih udžbenika za šesti razred ([3] i [4]), osim što se u udžbenicima šestog razreda ne pojavljuju liste.

Zadatak koji nas je privukao poučnim i zanimljivim tekstom je zadatak **Titanic** sa županijske razine 2019. godine :

Titanic, vjerojatno najpoznatiji brod na svijetu, potonuo je 15. travnja 1912. godine. Otisnuo se od obale u Southamptonu, a trebao je za nešto manje od tjedan dana putovanja stići u New York, no na putu mu se ispriječila ledena santa koja je uzrokovala takvu štetu na Titanicu da je ovaj brod dugačak čak 269 metara potonuo za dva sata i četrdeset minuta. Zanimljiva je činjenica da sve scene iz filma Titanic Jamesa Camerona iz 1997. godine smještene u 1912. godini ukupno traju upravo toliko.



Napišite proceduru TITANIC :a :b :n koja će nacrtati sliku Titanica po uputama sa skice. Broj dimnjaka koji se nalaze na brodu određen je varijablom :n. Lijeva i desna strana broda su simetrične. Na skici je primjer s :n=3.

Zadatak ukupno nosi 40 bodova te za riješiti zadatak **Titanic** ključno je da učenici odrede varijable :x i :y pomoću zadane varijable :n. Učenici pomoću zadanog kuta te polovice jednakostraničnog trokuta trebaju uočiti da je :x = :b/2, a da :y računaju tako da dio :a - :b podijele na :n*2-1 jednakih dijelova.

U test podacima vrijednim ukupno 40 % bodova, varijabla :b bit će jednaka 0, dok u test podacima vrijednim ukupno 20 % bodova, varijabla :n bit će jednaka 0 odnosno slika neće imati dimnjake.

Službeno rješenje zadatka:

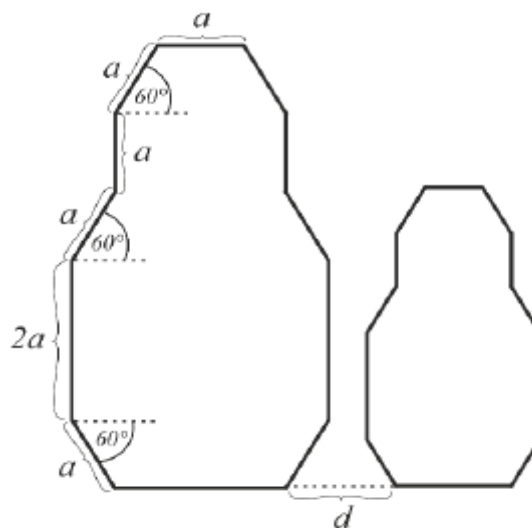
```
to titanic :a :b :n

  lt 30 fd :b rt 120 fd :a+:b rt 120 fd :b rt 60 fd :a
rt 60 fd :b rt 120 fd :b/2 lt 90
  fd :b/3 rt 90 fd :b/2 lt 90
  make "y (:a-:b)/(:n*2-1)
  if not :n = 0 [
    repeat :n-1 [
      repeat 2 [
        fd :y*2 rt 90 fd :y rt 90
      ]
      repeat 2 [
        fd :y*3 rt 90 fd :y rt 90
      ]
      rt 90 fd :y*2 lt 90
    ]
    repeat 2 [
      fd :y*2 rt 90 fd :y rt 90
    ]
    repeat 2 [
      fd :y*3 rt 90 fd :y rt 90
    ]
  ]
  rt 90 fd (abs :y)+:b/2 rt 90 fd :b/3
end
```

Sljedeći zadatak koji smo dodatno analizirali je zadatak **Babuške** sa županijske razine 2016. godine. Zadatak **Babuške** smo izabrali zbog korištenja liste u zadatku te simpatičnog teksta zadatka.

Zadatak **Babuške**

Napišite proceduru **BABUSKE** :l :d koja crta babuške iz liste :l. Dimenzije babuški opisane su na skici. Duljina a (sa skice) i-te babuške odgovara i-tom elementu liste :l. Babuške crtamo redom kako je navedeno u listi :l. Ako je trenutna babuška po dimenzijama manja od prethodne, stavljamo je unutar prethodne babuške. U suprotnom, stavljamo je pokraj uz razmak :d. Babuška se crta unutar druge babuške tako da donja vodoravna stranica bude na sredini iste stranice prethodne babuške.



Napomena: Lista :l neće biti prazna, a sadrži prirodne brojeve. Parametar :d je prirodan broj. Parametri su takvi da lik ne prelazi granice ekrana. Pozicija lika na ekranu nije bitna.

Kako bi si olakšali rješavanje zadatka, učenici mogu napraviti zasebnu proceduru `babuska :a` pomoću osnovnih naredbi za kretanje. Drugi dio rješenja je kreiranje procedure `babuske :l :d` koja koristeći funkciju `FIRST` crta prvu babušku pozivom `procedure babuska (first :l)`. Zatim učenici trebaju prebrojiti koliko ima elemenata zadana lista funkcijom `COUNT`, te pomoću naredbe ponavljanja `FOR` i usporedbe elemenata liste nacrtati na odgovarajuće mjesto babuške.

Dodatna pomoć učenicima pri rješavanju zadatka je detaljno pojašnjenje test primjera ispod teksta zadatka.

Službeno rješenje zadatka:

```
to babuska :a
```

```
  lt 90 fd :a rt 60 fd :a rt 30 fd 2*:a rt 30 fd :a lt
  30 fd :a rt 30 fd :a rt 60 fd :a
```

```
  rt 60 fd :a rt 30 fd :a lt 30 fd :a rt 30 fd 2*:a rt
  30 fd :a rt 60 fd :a rt 90
```

```
end
```

```
to babuske :l :d
```

```

make "max (first :l)
babuska (first :l)
for [i 2 [count :l] 1] [
  if (item :i :l) >= (item :i-1 :l) [
    pu rt 90 fd :max+:d+(item :i :l) lt 90 pd
    make "max (item :i :l)
  ]
  babuska (item :i :l)
]
end

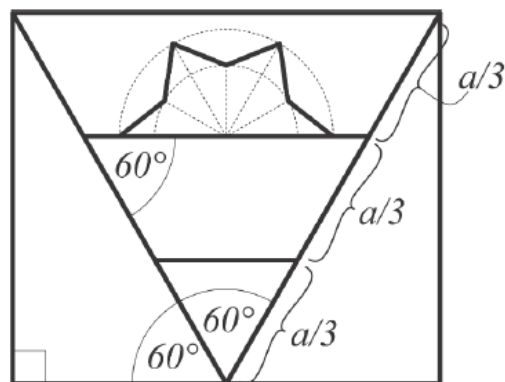
```

Zadatak nosi ukupno 60 bodova te u test primjerima vrijednim 12 bodova svaka će babuška stati unutar prethodne (sve se crtaju jedna unutar druge) ili nijedna babuška neće stati unutar prethodne (sve se crtaju jedna do druge).

Proučavajući zadatke, jedan od zadataka koji se istaknuo je zadatak **Antigva** sa županijske razine 2019.godine. Tekst zadatka je zanimljiv te poučan. Učenici saznaju za Antigvu i Barbudu, gdje su smještene na karti te kako izgleda i što simbolizira zastava te države. Također, rješenje zadatka vizualno je privlačno zbog različito obojenih dijelova zastave.

Zadatak **Antigva**

Antigva i Barbuda država je koja se nalazi između Karipskog mora i Atlantskog oceana, a sastoji se od dva velika i nekoliko manjih otoka. Zastava ove države sastoji se od naopakog jednakokravnog trokuta podijeljenog na bijeli, plavi i crni dio na kojem se nalazi izlazeće sunce. Sunce je simbol svanuća nove ere, crvena boja simbolizira moć ljudi, plava nadu, a crna afričko nasljeđe ljudi koji tamo žive. Kombinacija tih boja predstavlja pješčanu plažu.



