

| | |
|--------------------|-----------------------------|
| Zadatak UČO | Autor: Mihael Liskij |
|--------------------|-----------------------------|

Rješenje zadatka, s obzirom da nema nepoznatih duljina, je praćenje skice i crtanja onoga što vidimo. Za smanjenje količine posla se mogla iskoristiti naredba `REPEAT` za crtanje pravokutnika.

potrebno znanje: osnovne naredbe za pomicanje kornjače.

| | |
|-----------------------|--------------------------------|
| Zadatak LJUTKO | Autor: Frano Mihaljević |
|-----------------------|--------------------------------|

Za rješavanje zadatka bilo je potrebno nacrtati veliku količinu pravokutnika i jedan jednakostranični trokut. Mogli smo primjerice prvo nacrtati cipele (pomoću naredbe `REPEAT`, iako njeno korištenje nije nužno, ona olakšava zadatak), pomaknuti kornjaču prema nogama, nacrtati noge, pomaknuti kornjaču do tijela, nacrtati tijelo i ruke, pomaknuti kornjaču do glave, nacrtati glavu i potom kapicu okretanjem za 30° i crtanjem jednakostraničnog trokuta.

Budući da se crtež sastoji od 9 pravokutnika, za olakšavanje zadatka mogla se napisati procedura koja crta pravokutnik, a kao ulaz prima duljinu stranica. Takvu proceduru sadrži i službeno rješenje.

potrebno znanje: osnovne naredbe za pomicanje kornjače.

| | |
|-----------------------|--------------------------------|
| Zadatak SREČKO | Autor: Frano Mihaljević |
|-----------------------|--------------------------------|

U zadatku Srećko za polovicu bodova, bilo je dovoljno nacrtati kvadrat, pomaknuti kornjaču uz dizanje pera do usta, spustiti pero i nacrtati usta, pomaknuti kornjaču s podignutim perom do očiju, spustiti pero i nacrtati oči, pri čemu se između očiju zaustaviti da nacrtamo obrub novčića.

Za nacrtati krunu bilo je potrebno odrediti kut za koji se treba okrenuti da bismo tu krunu počeli crtati te kuteve za koje se trebamo okrenuti kad crtamo krunu. Da bismo to otkrili, bilo je potrebno primijetiti da dvije postranične iscrtkane crte u skici zadatka dopunjuju gornji jednakostranični trokut do jednakostraničnog trokuta s duljinom stranice jednako duljini stranice kvadrata (100 piksela). Iz ovoga izračunamo da su sve iscrtkane crte duljine 50 piksela i da čine zajedno sa stranicama krune jednakostranične trokute, a unutarnji kutevi jednakostraničnih trokuta iznose 60° . Ostatak crtanja krune sada je trivijalan.

potrebno znanje: osnovne naredbe za pomicanje kornjače, crtanje kružnice.

| | |
|-------------------------|-----------------------------|
| Zadatak POSPANKO | Autor: Mihael Liskij |
|-------------------------|-----------------------------|

Ovo je klasični zadatak sa ponavljanjem uzorka u jednoj liniji. Ako uspijemo napisati kod koji crta jedno od slova "Z", tada možemo upotrebom naredbe `REPEAT` to ponoviti `:n` puta.

Glavni problem kod crtanja slova "Z" je što kut nije fiksni, već je zadan preko varijable. Jedan mogući način kako ovaj problem riješiti je da izračunamo za koliko se kornjača mora rotirati, a drugi je da probamo intuitivno rotirati kornjaču za one kutove koji su nam poznati.

Krenimo s crtanjem slova odozgo prema dolje. Prvi korak je povući gornju vodoravnu liniju, a nakon što smo to napravili je pitanje za koliki kut se moramo rotirati. Ako se okrenemo za 180 stupnjeva u desno, onda rotacijom za `:kut` stupnjeva u lijevo dobivamo točnu rotaciju za nacrtati dijagonalnu liniju. Primijetimo da smo ovime zapravo izračunali vanjski kut koji iznosi $180 - :kut$. Isti način razmišljanja možemo primijeniti prilikom sljedeće rotacije, prvo se okrenemo za 180 stupnjeva u lijevo, a onda za `:kut` stupnjeva u desno, time smo dobili rotaciju za crtanje donje vodoravne linije. Nakon toga je jedino preostalo da se pomaknemo na sličan način u poziciju za crtanje sljedećeg slova.

