

Zadatak RAZOČARAN**Autor: Antea Hadviger**

Zadatak se rješava osnovim naredbama za kretanje kornjače. Naredbu FD (forward) koristimo za pomicanje unaprijed za zadani broj piksela koji je potrebno očitati sa skice. Naredbe RT (right) i LT (left) koristimo za okretanje udesno, odnosno ulijevo. Kornjaču je potrebno okretati samo za iznose kutova koji su višekratnici broja 90 jer su svi kutovi pravi. Primijetimo također da je lijevi dio slike simetričan desnom, što potencijalno može olakšati rješavanje zadatka.

potrebno znanje: osnovne naredbe za pomicanje kornjače

Zadatak LOGOTIPIČAN**Autor: Frano Mihaljević**

Za početak je trebalo nacrtati kružnicu radijusa 150 piksela. Nakon toga slijedi najveći problem zadatka - odrediti kut za koji se potrebno okrenuti kako bi nacrtali ravne linije u logotipu. Na više načina moguće je odrediti da se možemo okrenuti udesno za 30 stupnjeva, potom nacrtati liniju od 25 piksela, okrenuti se ulijevo za 120 stupnjeva i nacrtati i drugu liniju od 200 piksela. Postupak je potrebno ponoviti i s donje strane kružnice.

potrebno znanje: osnovne naredbe za pomicanje kornjače, crtanje kružnice

Zadatak MAGIČAN**Autor: Mihael Liskij**

U zadatku je bilo potrebno nacrtati dvije kružnice te krila vile koja su dva spojena jednakostranična trokuta. Glavni izazov je bio određivanje unutarnjih i vanjskih kutova (60 i 120 stupnjeva) te polovičnog kuta (30 stupnjeva) kako bi se nacrtala krila vile. Preporuča se pri implementiranju koda pisati ponovno upotrebljive funkcije za crtanje krila.

potrebno znanje: osnove kutova i kretanja kornjače, crtanje kružnice

Zadatak RASPLESAN**Autor: Mihael Liskij**

Kako bi nacrtali kretanje kornjače potrebno je pratiti na kojem smo koraku. To se lako ostvaruje upotrebom varijable brojača (npr. :n ili REPCOUNT) koju povećavamo svakim korakom i provjerom na kojem koraku smo trenutno. Provjera se najlakše izvodi upotrebom naredbe REMAINDER s kojom možemo direktno

saznati koji korak moramo izvesti (naprijed-desno ili nazad-lijevo) ovisno o ostatku pri dijeljenju s 3.

Moguće je bilo i rješenje koje ne koristi REMAINDER. Za takvo rješenje se može koristiti dodatni brojač koji pamti na kojem smo koraku te kada nam brojač koraka dođe na 3 ga smanjimo ponovno na 0.

Bitno je primijetiti da je postojala mogućnost ručno pokriti slučajeve do maksimalno 6 koraka te time ostvariti 70% bodova za zadatak.

potrebno znanje: osnove kutova i kretanja kornjače, ponavljanje uzorka

Zadatak DOSADAN	Autor: Frano Mihaljević
------------------------	--------------------------------

U zadatku je prvo potrebno nacrtati kvadrat. Možemo primijetiti da u svakom sljedećem potezu crtamo po jedan križ koji dijeli kvadrat na četiri dijela. Svaki put taj križ crtamo iz donjeg lijevog kuta kvadrata, no u svakom su koraku dimenzije križa dvostruko manje nego u prethodnom. Zato je potrebno u petlji koju koristimo nakon crtanja podijeliti dimenzije križa s dva.

Alternativno, bilo je moguće osvojiti parcijalne bodove pomoću naredbe grananja (IF) i računanja koliko se puta moraju smanjiti visina i širina križa u svakom koraku.

potrebno znanje: osnove kretanja kornjače, ponavljanje uzorka

Zadatak PRAVOKUTAN	Autor: Mihael Liskij
---------------------------	-----------------------------

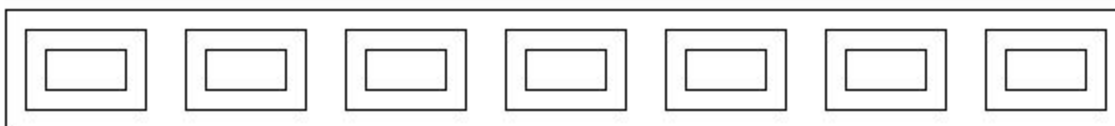
Samo crtanje ladica i stupaca ladica nije glavni problem u zadatku i mogao se riješiti for petljom i sa malo računanja duljina stranica stupaca. Kako bi se ostvarilo 100% bodova na zadatku bilo je potrebno izračunavanje duljine obruba ili pametno korištenje funkcija koordinatnog sustava. Prije konačnog rješenja proći ćemo kroz sve podzadatke.