Napišite proceduru ANTIGVA :a :b :c :n koja crta zastavu Antigve i Barbude po dimenzijama zadanima na skici. Sunce na zastavi ima :n krakova, no donja polovica je

prekrivena morem. Manja kružnica na slici je polumjera :b, a veća polumjera :c. Lijeva i desna polovica zastave su simetrične. Crvenu boju možete dobiti naredbom SETFC "RED, plavu sa SETFC "BLUE, crnu sa SETFC "BLACK, žutu sa SETFC "YELLOW, a bijelu sa SETFC "WHITE.

Kako bi riješili zadatak **Antigva** učenici moraju, osim korištenja osnovnih naredbi za kretanje, znati se pozicionirati unutar područja slike koje žele ispuniti zadanom bojom korištenjem naredbe FILL. Najteži dio zadatka je crtanje sunca kojeg je najlakše nacrtati korištenjem naredbe SETPOS, odnosno pamćenjem pozicija.

Službeno rješenje zadatka:

```
to antigva :a :b :c :n
  rt 30 fd :a make "poz pos
  bk :a rt 60 fd :a/2 setpos :poz
  lt 60 bk :a lt 120 fd :a/2 make "poz pos
  bk :a/2 rt 60 fd :a setpos :poz
  lt 60 bk :a/2
  rt 60
  repeat 3 [
    fd :a/3 rt 120 ]
  repeat 3 [
    fd 2*:a/3 rt 120 ]
  repeat 3 [
    fd :a rt 120 ]
  fd 2*:a/3 rt 120 fd :a/3 lt 180
  ifelse (not :n = 0) [
    for [i 1 [:n/2+1]] [
      pu fd :c pd
      make "tpoz pos
      if (not :i = 1) [setpos :ppoz]
      setpos :tpoz
    ]
  ]
end
```

```

        make "ppoz pos
        pu bk :c rt 180/:n fd :b pd
        make "tpoz pos
        if (not :i = :n/2+1) [setpos :ppoz]
        setpos :tpoz
        make "ppoz pos
        pu bk :b pd
        rt 180/:n ]
    rt 180/:n*( :n/2-2) ][lt 90]
pu bk 2 pd
setfc "yellow fill
pu fd 2 pd
pu rt 90 fd :a/3 rt 90 fd 3 pd
setfc "black fill
pu bk 3 rt 150 fd :a/3 lt 150 fd 3 pd
setfc "blue fill
pu bk 3 pd rt 90 pu bk 3 pd
setfc "red fill
pu fd :a/3+6 pd
fill
end

```

U test podacima vrijednim ukupno 14 bodova od ukupno 70, neće biti potrebno nacrtati sunce, dok u test podacima vrijednim dodatnih 28 bodova, sunce će izgledati kao polovica ($:n*2$)-terokuta.

4 Natjecateljski zadaci u sedmom razredu

4.1 Primjena algoritama

Učenicima sedmog razreda za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **Primjena algoritama** potrebno je znanje dvodimenzionalnog niza (lista, polje, tablica) uz sve prethodno navedene teme koje su se mogle pojaviti u natjecateljskim zadacima za peti i šesti razred.

Od algoritama i metoda za rješavanje natjecateljskih zadataka u sedmom razredu dodatno se pojavljuju složenije simulacije.

Proučavajući udžbenike za sedmi razred osnovne škole ([5] i [6]) uočili smo usklađenost s osnovama potrebnih znanja. U udžbenicima sedmog razreda naglasak je na tipovima podataka te odgovarajućim im ugrađenim metodama i funkcijama, sekvencijalnom pretraživanju te funkcijama (potprogramima). Zadaci, kao i gradivo, su većinom usklađeni s kurikulumom (*Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*) te samom domenom **Računalno razmišljanje i programiranje**:

- *Nakon sedme godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik razvija algoritme za rješavanje različitih problema koristeći se nekim programskim jezikom pri čemu se koristi prikladnim strukturama i tipovima podataka.*
- *Nakon sedme godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik primjenjuje algoritam (sekvencijalnoga) pretraživanja pri rješavanju problema.*
- *Nakon sedme godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik dizajnira i izrađuje modularne programe koji sadrže potprograme u programskom jeziku.*
- *Nakon sedme godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik se koristi simulacijom pri rješavanju nekog ne nužno računalnoga problema.*

Tekstovi zadataka su složeniji nego dosadašnji, ali primjereni učeničkoj dobi. Jedan od konceptualno novih tipova zadataka je zadatak **Tablica** sa školske razine 2018. godine koji za svoje rješavanje zahtjeva znanje dvodimenzionalnih nizova:

Mirku je uskoro rođendan. Slavko mu želi pokloniti tablicu od R redaka i S stupaca. Tablica je na početku prazna pa ju je Slavko odlučio ukrasiti.

*Prvo će rub tablice (prvi i zadnji redak te prvi i zadnji stupac) **ispuniti zvjezdicama** ('*'). Nakon toga će rub neispunjenog dijela tablice (drugi i predzadnji redak te drugi i predzadnji stupac) **ispuniti slovima M** (veliko slovo 'M'). Nakon toga će rub neispunjenog dijela ponovo ispuniti zvjezdicama pa preostali dio ponovo slovima M te tako sve dok cijela tablica ne bude ispunjena.*

Slavko nikad nije bio dobar u likovnom pa te moli da to učiniš umjesto njega.

U ovom zadatku učenici moraju osmisliti način na koji će ukrasiti tablicu te taj način nije jedinstven. Autor zadatka odlučio se na početku definirati varijable početnog i krajnjeg retka (gore i dolje) te početnog i krajnjeg stupca (lijevo i desno).

Na početku učenici trebaju ispuniti cijelu matricu zvjezdicama, odnosno sva polja u pravokutniku kojem je gornja granica gore, donja dolje, lijeva lijevo i desna desno. Zatim varijable gore i lijevo treba povećati za jedan, a varijable dolje i desno smanjiti za jedan. Nakon toga sva polja unutar novih granica treba ispuniti slovima M. Zatim opet treba promijeniti granice te nastaviti postupak sve dok je gore \leq dolje i lijevo \leq desno.

Službeno rješenje zadatka:

```
znak = "*M"
n, m = map(int, input().split())
mat = [['x'] * m for i in range(n)]
vrsta = 0
gore = 0
dolje = n - 1
lijevo = 0
desno = m - 1
while gore <= dolje and lijevo <= desno:
    for i in range(gore, dolje+1):
        for j in range(lijevo, desno+1):
            mat[i][j] = znak[vrsta]
    vrsta = (vrsta + 1) % 2
    gore += 1
    dolje -= 1
    lijevo += 1
    desno -= 1
for i in mat:
    print(' '.join(i))
```

Osim same ideje, učenicima je nužno znanje korištenja metode `join()` te općenito kreiranje matrice zadanih dimenzija.

Zadatak **Tablica** nosi ukupno 70 bodova, a za ostvariti 40 bodova zadatak treba riješiti za $\mathbf{R} = \mathbf{S}$.

Sljedeći zadatak koji smo proučavali je zadatak **Razdijeli** sa školskog natjecanja 2016. godine. Također, zadatak **Razdijeli** pojavljuje se na školskoj razini natjecanja učenika šestog razreda iste godine. Zadatak nosi 70 bodova za učenike sedmog razreda, dok za učenike šestog razreda u potpunosti točno riješen zadatak nosi 90 bodova.

Zadatak **Razdijeli**

*Za neko natjecanje iz informatike organizator je pripremio tri zadatka. Prvi zadatak nosi A bodova, drugi B , treći C bodova i vrijedi da je $A + B + C = N$. Organizatora zanima na **koliko načina** može odrediti A , B i C tako da svaki zadatak nosi **barem** jedan bod i da je uvijek $A \leq B \leq C$.*

Učenici trebaju riješiti zadatak tako da generiraju sve mogućnosti za uređenu trojku brojeva (A , B , C) koje odgovaraju zadanim uvjetima.