Stavljanjem prethodnog koda u `REPEAT` naredbu koja se ponavlja `:n` puta smo dobili rješenje zadatka.

potrebno znanje: osnovne naredbe za pomicanje kornjače, osnove kutova, petlje

Odredimo najprije hijerarhiju elemenata u zadatku. Konačna se slika sastoji od **redova piramida** kvadrata naslaganih jedan na drugi, redovi tih piramida sastoje se od samih **piramida** koje se pak sastoje od **redova kvadrata** naslaganih jedan na drugi koji se, u konačnici sastoje od samih **kvadrata**. Kada smo uočili strukturu slike, možemo krenuti na razradu skice rješenja (papir i olovka!). Razdvojiti ćemo implementaciju u sljedeće procedure (od najnižeg do najvišeg stupnja apstrakcije):

```
to kvadrat :d
  ; crta kvadrat duljine stranice :d.
  ; kornjača se na kraju nalazi u donjem lijevom kutu i gleda gore.
end
```

```
to red_kvadrata :n :d
  ; crta red od :n kvadrata duljine stranice :d
  ; kornjača se na kraju nalazi u donjem lijevom kutu i gleda gore.
end
```

```
to piramida :n :d
  ; crta piramidu od :n redova kvadrata duljine stranice :d
  ; kornjača se na kraju nalazi u donjem lijevom kutu i gleda gore.
end
```

```
to red_piramida :m :n :d
  ; crta red od :n piramida koje se sastoje od :n redova kvadrata.
  ; kornjača se na kraju nalazi u donjem lijevom kutu i gleda gore.
end
```

```
to stidljivko :m :n :d
  ; crta konačnu sliku.
end
```

Svaka od ovih funkcija sama po sebi je vrlo jednostavna i savjetujemo da ih sami pokušate tako implementirati. Ako slučajno zapnete, pogledajte službenu implementaciju koja slijedi istu strukturu.

potrebno znanje: rastavljanje problema na potprobleme, petlje.

Ideja zadatka je zamisliti fidget spinner s n krakova kao mnogokut koji ima $2 \cdot n$ stranica, ali se one ne crtaju kao stranice, već kao poluelipse. Stranice zamišljenog mnogokuta, odnosno poluelipse, okrenute su naizmjenice prema unutra i prema van te se njihove veličine mijenjaju sukladno skici u zadatku. Crtanjem poluelipsi nastaje obris spinnera.

Za osvajanje 30% bodova u zadatku nije trebalo poznavati naredbu `ellipsearc` kojom se crtaju poluelipse jer je bilo garantirano da će se raditi o polukružnicama koje se može nacrtati, osim pomoću `ellipsearc`, i naredbom `arc`.

Nakon crtanja obrisa spinnera bilo je potrebno ucrtati kružnice u krakove, te u središte spinnera. Kružnice u krakovima imaju središte na polovici zamišljene stranice mnogokuta pa ih se može crtati u repeat petlji tijekom crtanja obrisa spinnera. Središnji par koncentričnih kružnica u spinneru nalazi se na središnjoj poziciji između svih kružnica na krakovima. Zato se položaj tog para kružnica po osi x može dobiti kao aritmetička sredina x koordinata kružnica u krakovima spinnera. Njihov položaj po osi y dobije se također kao aritmetička sredina y koordinata kružnica u krakovima spinnera. Aritmetička sredina je prosjek koordinata. Npr. za izračunavanje aritmetičke sredine x koordinata kružnica iz zadatka, potrebno je tijekom crtanja zbrojiti njihove x koordinate i na kraju taj zbroj podijeliti s brojem pribrojnika, odnosno kružnica kojima smo zbrajali koordinate.

Već samim crtanjem obrisa spinnera moglo se dobiti 50% bodova u zadatku jer su u 50% slučajeva polumjeri kružnica r_1 i r_2 bili jednaki 0.

potrebno znanje: ponavljanje uzoraka, crtanje kružnice, crtanje dijelova elipse, osnove koordinatne grafike.

| | |
|-----------------------|------------------------------|
| Zadatak GLUPKO | Autor: Antea Hadviger |
|-----------------------|------------------------------|

Zadatak rješavamo simulacijom zadanih poteza, tj. premještanja diskova s jednog štapa na drugi. Nužno je unutar neke strukture pamtiti trenutno stanje igre, na kojem se štapu nalaze koji diskovi i u kojem poretku. Stanje ćemo pamtiti u listi :s koja se sastoji od :n podlisti. Svaka podlista opisuje stanje odgovarajućeg štapa, odnosno diskove koji se na njemu nalaze, odozdo prema gore. Na početku prva podlista liste :s sadrži brojeve od :m do 1 (u padajućem poretku), a ostale su prazne. Kako bi manipulacije sadržajem listi bile što jednostavnije, potrebno je napisati proceduru postavi :l :x :v koja na :x-to mjesto liste :l umjesto trenutne vrijednosti postavlja vrijednost :v.

