PRESEK
List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652
Letnik **16** (1988/1989)
Številka 6
Strani 343–347

Sandi Klavžar:

K NALOGAM S 3. REPUBLIŠKEGA TEKMOVANJA OSNOVNOŠOLCEV IZ RAČUNALNIŠTVA

Ključne besede: računalništvo, logo, pascal, tekmovanja.

Elektronska verzija: http://www.presek.si/16/954-Klavzar.pdf

K NALOGAM S 3. REPUBLIŠKEGA TEKMOVANJA OSNOVNOŠOLCEV IZ RAČUNALNIŠTVA

V 3. številki letošnjega Preseka smo objavili naloge, ki so jih reševali osnovnošolci na 3. republiškem tekmovanju iz računalništva. Tokrat si nekatere naloge pobliže oglejmo. Zapisali bomo rešitve za tiste, ki so reševalcem povzročale največ preglavic, ob vsaki pa skušali povedati še kaj več kot le rešitev. Za besedila nalog pobrskajte po omenjeni številki Preseka. Programi, ki jih objavljamo, so prepisani iz biltena tekmovanja, saj tako lahko bodoči tekmovalci vidijo, kaj tekmovalna komisija pričakuje od njih. Seveda to ne pomeni, da se ne bi dali programi napisati tudi kako drugače. Programi v jeziku logo so napisani v inačici loga za Commodore 64.

Naloge v jeziku logo

Izmed osmih nalog so tekmovalcem predstavljale najtrši oreh zadnje tri. Značilnost teh nalog je, da je za njihovo reševanje treba premišljevati rekurzivno. Čeprav je logo po svoji naravi rekurziven, reševanje takih problemov očitno povzroča učencem največje težave. Za zgled rešimo 6. in 7. nalogo. Slednje, mimogrede povedano, ni rešil noben tekmovalec. V 8. nalogi je potrebno narisati *spirolateral*, o risanju le teh pa je M. Lokar napisal članek v 3. številki letošnjega Preseka.

6. naloga. Kaj naredi želva v enem koraku? Potegne črto, ki je nekoliko daljša od črte, ki jo je napravila v predhodnem koraku, zatem pa se obrne v desno za določen kot. Na levi sliki pri besedilu naloge se obrne za 45° in na desni sliki za 90°. Postopek ponavljamo, dokler ne pridemo do roba zaslona.

```
TO SPIRALA1 :A
 IF :A > 84 THEN STOP
 FD :A RT 45
 SPIRALA1 :A + 2
END
```

```
TO SPIRALA2 :A
 IF :A > 98 THEN STOP
 FD :A RT 90
 SPIRALA2 :A + 2
END
```

7. naloga. Ker ne smemo uporabiti ukaza REPEAT, si moramo sami organizirati štetje objektov, ki jih rišemo. Število hiš, ki jih moramo še narisati, naj bo spravljeno v spremenljivki STEVILO, velikost hiške v A, KRAT pa je število narisanih stranic trikotnika in kvadrata.

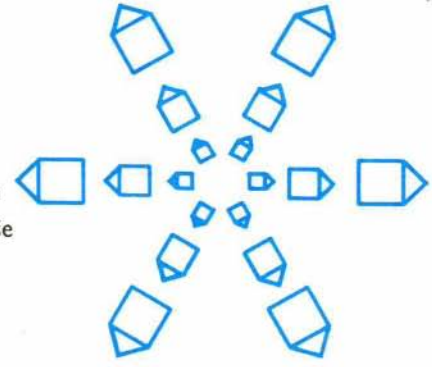
```
TO HISE :A :STEVILO
 IF :STEVILO=0 THEN STOP
 HISA :A
 PU RT 90 FD :A+10 LT 90 PD
 HISE :A+20 :STEVILO-1
END
```

```
TO HISA :A
 KVADRAT :A
 FD :A RT 30
 TRIKOTNIK :A
 LT 30 BK :A
END
```

```
TO TRIKOTNIK :A :KRAT
 IF :KRAT > 3 THEN STOP
 FD :A RT 120
 TRIKOTNIK :A :KRAT+1
END
```

```
TO KVADRAT :A :KRAT
 IF :KRAT > 4 THEN STOP
 FD :A RT 90
 KVADRAT :A :KRAT+1
END
```