Službeno rješenje zadatka (1. način):

```
N = int(input())
ukupno = 0
for i in range(1, N - 1):
    for j in range(1, N - 1):
        for k in range(1, N - 1):
            if i <= j <= k and i + j + k == N:
                ukupno += 1
print(ukupno)
```

Službeno rješenje zadatka (2. način):

```
N = int(input())
ukupno = 0
for i in range(1, N - 1):
    for j in range(i, N - 1):
        for k in range(j, N - 1):
            if i + j + k == N:
                ukupno += 1
print(ukupno)
```

Proučavajući službena rješenja uočili smo da zadatak možemo riješiti jednostavnije i elegantnije pomoću dvije petlje *for* umjesto tri kao u službenom rješenju.

Prijedlog rješenja:

```
N = int(input())
ukupno = 0
for i in range(1, N - 1):
    for j in range(i, N - 1):
        if i + j <= N - j:
            ukupno += 1
print(ukupno)
```

Prosječna riješenost zadatka učenika šestog razreda je 14,51%, dok kod učenika sedmog razreda prosječna riješenost zadatka je veća i iznosi 39,78%.

Prilikom proučavanja zadataka, privukao nas je zadatak simulacije **Neron** sa školske razine 2015. godine. Zanimljivim tekstom, zadatak **Neron** uvodi učenike u kriptografiju i šifriranje podataka te daje kratak povijesni osvrt na nazive šifri.

Zadatak **Neron**

*Perica je ljubitelj šifriranja podataka i povijesti. On zna da postoji tzv. Cezarova šifra, nazvana po velikom rimskom vojskovođi Cezaru. Nedavno je čuo da postoji i tzv. Neronova šifra, nazvana po slavnom caru Rimskog carstva, Neronu. Neron je tom šifrom latinske riječi **pretvarao** u prirodne brojeve. Opišimo kako za zadani broj možemo otkriti koju latinsku riječ on predstavlja.*

*Osnovu postupka otkrivanja čini **ključna riječ**. Svakom slovu ključne riječi **pridružujemo** njegov redni broj unutar ključne riječi. Zadani prirodni broj pretvaramo u riječ tako da svaku znamenku tog broja **zamijenimo** slovom koje odgovara toj znamenki u ključnoj riječi. Ako neka znamenka nema svoje pridruženo slovo, tada se ta znamenka zamjenjuje slovom „X“. Traženu latinsku riječ dobijemo tako da iz dobivene riječi izbacimo sva slova „X“. Npr.:*

Ključna riječ: OREGON

Prirodan broj: 6329586

1	2	3	4	5	6
O	R	E	G	O	N

6	3	2	9	5	8	6
N	E	R	X	O	X	N

Latinska riječ koja odgovara zadanom prirodnom broju:

NERON.

Napiši program koji za zadanu ključnu riječ i zadani prirodan broj, ispisuje traženu latinsku riječ.

Jedna od ideja kako bi učenici mogli riješiti zadataka **Neron** je da ključnu riječ i zadani broj učitaju kao stringove te svaki znak u broju zamijene pripadajućim slovom na odgovarajućem mjestu u ključnoj riječi.

Također, učenici mogu zanemariti izbacivanje znaka X iz nove riječi ako prilikom kreiranja te riječi zanemare sve znamenke koje su veće od duljine ključne riječi ili jednake nuli.

Službeno rješenje zadatka:

```
kljucna = input()
N = input()
rijec = ""
for i in N:
    if int(i) <= len(kljucna) and int(i) != 0:
        rijec += kljucna[int(i) - 1]
print(rijec)
```

Zadatak **Neron** nosi ukupno 90 bodova, a za dobiti polovicu ukupnih bodova učenici ne trebaju izbrisati slova „X“ iz dobivene riječi.

Sljedeći zadatak koji smo promatrali je zadatak **Rok** sa županijske razine 2018. godine. Zadatak ispituje bitna učenička znanja kao što su naredba odlučivanja, naredba ponavljanja te algoritam traženje minimalnog elementa.

Zadatak Rok

Krešimir je u svojoj lokalnoj trgovini kupio N voćnih jogurta označenih brojevima od jedan do N . Poslije se sjetio da nije provjerio mogu li se ti jogurti uopće jesti, tj. je li im istekao rok uporabe.

*Krešimir zna koji je danas datum i koji datum piše na svakom od N jogurta koje je kupio. Datum je opisan svojim **danom**, **mjesecom** i **godinom**. Jogurt se može pojesti ako je današnji datum u vremenu prije ili je isti kao datum zapisan na jogurtu, a ako nije tada se jogurt ne smije jesti jer se pokvario.*

Krešimir želi za svaki od N jogurta ispisati je li jestiv (DA) ili nije jestiv (NE) na današnji datum. Isto tako želi odrediti datum zapisan na jogurtu kojem je prvom istekao rok uporabe ili će mu tek isteći.

Ideja za uspješno riješiti zadatak je da svaki od N datuma treba usporediti s današnjim datumom i za svakog ispisati odgovarajuću poruku. To se može postići uspoređujući redom godine, pa mjeseci i na kraju dane. Uspješno riješen ovaj dio zadataka učenicima donosi 20 od ukupno 40 bodova.

Za drugi dio zadatka učenici trebaju pronaći datum koji ima najmanju godinu, pa tada najmanji mjesec i na kraju najmanji dan te za uspješno napisan drugi dio zadataka ostvaraju preostalih 20 bodova.

Službeno rješenje zadatka:

```
N = int(input())
D, M, G = map(int, input().split())
L = [2020, 12, 30]
for i in range(N):
    Di, Mi, Gi = map(int, input().split())
    ok = 1
    if Gi < G:
        ok = 0
    elif Gi == G:
        if Mi < M:
            ok = 0
        elif Mi == M:
            if Di < D:
                ok = 0
    if ok:
        print('DA')
    else:
        print('NE')

    if [Gi, Mi, Di] < L:
        L = [Gi, Mi, Di]

print(L[2], L[1], L[0])
```

4.2 LOGO

Potrebna znanja učenicima sedmog razreda, osim navedenih u prethodnim poglavljima za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **LOGO** su:

- sortiranje liste
- rad s listama većih dubina
- Pitagorin poučak
- crtanje elipse
- koordinatna grafika
- jednostavnije rekurzije.

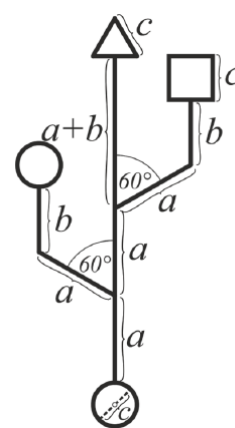
U promatranim udžbenicima ([5] i [6]) se ne obrađuje većina prethodno navedenoga gradiva iz službenih propozicija ([10], *Prilog 3: Preporučeno znanje na pojedinoj razini natjecanja*). Navedeno gradivo se obrađuju u udžbenicima za osmi razred iz predmeta Informatika. Također, prethodno navedeno gradivo za učenike sedmog razreda nije usklađeno s gradivom iz predmeta Matematika budući da Pitagorin poučak učenici uče u 8. razredu, a pojam elipse u 3. razredu srednje škole. Ipak, istaknuli bismo da nismo pronašli zadatke s crtanjem elipse i primjenom Pitagorinog poučka u proučavanim zadacima na školskoj i županijskoj razini.

Jedan od konceptualno standardnih zadataka na školskoj razini je zadatak **USB** (školska razina 2018. godine). Osim što provjerava učenikovo korištenje osnovnih naredbi za pomicanje kornjače i crtanje kružnice naredbom **CIRCLE**, zadatak **USB** je poučan i zanimljiv.

Zadatak **USB**

Jeste li se ikad pitali koje je značenje USB logotipa? Naime, kao što je vidljivo na skici, logotip podsjeća na trozubac, oružje boga Posejdona. Tvorci USB standarda tim su simbolom i različitim oblicima na krajevima zubaca htjeli ukazati na njegovu veliku moć da radi s raznim vanjskim uređajima i time nam uvelike olakša korištenje računala.