Za prvih 20% bodova je vrijedilo da će lista imati samo jedan član i da će razmaci i obrubi ladica biti jednaki 0. Ovime se zadatak svodi na crtanje stupca od :n pravokutnika dimenzija :a * :b. Za to možemo koristiti REPEAT u kojem crtamo pravokutnike.

Za sljedećih 30% bodova još uvijek vrijedi da su razmaci i obrubi jednaki 0. Za rješenje ovog zadatka je potrebno znanje o listama. Treba iterirati po elementima

liste i za svaku vrijednost pozvati rješenje prethodnog potproblema za crtanje stupca pravokutnika čime smo ostvarili 50% bodova na zadatku.

Za dodatnih 20% bodova je vrijedilo da će svi elementi liste biti jednaki 1. Više ne vrijedi da su razmaci i obrubi nužno jednaki 0, ali skicirajući primjer možemo vidjeti da problem nije puno teži od dosadašnjih potproblema te se svodi na crtanje reda pravokutnika sa pravilnim razmacima i dvostrukim obrubom.



Za konačnih 30% bodova i ukupno 100% bilo je potrebno nacrtati obrubljene stupce te dodano i njih obrubiti. Za jednostavniju implementaciju preporučeno je napisati potprogram za crtanje jednog obrubljenog stupca ladica te njega pozivati na točnim pozicijama gdje se stupci trebaju nalaziti.

Glavni problem ostaje crtanje vanjskog obruba svih stupaca. Jedno rješenje je računanje svih duljina te upotreba osnovnih naredba za kretanje kornjače. Time je moguće riješiti problem, ali nije nužno jednostavno za izvesti.

Drugo rješenje koje je i službeno i koristi svojstvo da je pozicija lika na ekranu bitna je pamćenje koordinata kutova, točaka gdje obrub mijenja smjer. Prva točka koju ubacujemo u neku pomoćnu listu je donji lijevi kut obruba. Ona će nam biti početna točka. Prilikom crtanja stupaca u tu istu listu ubacujemo koordinate točke koja je za :d gore i za :d desno od gornjeg desnog vrha svakog stupca. Na kraju još ubacimo donji desni kut vanjskog obruba. Ovime smo si pripremili dovoljno informacija da nacrtamo obrub. Slijedi algoritam spajanja točka koji stvara obrub:

1. Pomakni se u prvu točku u listi (donji lijevi kut obruba)
SETPOS (FIRST :pom)
2. Ponavljaj za sve točke:
 - a. Postavi y-koordinatu kornjače na y-koordinatu trenutne točke, npr.
SETXY XCOR (LAST (ITEM :i :pom))
 - b. Postavi poziciju kornjače na trenutnu točku
SETPOS (ITEM :i :pom)
3. Pomakni kornjaču ponovno u prvu točku liste
SETPOS (FIRST :pom)

Ovim algoritmom za spajanje smo dobili potpuni obrub bez dodatnog računanja dimenzija obruba.

U zadatku je trebalo pripaziti na uvjet da je lik centriran na ekranu. To je najlakše ostvarivo računanjem ukupne širine cijelog lika te se prije početka crtanja pomaknuti za pola te duljine u lijevo. Tražena formula je: broj_elemenata_liste * (4

* :d + :a). U slučaju da nas zanima širina lika bez obruba, od gornje formule oduzmemo $2 * :d$.

Zadatak ŠUMOVIT

Autor: Ivan Paljak

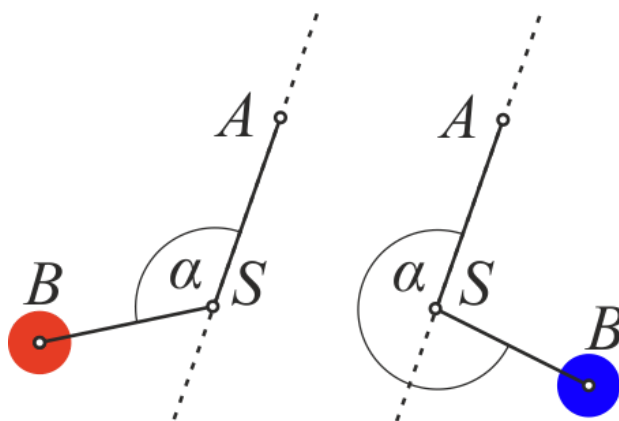
Razmotrimo najprije rješenja podzadataka iz sekcije bodovanje.