Če hočemo imeti tako sliko, kot jo zahteva naloga, se moramo z želvo najprej prestaviti nekoliko v levo, pero obrniti navzgor, zbrisati zaslon in nato narediti klic HISE 10 4. Ukaza TRIKOTNIK in KVADRAT bi seveda lahko napisali brez uporabe rekurzije - le trikrat oz. štirikrat moramo ponoviti ukaza FD in RT. Nekaj podobnega bi lahko napravili tudi z ukazom HISE, a taka rešitev v tem primeru ni najboljša, saj s tem dobimo precej šibkejši ukaz. Za konec pa še zahtevnejša naloga: Popravi ukaze, tako da se bodo vedno večje hiške navijale po spirali. Dobljena slika naj bi izgledala nekako takole:

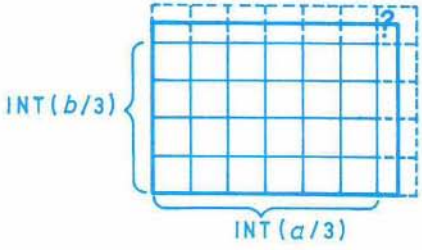


Če vam bo uspela kakšna lepa slika, nam jo pošljite skupaj s programom. Z veseljem ju bomo objavili. Seveda ni prepovedano med hiše postaviti še kako drevo, . . .

Naloge za 5.-6. razred

Druga in četrta naloga sta bili lahki, kar se je pokazalo tudi pri rezultatih. Največ preglavic je povzročala tekmovalcem prva naloga, takoj za njo pa tretja. Rešimo ju.

1. naloga. Formulacija naloge je nekoliko pomanjkljiva, saj ne vemo, v katerih enotah sta podani dolžini stranic a in b. Predpostavimo torej, da sta podani v mm. V eno vrstico najprej postavimo toliko kvadratkov, kolikorkrat gre 3 v a. Še en kvadratek gre v vrstico, če je ostanek pri deljenju a s 3 vsaj 1.5.



Izračunati moramo torej celi del števila $a/3$ in še ostanek pri tem deljenju. Za drugo stranico napravimo isti premislek, kot smo ga za prvo. Število kvadratkov je zato kar produkt števila kvadratkov v eni vrstici s številom kvadratkov v drugi. Paziti moramo le še na robne pogoje. Včasih se zgodi, da smo zgornji desni kvadratek šteli, čeprav ga ne bi smeli. To se lahko zgodi le v primeru, ko smo šli tako levo kot tudi zgoraj čez rob osnovnega kvadrata. To popravimo v vrstici 100 v spodnjem programu.

```
10    INPUT "PODAJ A IN B", A, B
20    LET POACELE = INT(A/3)
30    LET POBCELE = INT(B/3)
40    OSTPOA = A - 3*POACELE
50    OSTPOB = B - 3*POBCELE
60    IF OSTPOA >= 1.5 THEN LET POACELE = POACELE+1
70    IF OSTPOB >= 1.5 THEN LET POBCELE = POBCELE+1
80    LET SKUPAJ = POACELE * POBCELE
90    REM Še vogal
100   IF (OSTPOA*OSTPOB<4.5) AND (OSTPOA>=1.5) AND
       (OSTPOB>=1.5) THEN LET SKUPAJ = SKUPAJ-1
110   PRINT "STEVILO KVADRATKOV = ", SKUPAJ
120   END
```

3. naloga. Naloga nas približa podatkovni strukturi, ki ji učeno rečemo dvojiško drevo. Kar na primeru povejmo, kaj je v drevesu *oče* in kaj *sin*. Element $a(5)$ je oče elementoma $a(10)$ in $a(11)$, ki sta seveda njegova sinova, $a(5)$ pa je sin elementa $a(2)$. V splošnem drevesu ima lahko oče poljubno število sinov. Kadar ima vsak oče največ dva sinova, drevo imenujemo *dvojiško drevo*. Če ima vsak oče natanko dva sinova, imamo *polno dvojiško drevo* in táko je drevo iz naloge.