*Napišite proceduru **USB :a :b :c** koja crta logotip kao na skici. Trokut je jednakostraničan, a okomite dužine koje dodiruju trokut i kvadrat spajaju se s polovištima njihovih stranica. Duljine stranica kvadrata i trokuta te **promjeri** kružnica iznose :c.*



Službeno rješenje zadatka:

to usb :a :b :c

```

fd :a lt 60 fd :a rt 60 fd :b pu fd :c/2 pd circle :c/2
pu home pd
fd :a*2 rt 60 fd :a lt 60 fd :b lt 90
fd :c/2 repeat 4 [rt 90 fd :c]
pu home pd
fd :a*3 fd :b rt 90 fd :c/2 lt 120 fd :c lt 120 fd :c lt
120 fd :c/2
pu home bk :c/2 pd circle :c/2
end

```

Zadatak nosi 50 bodova te za osvajanje 20 bodova dovoljno je nacrtati samo okomite i kose linije lika.

Zadatak koji nas je prvotno privukao zanimljivim tekstom je zadatak **Par** sa školske razine 2019. godine. Zadatak **Par** tekstom povezuje pisanje procedure s gradivom predmeta Hrvatski jezik.

Zadatak **Par**

Stjepan se u zadnje vrijeme zainteresirao za parove slova, tzv. digrafe. No, budući da u hrvatskom jeziku postoje samo tri digrafa (lj, nj i dž), Stjepanu ih je ubrzo postalo dosadno promatrati pa je odlučio dodatno proučavati sve moguće kombinacije dva slova (kao što su “aa”, “ab”, “ba”, itd.). Stjepana sada za neke dvije riječi zanima koliko postoji različitih digrafa koji se nalaze u obje. Kako Stjepanu brojanje nije jača strana, zamolio vas je za pomoć.

Napišite proceduru PAR :a :b koja vraća ukupan broj različitih digrafa koji se pojavljuju i u riječi :a i u riječi :b. Pri tome nije nužno da se digraf nalazi na istoj poziciji u obje riječi.

Kako bi učenici riješili zadatak **Par** potrebno je znati rješavati zadatke s listama dubine jedan. Također, prilikom kretanja po slovima zadanih riječi potrebne su im dvije petlje FOR, naredba COUNT koja broji slova u riječi te naredba IF i logički operator AND pomoću kojih pronalaze iste parove slova u riječi.

Pomoću naredbi WORD i MEMBERP učenici trebaju smjestiti parove slova u listu (ako se par već nalazi u listi, ne spremaju ga) te konačno vratiti broj elemenata te liste pomoću naredbe COUNT.

Primjeri test podataka sadrže objašnjenja koja učenicima dodatno mogu olakšati rješavanje zadatka.

Službeno rješenje zadatka:

```
to par :r1 :r2
```

```

make "rez []
for [i 1 [(count :r1) - 1] 1] [
  for [j 1 [(count :r2) - 1] 1] [
    if and ((item :i :r1) = (item :j :r2)) ((item :i+1
:r1) = (item :j+1 :r2)) [
      make "w word (item :i :r1) (item :i+1 :r1)
      if not memberp :w :rez [make "rez lput :w :rez]
    ]
  ]
]
op (count :rez)
end

```

Zadatak nosi ukupno 50 bodova, a za ostvariti 20 bodova dovoljno je da procedura radi za dvije jednake riječi unutar kojih neće postojati dva ista slova.

Sljedeći zadatak koji smo izabrali je zadatak **Grančica** (županijska razina 2016. godine) u kojem učenici trebaju rekurzivno nacrtati grančicu sa zadanim brojem izdanaka. Kao što smo rekli na početku ovog poglavlja, učenici sedmog razreda jednostavne rekurzije ne uče na redovnoj nastavi predmeta Informatika, ali se nalazi na popisu gradiva iz službenih propozicija ([10], *Prilog 3: Preporučeno znanje na pojedinoj razini natjecanja*).

Zadatak **Grančica**

Napišite proceduru GRANCICA :n :d :k :kut koja ne briše ekran i crta grančicu kao na skici. Grančica se sastoji od :n manjih izdanaka pravilno raspoređenih na stabljici duljine :d. Izdanci se u odnosu na stabljiku crtaju pod kutom od :kut stupnjeva i to naizmjenice (lijevo pa desno) kao što je prikazano na skici. Svaki se izdanak crta kao manja grančica čija je stabljika :k puta kraća, a broj izdanaka se smanjio za dva. Ovaj se postupak ponavlja sve dok se ne iscrtaju i najmanje grančice koje više nemaju izdanaka.



Napomena: Parametar :n je paran broj veći ili jednak 0.

Za uspješno riješiti zadatak **Grančica**, ponavljamo, učenici trebaju znati jednostavne rekurzije. Na početku učenici trebaju inicijalizirati rekurziju te pomoću petlje FOR, naredbe REMAINDER i poziva procedure `grancica (:n-2) (:d/:p) :p` rekurzivno nacrtati izdanke na naizmjeničnim stranama grančica.

Službeno rješenje zadatka:

```
to grancica :n :d :p :kut
  if :n=0 [fd :d bk :d stop]
  for [i 1 :n] [
    localmake "k 1
    if (remainder :i 2)=1 [make "k -1]
    fd :d/:n rt :k*:kut grancica (:n-2) (:d/:p) :p
:kut lt :k*:kut
  ]
  bk :d
end
```

Zadatak nosi ukupno 70 bodova, a za ostvariti 28 bodova zadatak treba ispravno raditi kada je broj izdanaka manji ili jednak 4.

Na županijskom natjecanju 2016. godine, zadatak **Grančica** u potpunosti točno riješilo je troje od ukupno 50 učenika.

5 Natjecateljski zadaci u osmom razredu

5.1 Primjena algoritama

Učenicima osmih razreda za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **Primjena algoritama** potrebno je poznavanje svih tema koje su se mogle pojaviti u natjecateljskim zadacima za prethodne razrede. Također, na ovoj razini natjecanja mogući su zadaci u kojima značajnu ulogu igra vremensko ograničenje, stoga je potrebno osmisliti ne samo točan, nego i dovoljno efikasan algoritam u smislu broja operacija.

Proučavajući udžbenike za osmi razred osnovne škole ([7] i [8]) uočili smo usklađenost s osnovama potrebnih znanja. U udžbenicima osmih razreda naglasak je na sortiranju niza, algoritmima sortiranja te rekurzijama. Zadaci, kao i gradivo, većim su dijelom usklađeni s kurikulumom (*Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*) te samom domenom **Računalno razmišljanje i programiranje**:

- *Nakon osme godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik identificira neki problem, stvara program za njegovo rješavanje, dokumentira rad programa i predstavlja djelovanje programa drugima.*
- *Nakon osme godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik prepoznaje i opisuje algoritam sortiranja, primjenjuje jedan algoritam sortiranja za rješavanje zadanog problema u programskom jeziku.*
- *Nakon osme godine učenja predmeta Informatika u domeni Računalno razmišljanje i programiranje učenik prepoznaje i opisuje mogućnost primjene rekurzivnih postupaka pri rješavanju odabranih problema te istražuje daljnje mogućnosti primjene rekurzije.*

Tekstovi zadataka su primjereni učenicima osmih razreda. Zadatak koji nam se svidio zbog svoje poveznice s drugim gradivom iz predmeta Informatika je zadatak **Preciznost** sa školske razine 2015. godine. Naime, u zadatku se ukazuje učenicima na nepreciznost prilikom pohrane i zapisa realnih brojeva u računalima na primjeru u programskom jeziku *Python*. Zadatak **Preciznost** nosi ukupno 40 bodova.

Zadatak **Preciznost**

Ako u svom programskom jeziku napišeš usporedbu $0.1 + 0.2 == 0.3$ (ili $0.1 + 0.2 = 0.3$) rezultat će vjerojatno biti *False*! Drugim riječima, računalo misli da su broj $0.1 + 0.2$ i broj 0.3 različiti, iako očito nisu. Kako je to moguće?