Za osvajanje 10 bodova na zadatku bilo je dovoljno znati nacrtati Jokinu poziciju. Ovaj se dio zadatka rješava osnovama koordinatne grafike (naredbe SETPOS i SETH), a implementaciju možete pogledati u proceduri `joka :p :k` u sklopu službenog rješenja.

Za dodatnih 20 bodova, znali smo da majstor Joka gleda "prema gore" pa možemo zaključiti da sva stabla hrasta lužnjaka (lijevo od joke) imaju vrijednost x-koordinate strogo manju od Jokine. Analogno, stabla hrasta kitnjaka nalaze se na pozicijama čije su x-koordinate strogo veće od Jokine.

Sličnom analizom se rješava i posljednji podzadatak gdje je Jokin smjer gledanja bio višekratnik broja 90. Primjerice, gleda li Joka "udesno" ($:k = 90$), stabla hrasta lužnjaka imat će y-koordinate strogo veće od Jokine, dok y-koordinate stabala hrasta kitnjaka biti strogo manje od Jokine. Analizu preostalih slučajeva ($:k \in \{180, 270\}$), ostavljamo čitatelju za vježbu.

Da bismo u potpunosti riješili zadatak valja utvrditi neku zakonitost kojom bismo mogli razlikovati hrastove. Označimo li poziciju majstora Joke točkom S, poziciju hrasta točkom B te istaknemo li bilo koju točku A na polupravcu određenom Jokinom pozicijom i njegovim smjerom gledanja, možemo opaziti da nam iznos kuta ASB (alfa sa skice) jednoznačno govori o kojoj se vrsti hrasta radi. Ako se radi o hrastu lužnjaku, iznos tog kuta bit će manji od 180° , dok će u protivnom biti veći.



Implementacijski gledano, nalazimo li se na poziciji majstora Joke (točka S) te označimo li sa h_1 smjer gledanja (HEADING) prema točki A, a sa h_2 smjer gledanja prema točki B. Iznos kuta alfa dobivamo naredbom `"REMAINDER (:h2-:h1+360) 360"`. Intuitivno, čini se dovoljnim gledati samo razliku vrijednosti h_2 i h_1 , no taj izraz može ispasti manji od 0 zbog toga što naredbom HEADING dobivamo iznos kuta u

odnosu na vertikalu. Da bismo dobili točan iznos potrebno je, u slučaju da iznos ispadne negativan, tom broju dodati 360. Da bismo izbjegli poseban slučaj, dodajemo 360 neovisno o iznosu promatrane razlike i gledamo ostatak pri dijeljenju sa 360.

potrebno znanje: rad s koordinatnom grafikom, matematika.

Zadatak ZANIMLJIV	Autor: Marija Gegić
--------------------------	----------------------------

Kako bismo pronašli :n-ti zanimljivi broj, možemo redom za svaki broj provjeriti je li zanimljiv, te brojati pritom koliko smo do sada našli zanimljivih brojeva. Kada nađemo :n-ti, ispišemo ga i zaustavimo izvođenje programa.

Kako bismo provjerili je li neki broj A zanimljiv, najprije moramo pronaći sve njegove djelitelje. Zatim se za svaki broj manji ili jednak A pitamo: možemo li ovaj broj prikazati kao zbroj nekih djelitelja broja A ? Da to provjerimo, možemo na sve moguće načine odabrati podskup nekih djelitelja broja A i provjeriti je li suma tog podskupa jednaka tom broju. Ukoliko možemo pronaći takav podskup za svaki broj manji ili jednak A , onda je A zanimljiv. Ukoliko nađemo bar jedan broj koji se ne može tako prikazati, možemo odmah zaključiti da A nije zanimljiv.

Ukoliko želimo ubrzati ovo rješenje, možemo primijetiti da ne moramo provjeravati za svaki broj manji ili jednak A može li se prikazati kao suma djelitelja broja A , već možemo to raditi zajedno za sve brojeve manje ili jednake A , tako da odaberemo sve moguće podskupove, pazeći pritom da dok odabiremo podskup njegova suma ne pređe A . Nakon odabira svakog podskupa, zabilježimo da je suma tog podskupa ostvariva. Za implementacijske detalje pogledajte službeno rješenje.

potrebno znanje: rekurzije