Iteracijom po listi poteza provodimo korake algoritma. Ako je potez oblika [:a :b :c], u privremene varijable upišemo :a-tu i :b-tu podlistu liste :s. Iz :a-te podliste uzimamo :c zadnjih članova i stavljamo ih na kraj :b-te podliste. To možemo provesti tako da u petlji koja se ponavlja :c puta uzimamo zadnji član :a-te podliste i stavljamo ga na početak privremene liste :t. Nakon toga sve članove privremene liste :t stavljamo na kraj :b-te podliste. Na kraju modificiramo listu :s pomoću prethodno napisane procedure postavi :l :x :v tako da na :a-to i :b-to mjesto stavimo novodobivene podliste. Nakon provođenja svih poteza, još jednim prolaskom kroz listu :s nacrtamo stanje na kraju igre.

potrebno znanje: liste, petlje.

| | |
|----------------------------|----------------------------|
| Zadatak SNJEGULJICA | Autor: Marija Gegić |
|----------------------------|----------------------------|

Za osvajanje 10% bodova na ovom zadatku, dovoljno je bilo nacrtati jednu kružnicu radijusa r . Za osvajanje dodatnih 40% bodova koje pokrivaju test primjeri iz drugog dijela sekcije bodovanja, potrebno je bilo uočiti da će se na prvoj razini nalaziti točno jedna kružnica, a na svakoj sljedećoj razini će se nalaziti duplo više kružnica (soba) nego na prethodnoj. Uz to opažanje, jednostavno je odrediti broj razina jer znamo ukupan broj kružnica koje je potrebno nacrtati.

Za osvajanje svih bodova, potrebno je bilo uočiti neke pravilnosti u listama koje opisuju Snjeguljičin i Prinčev obilazak soba. U listi :a koja predstavlja Snjeguljičin obilazak, za svaku će sobu X vrijediti da se sve sobe u koje možemo doći kad krenemo lijevom hodnikom iz sobe X nalaze u listi :a prije svih soba u koje možemo doći ako krenemo desnim hodnikom iz sobe X . U listi :b koja predstavlja Prinčev obilazak, za svaku sobu će vrijediti da se sve sobe do kojih možemo doći ako krenemo lijevom hodnikom iz sobe X nalaze u listi :b prije sobe X i prije svih soba do kojih možemo doći ako krenemo desnim hodnikom iz sobe X .

Oznaka prve sobe u listi :a ujedno je i oznaka jedine sobe koja nema ulazne hodnike. Ako pronađemo poziciju oznake te sobe u listi :b (označimo tu poziciju s i), znamo da postoji točno $(i-1)$ soba u koje možemo doći ako krenemo lijevom hodnikom iz te prve sobe. Do svih soba koje se nalaze nakon i -te pozicije možemo doći krenemo li desnim hodnikom iz početne sobe. Sada kada znamo koje se sobe nalaze lijevo a koje desno, na skupovima tih soba rekursivno provedemo isti postupak.

Preostaje još odabrati dobar način za pamćenje tih odnosa (koje se sobe nalaze lijevo, a koje desno od određene sobe) kako bismo najlakše mogli nacrtati traženo rješenje. Jedan od mogućih načina je da svaku sobu predstavimo listom s tri elementa. Prvi element je oznaka te liste, drugi element je broj -1 ili podlista koja sadrži sobe koje se nalaze lijevo od te sobe, a treći element je broj -1 ili podlista koja sadrži sobe koje se nalaze desno od te sobe. Ako su drugi ili treći element -1 , to nam označava da ne postoji niti jedna soba desno ili lijevo od te sobe. Ako postoje sobe lijevo ili desno, onda podliste kojima njih opisujemo također mogu sadržavati podliste koje opisuju sobe koje se njima nalaze lijevo ili desno. Rješenje drugog test primjera zadatka bismo onda opisali listom $[4 [5 [14 -1 -1] [15 -1 -1]] [3 [6 -1 -1] [13 [8 -1 -1] [22 -1 -1]]]$.

Dakle, kada našim rekursivnim načinom gradimo rješenje, ako soba s oznakom X nema hodnik koji vodi lijevo ili desno iz nje, onda vratimo listu $[X -1 -1]$. Inače,

vratimo listu koju dobijemo spajanjem broja X , te listi koje dobijemo kao rješenja rekurzivnih poziva za sobe lijevo te sobe desno od sobe X .

potrebno znanje: rekurzije.