Razlog je nepreciznost u pohrani i računanju s realnim brojevima. Većinu realnih brojeva računala ne mogu posve točno pohraniti jer ti brojevi imaju beskonačno mnogo znamenaka (binarnih, naravno) iza decimalne točke. Prilikom računanja s takvim brojevima, nepreciznost se povećava. Primjerice, na autorovom računalu Python ispisuje sljedeće:

```
>>> 0.1 + 0.2
```

`0.30000000000000004` Zbog toga se realni brojevi najčešće uspoređuju tako da se proglašavaju **jednakima** ako su “skoro jednaki”, tj. **ako se međusobno razlikuju za manje od 0.000001**. Ako to ne vrijedi, uspoređuju se na uobičajen način. Napiši program koji unosi realne brojeve **A**, **B** i **C** te uspoređuje broj **A + B** s brojem **C** na ovaj način.

U jedini redak ispiši rezultat usporedbe broja **A + B** i broja **C**: znak ‘=’ ako su jednaki, znak ‘>’ ako je prvi broj veći, a znak ‘<’ ako je drugi broj veći.

Na početku, učenici trebaju znati učitati realni tip brojeva pomoću metode `float()` koja pretvara string ili cijeli broj u realan broj.

Kako bi učenici provjerili razlikuju li se broj **A + B** i broj **C** za manje od 0.000001, najlakše je pogledati je li njihova razlika po apsolutnoj vrijednosti manja od 0.000001. Dakle, ako je $|A + B - C| < 0.000001$, učenici trebaju ispisati ‘=’, a inače pomoću naredbe `elif` provjeriti je li $A + B > C$ te ako je ispisati ‘>’, a ako nije trebaju ispisati ‘<’.

Službeno rješenje zadatka:

```
a = float(input())
b = float(input())
c = float(input())
if abs(a + b - c) < .000001:
    print ('=')
elif a + b < c:
    print ('<')
else:
    print ('>')
```


Sljedeći zadatak koji smo odabrali je zadatak **Dobrilo** sa županijske razine 2018. godine. Osim što zadatak zahtijeva od učenika poznavanje rada sa stringovima i drugim strukturama podataka, simpatičnim tekstom podučava učenike općoj kulturi.

Zadatak **Dobrila**

Kuna je novčana jedinica kojom plaćamo robu i usluge. Pored kovanica, postoji i 7 vrsta novčanica različite vrijednosti. Na svakoj od novčanica nalazi se neki povijesni lik prema kojem možemo prepoznati o kojoj se novčanici radi.

Jakov je u sobi, u nekoliko različitih skrovišta, sakrio nepoznat broj novčanica koje Filip

vrijednost	izgled	lik
10		DOBRILA
20		JELAČIĆ
50		GUNDULIĆ
100		MAŽURANIĆ
200		RADIĆ
500		MARULIĆ
1000		STARČEVIĆ

treba pronaći. Kada u nekom od skrovišta Filip pronađe novčanice, naglas izgovora likove koji se nalaze na njima. Npr. ako kaže „DOBRILA“ znači da je u skrovištu pronašao novčanicu od 10 kuna. Ako je u skrovištu našao X ($2 \leq X \leq 1000$) istih novčanica, tada će zajedno s imenom lika s novčanice izgovoriti i broj X . Npr. ako kaže „3JELACIC“ znači da je pronašao tri novčanice od 20 kuna što je ukupno 60 kuna.

Jakov je tijekom potrage pratio i redom zapisivao sve što Filip izgovora stavljajući znak „#“ između izjava kako bi lakše odredio ukupnu vrijednost pronađenih novčanica.

Napiši program koji će na osnovi zadanog izgovorenog slijeda Filipovih izjava ispisati ukupnu vrijednost novčanica.

Zadatak se može riješiti na razne načine igrajući se sa strukturama podataka te je autor zadatka odlučio riješiti zadatak **Dobrila** pomoću rječnika.

Službeno rješenje zadatka:

```
S = input()
L = S.split('#')

novcanice = {'DOBRILA':10, 'JELACIC':20, 'GUNDULIC':50}
novcanice.update({'MAZURANIC':100, 'RADIC':200,
'MARULIC':500, 'STARCEVIC':1000})
```

```

likovi = novcanice.keys()
ukupno = 0
for i in L:
    for j in likovi:
        if i.find(j) == 0:
            ukupno += novcanice[j]
        if i.find(j) > 0:
            ukupno += int(i[:i.find(j)])*novcanice[j]
print(ukupno)

```

Analizirajući udžbenike za osnovnu školu iz predmeta Informatika te službenih propozicija ([10], *Prilog 3: Preporučeno znanje na pojedinoj razini natjecanja*) nismo pronašli rad s rječnicima, no, kao što smo već rekli, zadatak se mogao riješiti i na druge načine na primjer pomoću listi ili liste listi.

Prijedlog rješenja pomoću liste listi:

```

from random import *
S = input()
L = S.split('#')
M = [['DOBRILA', 10], ['JELACIC', 20], ['GUNDULIC', 50],
      ['MAZURANIC', 100], ['RADIC', 200], ['MARULIC', 500],
      ['STARCEVIC', 1000]]
ukupno = 0
for i in L:
    for j in M:
        if i.find(j[0]) == 0:
            ukupno += j[1]
        if i.find(j[0]) > 0:
            ukupno += int(i[:i.find(j[0])])*j[1]
print(ukupno)

```

Zadatak nosi ukupno 40 bodova te u primjerima vrijednima 20 bodova Filip će izgovarati samo osnovni oblik opisa novčanice.

Pri rješavanju zadatka, opis test podataka dodatno olakšava učenicima rješavanje zadatka.

Zadatak **Dobrila** u potpunosti je točno riješilo 14 od 53 učenika koji su ga rješavali.

Sljedeći zadatak koji smo odabrali je zadatak **Cesta** sa školske razine 2016. godine. Zanimljivo je da se zadatak **Cesta** pojavljuje, osim u zadacima za osmi razred gdje nosi 70 bodova, u natjecateljskim zadacima za sedmi razred gdje nosi ukupno 90 bodova.

Zadatak **Cesta**

*Uz cestu je N gradova označenih brojevima od 1 do N redom kojim se nalaze uz cestu. Poznate su udaljenosti (u kilometrima) između svakih dvaju susjednih gradova na cesti. Dora želi stići iz grada **A** u grad **B** i zanima je udaljenost tih dvaju gradova.*

Na primjer, slika prikazuje cestu iz drugog test primjera niže:



Sa slike zaključujemo da udaljenost gradova 2 i 5 iznosi 30 kilometara.

Jedna ideja kako riješiti zadatak **Cesta** je da učenici zadane udaljenosti susjednih gradova spreme u listu. Kada unesu brojeve A i B , najprije ih trebaju međusobno zamijeniti ako je A veći od B , tako da A bude manji od B . Potom petljom *for* trebaju proći po listi, i to po onim udaljenostima koje se nalaze između A i B , počevši od udaljenosti gradova A i $A+1$, pa sve do udaljenosti gradova $B-1$ i B , usput pribrajujući te udaljenosti u varijablu ukupno koju na kraju trebaju ispisati.

Službeno rješenje zadatka:

```
N = int(input())
cesta = list(map(int, input().split()))
A, B = map(int, input().split())
if A > B:
    A, B = B, A
    #print(A, B)
ukupno = 0
for i in range(A-1, B-1):
    ukupno += cesta[i]
print(ukupno)
```

Od 97 učenika koji su sudjelovali na natjecanju za osmi razred, potpuno točno riješen zadatak ima 46 učenika, dok je 35 od 95 učenika sedmog razreda u potpunosti točno riješilo zadatak **Cesta**.

5.2 LOGO

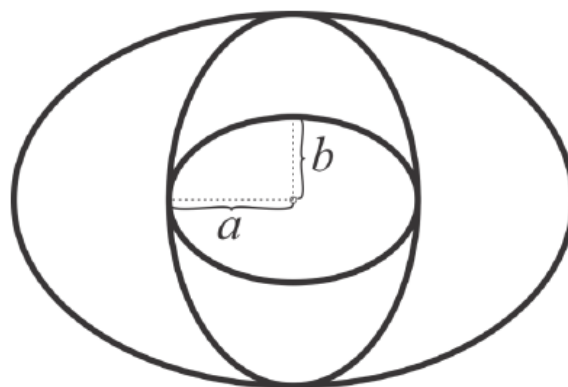
Učenicima osmog razreda, osim navedenih znanja u prethodnim poglavljima, za riješiti natjecateljske zadatke na školskoj i županijskoj razini u kategoriji **LOGO** potrebno je znanje crtanja kružnih lukova i lukova elipse.