V polnem dvojiškem drevesu najprej opazimo, da se z vsako vrstico podvoji število elementov prejšnje vrstice. Res, število elementov po vrsticah je $1, 2, 4, 8, \ldots$ To pa so ravno potence števila 2! Prvi element v i-ti vrstici ima tako indeks 2^{i-1}, zato ima j-ti element v i-ti vrstici indeks $2^{i-1} + j - 1$. S tem je naloga že rešena. Še naloga: Kako iz indeksa očeta ugotovimo indeksa njegovih sinov. Drevo in dvojiško drevo smo opisali zelo površno. Morda ste že iz te naloge zaslutili, da gre za zanimivo podatkovno strukturo. Kdor bi rad kaj več izvedel o drevesih, lahko svojo radovednost poteši s knjigo J. Kozaka "Podatkovne stukture in algoritmi".

Naloge za 7.-8. razred

Največ točk so tekmovalci nabrali pri 2. in 5. nalogi. Predvsem 2. so rešili skoraj vsi. Najslabše so reševali 1. in 4. nalogo, slednje ni v celoti rešil nihče.

1. naloga. Naloga ni težka, zato je presenetljivo, da so pri tej nalogi tekmovalci nabrali najmanj točk. Je razlog v tem, da je bilo potrebno poznati enačbo kroga? Kakorkoli že, naloga je še en opomin tistim, ki mislijo, da lahko obvladajo računalništvo brez znanja matematike.

Če imamo zadetek s koordinatama (x, y), potem je oddaljenost te točke od središča tarče podana s formulo $\sqrt{x^2 + y^2}$. Kdor formule ne pozna, naj jo dokaže s pomočjo Pitagorovega izreka. Zadetek je v notranjem krogu, če ni dlje kot 2.5 cm od izhodišča (pazi: podatek je premer, ne polmer!), v srednjem krogu, če je na oddaljenosti med 2.5 in 7.5 cm; v zunanjem pa, če je na razdalji med 7.5 in 12.5 cm od izhodišča. Program v basicu:

```
5     LET S = 0
10    FOR I = 1 TO 10
20      PRINT AT 21,1; "PODAJ KOORDINATE ZA TOČKO "; I
30      INPUT "X = "; X
31      INPUT "Y = "; Y
35      REM Izračunamo oddaljenost zadetka od središča
         tarče
40      LET R = SQR(X^2 + Y^2)
50      IF R <= 2.5 THEN LET S = S+10
60      IF (R <=  7.5) AND (R > 2.5) THEN LET S = S+5
70      IF (R <= 12.5) AND (R > 7.5) THEN LET S = S+3
80    NEXT I
90    PRINT "SKUPNE TOČKE: "; S
100   END
```

4. naloga. Je poenostavitev naloge, ki so jo leta 1979 reševali srednješolci v prvi skupini na republiškem tekmovanju. Razlago in rešitev te naloge najdemo v knjigi "Enajsta šola računalništva", zato tu podajmo le rešitev iz biltena tekmovanja. Za spremembo je program v pascalu.

```
program Most (output);
  var st, y, y1, x: integer;

  function Semafor: integer; extern;
  function Senzor: integer; extern;

  begin {Most}
    repeat until Semafor = 0;
    repeat
      st:= 0; y1:= 0;
      repeat until Semafor = 1;
      repeat
        y:= Senzor;
        if y > y1 then st:= st+1
        y1:= y;
      until Semafor = 0;
      writeln(st);
    until false;
  end. {Most}
```

Sandi Klavžar