Potrebna znanja većinom su usklađena s osnovama gradiva promatranih udžbenika ([7] i [8]) za osmi razred iz predmeta Informatika.

Zadatak **Hiperoko** sa školskog natjecanja 2019. godine zanimljiv je jer, osim simpatičnog teksta, učenici u zadatku trebaju crtati elipse što, kao što smo već istaknuli u poglavlju 4.2., kao pojam još nisu učili iz predmeta Matematika.

Zadatak **Hiperoko**

Jedno od čuda prirode svakako je mačji vid. Za razliku od ljudskih, mačje oči prilagođene su tami te im omogućavaju dovoljno dobar vid za lov pri najslabijem osvjetljenju. Zanimljivo, trokutaste mace iz prethodnog zadatka imaju posebne oči koje nazivamo hiperočima. Za razliku od običnog oka, hiperoči se sastoje od niza elipsi kao što je prikazano na skici.



*Napišite proceduru **HIPEROKO** :a :b :l koja crta jedno takvo hiperočko. Vodoravna poluos najmanje elipse iznosi :a piksela dok okomita poluos najmanje elipse iznosi :b piksela. Omjer duljina male i velike poluosi je jednak u svim elipsama, a elipse se crtaju sve dok su im duljine obje poluosi manje ili jednake :l piksela. Kao na skici, mala i velika os mijenjaju orijentaciju iz elipse u elipsu. Primjerice, ako je u nekoj elipsi mala os vodoravna, u sljedećoj elipsi (ona u koju je prethodna elipsa upisana) će mala os biti okomita i zajednička s velikom osi prethodne elipse. U sljedećoj elipsi mala os će opet biti vodoravna i tako naizmjenice do najveće elipse.*

Da bi učenici riješili zadatak **Hiperoko** prvotno trebaju odrediti omjer duljina male i velike poluosi. Zatim pomoću petlje **WHILE** trebaju crtati elipse naredbom **ELLIPSE** održavajući omjer duljina velike i male poluosi jednakim u svim elipsama, pritom mijenjajući orijentaciju elipsi sve dok su im duljine obje poluosi manje ili jednake zadanom limitu.

Službeno rješenje zadatka:

```

to hiperoko :a :b :limit
  ifelse :a>:b [localmake "k :a/:b] [localmake "k :b/:a]
  while [and :a <= :limit :b <= :limit] [
    ellipse :a :b
    ifelse (:a > :b) [make "b :a*:k] [make "a :b*:k]
  ]
end

```

Zadatak nosi ukupno 50 bodova, a za ostvariti 10 bodova dovoljno je nacrtati jednu elipsu, dok za ostvariti 15 bodova hiperoko će se sastojati od dvije elipse.

Sljedeći zadatak koji smo odabrali je zadatak **Kalodont** sa školskog natjecanja 2018. godine. Zadatak zanimljivim testom ukazuje učenicima kako se igra igrica „Kalodont” koja je primjerena učeničkoj dobi. Osim zanimljivog teksta, zadatak provjerava učeničko znanje rada s listama dubine jedan, rad s riječima te korištenje naredbe IF.

Zadatak **Kalodont**

Mirko i Slavko igraju popularnu igru „Kalodont”. Igra se igra tako da jedan od igrača kaže neku riječ, a drugi igrač u sljedećem koraku odgovara riječju koja počinje na dva slova kojima završava riječ koju je rekao igrač koji je prije toga bio na potezu. Igra se u krug sve dok neki od igrača ne može smisliti riječ koja počinje na zadana slova. Pobjednik je igrač koji je izrekao posljednju riječ.

(Zato se i igra zove „Kalodont” - najčešće završava tako da netko kaže riječ koja završava na KA na što iduća osoba kaže KALODONT i time igra završava jer ne postoji riječ koja počinje na NT.)

No, Mirko i Slavko nisu baš najbolji u ovoj igri pa često ne primijete kad jedan od njih pogriješi u poštivanju pravila, odnosno kad jedan od njih ne kaže riječ koja počinje slovima kojima završava prethodna riječ.

Napišite funkciju KALODONT :l koja vraća listu koja se sastoji od dva broja: prvi govori koliko je puta pogriješio Mirko, a drugi koliko je puta pogriješio Slavko. Lista :l sadrži riječi koje su, redom, izgovorili Mirko i Slavko prilikom jedne igre. Mirko uvijek počinje prvi.

Napomena: *Da bi rješenja osvojila bodove, potrebno je vratiti listu koja se sastoji od točno dva prirodna broja.*

Jedna od ideja kako riješiti zadatak je da učenici najprije uoče da je Mirko izgovorio sve riječi koje se u listi nalaze na neparnom mjestu, a Slavko sve riječi koje se nalaze na parnom mjestu. Također, na početku učenici trebaju stvoriti dvije varijable koje će služiti kao brojači pogrešaka za Mirka i Slavka. Uz pomoć neke petlje i brojača mogu proći sve

riječi u listi te za svaku riječ provjeriti nalazi li se na parnom ili neparnom mjestu tako da koristeći naredbu REMAINDER provjere ostatak pri dijeljenju njezinog rednog broja s 2.

Kad učenici otkriju tko je izgovorio trenutnu riječ, potrebno je provjeriti zadovoljava li izgovorena riječ pravila igre, tj. počinje li onim slovima kojima prethodna riječ završava. Za dobivanje prvog i zadnjeg slova učenici mogu koristiti naredbe FIRST i LAST, a za dobivanje predzadnjeg i drugog slova u riječi mogu koristiti naredbu ITEM i zadati joj odgovarajući indeks i duljinu riječi. Drugi mogući način za dobivanje predzadnjeg slova u riječi je korištenje izraza LAST (BUTLAST), a za dobivanje drugog slova po redu FIRST (BUTFIRST).

Nakon što učenici uspješno dobiju slova na odgovarajućim pozicijama, korištenjem naredbe IF ih mogu usporediti. Ako zadnje slovo prethodne riječi nije jednako prvom slovu trenutne ili predzadnje slovo prethodne riječi nije jednako drugom slovu trenutne, učenici znaju da je osoba koja je izgovorila trenutnu riječ pogriješila, pa trebaju povećati brojač pogrešaka za tu osobu. Nakon što prođu sve riječi u listi, uz pomoć naredbi PRINT i LIST učenici ispisuju te brojače.

Pri rješavanju zadatka, dodatna pomoć učenicima je detaljno objašnjenje ispisa test primjera.

Službeno rješenje zadatka:

```
to kaladont :l
  localmake "Mirko 0
  localmake "Slavko 0
  foreach :l [
    if #=count :l [
      pr sentence :Mirko :Slavko
      stop
    ]
    if not (word (last bl ?) (last ?))=(word (first first
?rest) (first bf first ?rest)) [
      ifelse (remainder # 2)=0 [
        make "Mirko :Mirko+1
      ] [

```

```

        make "Slavko :Slavko+1
    ]
]
]
end

```

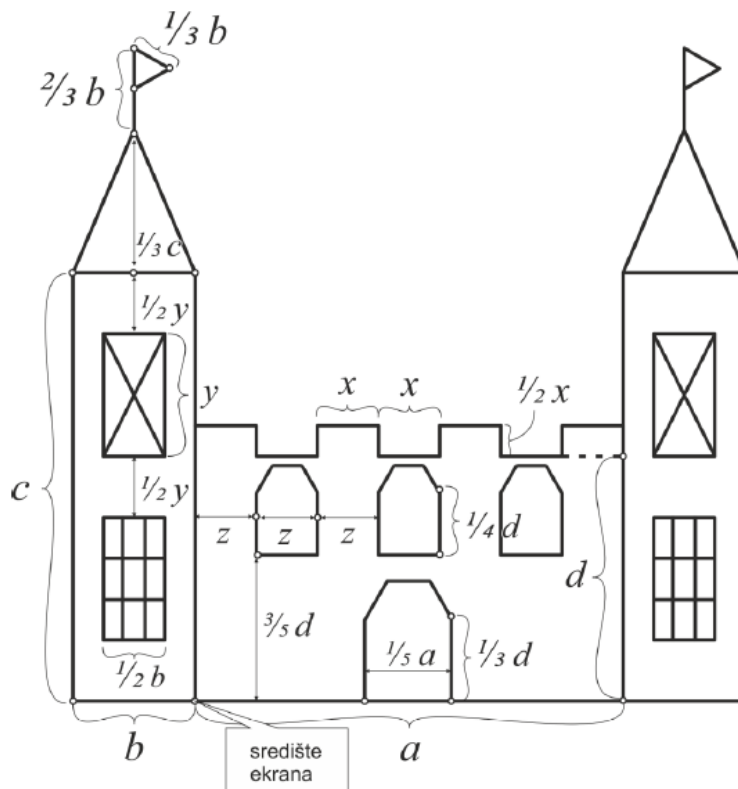
Zadatak ukupno nosi 50 bodova, a za dobiti 20 bodova dovoljno je napisati rješenje kada lista sadrži najviše tri riječi.

Zadatak koji nas se posebno dojmio je zadatak **DVORAC** sa županijske razine 2015. godine koji se pojavljuje u natjecateljskim zadacima županijske razine za sve razrede osnovne škole te godine. Naime, zadatak **DVORAC** u petom i šestom razredu se jedino razlikuje od zadatka **DVORAC** u sedmom i osmom razredu u tome što je unaprijed određen mnogokut (šesterokut) čija se polovica nalazi iznad vrata i prozora u središnjem dijelu dvorca.

Zadatak **DVORAC**

Napišite proceduru DVORAC :a :b :c :d :n :m :k :l koja briše ekran i crta dvorac. Dvorac se sastoji od središnjeg dijela i dva tornja. Sve dimenzije objašnjene su na skici.

Na sredini središnjeg dijela crtaju se vrata, a iznad njih :n pravilno raspoređenih prozora. Horizontalni razmak između prozora i razmak između prozora i lijevog i desnog ruba središnjeg dijela jednak je širini prozora. Gornji dio vrata i prozora jest polovina pravilnog mnogokuta sa 2:l stranica (čije duljine odredite sami) okrenutog prema gore. Iznad prozora nalazi se :m pravokutnih izbočenja. Izbočenja su jednako široka kao i razmak između njih. Visina izbočenja jednaka je polovini širine izbočenja.*



Unutar tornjeva nalazi se k pravilno raspoređenih prozora. Vertikalni razmak između prozora i razmak od prozora do gornjeg i donjeg ruba tornja jednak je polovini visine prozora. Širina prozora jednaka je polovini širine tornja. Na prvom prozoru odozdo crta se k pravilno raspoređenih horizontalnih i vertikalnih rešetki. Na sljedećem se crtaju dijagonale. Na sljedećem rešetke, pa dijagonale...

Krov tornja je jednakokrtačan trokut, a na vrhu krova je zastavica sastavljena od štapa i jednakostraničnog trokuta.

a , b , c i d su brojevi veći od nule. n , m i k su prirodni brojevi, l je prirodan broj veći od jedan (i manji od 36). Parametri će biti takvi da lik ne prelazi rubove ekrana. Pozicija lika na ekranu je **bitna** (vidi skicu).

Zadatak **DVORAC** istiskuje dosta vremena te preciznost pri crtanju i praćenju pozicija. Učenicima može olakšati ako napišu proceduru za toranj i prozor te ih pozivaju po potrebi.

Službeno rješenje zadatka:

```
to dvorac :a :b :c :d :n :m :k :l
```

```
  home
```

```
  rt 90/:l repeat :l [fd 10 rt 180/:l] make "v xcor
```



```

cs
rt 90 bk :b toranj :b :c :k pu home pd
rt 90
fd 2*:a/5 lt 90 fd :d/3
rt 90/:l repeat :l [fd :a/5*10/:v rt 180/:l] lt 90/:l
fd :d/3 lt 90 bk :a/5 fd 3*:a/5
lt 90 fd :d
make "x :a/(2*:m-1)
fd :x/2 lt 90 fd :x lt 90 fd :x/2
repeat :m-1 [rt 90 fd :x rt 90 fd :x/2 lt 90 fd :x lt 90
fd :x/2] home
make "z :a/(2*:n+1)
pu fd 3*:d/5 rt 90
repeat :n[fd :z lt 90 pd fd :d/4
rt 90/:l repeat :l [fd :z*10/:v rt 180/:l] lt 90/:l
fd :d/4 lt 90 bk :z fd :z pu]
pu home rt 90 pd fd :a+:b bk :b toranj :b :c :k
end
to prozor :y :b :k :s
pu fd :y/2 pd repeat 2 [fd :y rt 90 fd :b rt 90] fd :y
ifelse :s=0 [
repeat :k-1 [bk :y/:k rt 90 fd :b bk :b lt 90] fd :y*( :k-
1)/:k
rt 90
repeat :k-1 [fd :b/:k rt 90 fd :y bk :y lt 90] bk :b*( :k-
1)/:k
lt 90][
setxy xcor+:b ycor-:y

```

```

setxy xcor ycor+:y
setxy xcor-:b ycor-:y
setxy xcor ycor+:y ]
end
to toranj :b :c :k
  lt 90 fd :c setxy xcor+:b/2 ycor+:c/3
  fd 2*:b/3 rt 120 fd :b/3 rt 120 fd :b/3 rt 120 bk :b/3
  setxy xcor+:b/2 ycor-:c/3 rt 90 bk :b fd :b rt 90 fd :c
  rt 90 fd 3*:b/4 rt 90
  make "s 1
  repeat :k [make "s 1-:s prozor 2*:c/(3*:k+1) :b/2 :k :s]
end

```

Zadatak nosi ukupno 70 bodova, odnosno provjerava se 7 test primjera te svaki nosi 10 bodova. Nacrtno samo vanjski rub dvorca nosi 14 bodova, ali ako je uz vanjski rub dvorca nacrtno i središnji dio ili jedan stup rješenje donosi 35 bodova.

Ako učenik za neki test primjer treba dobiti navedene bodove, a nije obrisao ekran, pojavila se neka greška, ili ima neka linija viška, ili neka nedostaje, ili nije dobra dimenzija vrata, ili nije dobra dimenzija prozora u tornju, ili nije dobra dimenzija prozora u tijelu, ili polovina pravilnog mnogokuta nije dobra, ili rešetke nisu dobre, ili dijagonale na prozorima nisu dobre ili pravokutna izbočenja nisu dobra, ili lukovi na prozorima nisu dobre duljine, ili luk na vratima nije dobre duljine, ili je ispisana poruka o nekakvoj grešci, ili je program ostao u beskonačnoj petlji, oduzima se po jedan bod za svaki od navedenih nedostataka.

Zaključak

Proučavajući natjecateljske zadatke u kategoriji **Primjena algoritama** za osnovnu školu te njihova rješenja uočili smo da su potrebna znanja za rješavanje proučavanih zadataka većim dijelom usklađena sa službenim dokumentima. Također, htjeli bismo istaknuti da su tekstovi zadataka zanimljivi i poučni.

Najviše primjedbi imamo na zadatke u kategoriji **LOGO** u nižim razredima osnovne škole, gdje se od učenika očekuju znanja mnogih pojmova s kojima se nisu do tada susreli kroz školovanje.

Literatura

- [1] M. Babić, N. Bubica, Z. Dimovski, S. Leko, N. Mihočka, I. Ružić, M. Stančić, B. Vejnović, #mojportal5, *Udžbenik informatike u petom razredu osnovne škole*, Školska knjiga, 2018.
- [2] V. Galešev, H. Kovač, I. Kniewald, G. Sokol, V. Vlahović, *Informatika +5, Udžbenik iz informatike za 5. razred osnovne škole*, SysPrint, 2.izdanje, 2018.
- [3] M. Babić, N. Bubica, Z. Dimovski, S. Leko, N. Mihočka, I. Ružić, M. Stančić, B. Vejnović, #mojportal6, *Udžbenik informatike u šestom razredu osnovne škole*, Školska knjiga, 2018.
- [4] V. Galešev, H. Kovač, I. Kniewald, G. Sokol, V. Vlahović, *Informatika +6, Udžbenik iz informatike za 6. razred osnovne škole*, SysPrint, 2.izdanje, 2018.
- [5] M. Babić, N. Bubica, Z. Dimovski, S. Leko, N. Mihočka, I. Ružić, M. Stančić, B. Vejnović, #mojportal7, *Udžbenik informatike u sedmom razredu osnovne škole*, Školska knjiga, 2018.
- [6] V. Galešev, D. Kager, H. Kovač, I. Kniewald, N. Kunštek, G. Sokol, V. Vlahović, *Informatika +7, Udžbenik iz informatike za 7. razred osnovne škole*, SysPrint, 2.izdanje, 2018.
- [7] M. Babić, N. Bubica, Z. Dimovski, S. Leko, N. Mihočka, I. Ružić, M. Stančić, B. Vejnović, #mojportal8, *Udžbenik informatike u osmom razredu osnovne škole*, Školska knjiga, 2018.
- [8] V. Galešev, D. Kager, H. Kovač, I. Kniewald, N. Kunštek, G. Sokol, J. Purgar, *Informatika +8, Udžbenik iz informatike za 8. razred osnovne škole*, SysPrint, 2.izdanje, 2018.
- [9] *Natjecateljski zadaci i rješenja*, dostupno na <https://informatika.azoo.hr/> (travanj 2019.)
- [10] *Službene propozicije*, dostupno na <https://informatika.azoo.hr/Content/Downloads/Propozicije-Informatika-2019.pdf> (travanj 2019.)
- [11] *Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*, dostupno na https://narodne-novine.nn.hr/clanci/sluzbeni/2018_03_22_436.html (travanj 2019.)
- [12] *Matematika za 5. razred: pregled lekcija*, dostupno na <http://www.eduvizija.hr/> (travanj 2019.)
- [13] *Matematika za 6. razred: pregled lekcija*, dostupno na <http://www.eduvizija.hr/> (travanj 2019.)

- [14] *Matematika za 7. razred: pregled lekcija*, dostupno na <http://www.eduvizija.hr/>
(svibanj 2019.)
- [15] *Matematika za 8. razred: pregled lekcija*, dostupno na <http://www.eduvizija.hr/>
(svibanj 2019.)

Sažetak

U ovome radu analizirali smo natjecateljske zadatke za osnovnu školu iz predmeta Informatika posljednjih pet godina te njihova rješenja. Također, uspoređivali smo potrebna znanja iz službenih propozicija s gradivom odobrenih udžbenika (*SysPrint* i *Školska knjiga*) prema Katalogu obveznih udžbenika za školsku godinu 2018./2019. te s ishodima domene **Računalno razmišljanje i programiranje** (*Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*).

Kroz prvo poglavlje detaljno smo opisali natjecanje iz predmeta Informatika u Republici Hrvatskoj. Objasnili smo kako je organizirano te koje su kategorije i razine natjecanja. Posebnu pažnju posvetili smo kategoriji **Algoritmi za osnovnu školu: Primjena algoritama** i **LOGO**, čije zadatke smo detaljno analizirali u sljedećim poglavljima.

U drugom poglavlju proučavali smo zadatke s natjecanja za peti razred osnovne škole u kategorijama **Primjena algoritama** i **LOGO**. Osvrnuli smo se na potrebna znanja učenicima za natjecateljske zadatke te ih usporedili s gradivom promatranih udžbenika te *Kurikulumom nastavnog predmeta Informatika za osnovne škole i gimnazije u Republici Hrvatskoj*. Za svaki zadatak, istaknuli smo zašto smo odabrali pojedini zadatak te koja znanja su potrebna učeniku za riješiti zadatak.

Treće poglavlje je analiza natjecateljskih zadataka za šesti razred osnovne škole u kategorijama **Primjena algoritama** i **LOGO**. Na početku poglavlja, kratko smo usporedili potrebna znanja učenika iz službenih propozicija s temama obrađenim u udžbenicima te ishodima domene **Računalno razmišljanje i programiranje**. Također, kao i u prethodnom poglavlju, komentirali smo izdvojene natjecateljske zadatke te rješenja tih zadataka.

U četvrtom poglavlju proučavali smo zadatke s natjecanja za učenike sedmog razreda u kategoriji **Algoritmi za osnovnu školu**. Na početku poglavlja usporedili smo potrebna znanja za natjecateljske zadatke s gradivom redovne nastave predmeta Informatika. Zatim smo odabrali ogledne zadatke s natjecanja te ih detaljno opisali.

Peto poglavlje je analiza zadataka s natjecanja za učenike osmog razreda u kategoriji **Algoritmi za osnovnu školu**. Kao i u prethodnim poglavljima, prvo smo usporedili gradivo natjecateljskih zadataka s gradivom redovne nastave, a zatim smo proveli detaljnu analizu zadataka koji se pojavljuju na natjecanju učenika osmog razreda.

Summary

In this thesis we analyzed the competitive tasks for middle school in Informatics for the past five years and their solutions. We also compare the necessary knowledge from the official propositions with the material of the approved textbooks (*SysPrint* and *Školska knjiga*) according to the Catalogue of compulsory textbooks for school year 2018/2019 and with the outcomes of the domain **Computational Thinking and Programming** (*Curriculum of Middle and Secondary School Informatics in the Republic of Croatia*).

Through the first chapter we described in detail the competition in Informatics in the Republic of Croatia. We explained how organized and what are the categories and levels of competition. Particular attention was given to the category **Algorithms for Middle School: Application Algorithms** and **LOGO**, whose tasks have been analyzed in detail in the following chapters.

In the second chapter we studied the tasks from the competition for the fifth grade of middle school in the **Applications of Algorithms** and **LOGO** categories. We looked at the students competences needed for competitive tasks and compared them with the contents of the studied textbooks and the *Curriculum of Elementary and Secondary School Informatics in the Republic of Croatia*. For each task, we pointed out why we chose an individual task and what knowledge is needed for the student to solve the task.

The third chapter was an analysis of competing tasks for the sixth grade of middle school in the **Applications of Algorithms** and **LOGO** categories. At the beginning of the chapter, we briefly compared student's knowledge of the official propositions with the topics discussed in the textbooks and the outcomes of the **Computational Thinking and Programming** domain. Also, as in the previous chapter, we commented on featured competitive tasks and solutions to these tasks.

In the fourth chapter, we studied competing tasks for seventh grade students in the **Algorithms for Middle School** category. At the beginning of the chapter, we compared the necessary knowledge for compulsory competitions with the elementary course of Informatics. Then we chose the sample assignments from the competition and described them in detail.

The fifth chapter was an analysis of competitions for students of the eighth grade in the **Algorithms for Middle School** category. As in the previous chapters, we first compared the compilation of competitive assignments with regular teaching, and then we conducted a detailed analysis of the tasks appearing in the eighth-grade student competition.

Životopis

Rođena sam 12. travnja 1994. godine u Čapljini u Bosni i Hercegovini. Školovanje sam započela u Osnovnoj školi Lipanjske zore u Višićima nakon čega sam 2009. godine upisala gimnaziju u Metkoviću, prirodoslovno-matematički smjer. Po završetku gimnazije, 2013. godine, upisujem preddiplomski sveučilišni studij Matematika na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu. 2015. godine se prebacujem na preddiplomski sveučilišni studij Matematika; smjer: nastavnički. Dvije godine poslije, 2017. godine, stječem akademski naziv sveučilišne prvostupnice edukacije matematike. Iste godine, upisujem diplomski sveučilišni studij Matematika i informatika; smjer: nastavnički. Metodičku praksu iz matematike i informatike odradila sam u Osnovnoj školi Vrbani te u XV. gimnaziji u Zagrebu.