INFORMATICA 1/1979

TRETJE REPUBLIŠKO TEKMOVANJE SREDNJEŠOLCEV S PODROČJA RAČUNALNIŠTVA

R. REINHARDT
M. MARTINEC
R. DORN

UDK: 371.3 : 681.3

SLOVENSKO DRUŠTVO INFORMATIKA, LJUBLJANA

Povzetek. Prispevek predstavlja poročilo o tretjem republiškem tekmovanju srednješolcev iz področja računalništva, ki ga je organiziralo Slovensko društvo Informatika v aprilu 1979. V prispevku so vse naloge z rešitvami in pregled rezultatov tekmovanja.

THIRD COMPUTER SCIENCE CONTEST FOR HIGH-SCHOOL STUDENTS. The article represents a report on third Computer Science Contest. It includes the complete set of problems with their solutions and a short overwiev of contest results.

I. Uvod

Ena od rednih dejavnosti Slovenskega društva Informatika je tudi popularizacija računalništva in informatike med srednješolsko mladino. Komisija za popularizacijo računalništva je zato organizirala že tretje republiško tekmovanje srednješolcev iz področja računalništva.

Tekmovanje je bilo 21. aprila na Fakulteti za elektrotehniko v Ljubljani, udeležilo pa se ga je rekordno število tekmovalcev: 56 po prvem letu pouka in 36 po drugem letu pouka računalništva.

Pri organizaciji letošnjega tekmovanja so poleg Društva in Fakultete za elektrotehniko sodelovali še Inštitut Jožef Stefan, sodelavci koordinativne delovne skupine za izvedbo projekta Pouk računalništva v usmerjenem izobraževanju, finančno pa so tekmovanje podprli Elektrotehna - TOZD Digital, Iskra - TOZD Računalniki, Intertrade, Republiški računski center in Hotel Lev.

Tekmovanje je otvoril rektor Univerze E. Kardelja v Ljubljani prof. dr. Slavko Hodžar, tekmovalce pa so pozdravili: predsednik Slovenskega društva Informatika prof. dr. Anton P. Železnikar, dekan Fakultete za elektrotehniko prof. dr. Jernej Virant in predsednik komisije za popularizacijo računalništva Roman Dorn.

Primarni cilj tekmovanja je popularizacija računalništva, obenem pa se učenci srednjih šol seznanijo z možnostmi študija na področju računalništva. Ker večino tekmovalcev spremljajo učitelji računalništva, je tekmovanje tudi priložnost za izmenjavo izkušenj in mnenj. Zato je potekal vzporedno s tekmovanjem tudi pogovor o pouku računalništva v usmerjenem izobraževanju, računalniških poklicih, računalniški opremi za srednje šole in kadrovskih potrebah. Na tem pogovoru so se srečali predstavniki višjih in visokih šol, predstavniki izobraževalne skupnosti in uporabniki iz različnih delovnih organizacij.

Po tekmovanju so si udeleženci organizirano ogledali bližnje računalniške centre, predstavniki višjih in visokih šol iz obeh slovenskih univerz pa so jih podrobneje seznanili s študijem in učnimi načrti svojih organizacij. Sledila je razglasitev rezultatov, na kateri so prvouvrščeni tekmovalci prejeli plakete in knjižne nagrade.

II. Naloge za učence po prvem letu pouka računalništva.

Čas reševanja je 2 uri in 30 minut. Dovoljena je uporaba vse literature. Ena naloga od petih je neobvezna.

1. Napiši program, ki izpiše naslednjo tabelo števil:

```
... 0 0 0 1 0 0 0 ...
... 0 0 1 1 1 0 0 ...
... 0 1 2 3 2 1 0 ...
... 1 3 6 7 6 3 1 ...
```

V tabeli je prva vrstica sestavljena iz samih ničel, le srednje število je 1, vsako število v naslednjih vrsticah pa je enako vsoti treh nad njim ležečih števil iz prejšnje vrstice. Tabela naj se izpiše v 21 stolpcih, izpis pa se naj konča, ko so vsa izpisana števila v zadnji vrsti različna od nič.

2. Imamo tako ozek most, da se na njem dva avtomobila ne moreta srečati. Na vsakem koncu mostu je postavljen semafor in tipalo, ki pove, če pred mostom čaka kak avtomobil. Tudi sam most je opremljen z instrumentom, ki pove, če je na mostu kak avtomobil. Napiši postopek za krmiljenje semaforjev, ki bo skrbel, da nihče ne čaka po nepotrebnem in da se promet v konicah odvija izmenično (most je zelo kratek). Naprave ob mostu krmilimo z naslednjima podprogramoma in podprogramsko funkcijo:

ODPRI(str) — str je oznaka ene izmed strani mostu. ODPRI povzroči, da se na imenovani strani odpre semafor;

ZAPRI(str) — str je spet oznaka strani mostu. ZAPRI zapre semafor na imenovani strani;

TIPALO(t) — t označuje, katero tipalo bi radi vprašali, če vidi kak avto. t lahko označuje bodisi tipalo na eni izmed strani mostu, ali pa tipalo na mostu. TIPALO je funkcija, ki pove, če imenovano tipalo zaznava kakšen avtomobil. Če ga zaznava, je njena vrednost DA, sicer pa NE.

Imeni strani mostu sta A in B, ime tipala na mostu pa je M. A, B, M, DA in NE so vnaprej definirane konstante.

Primeri:

ODPRI(A)	Odpre semafor na strani A.
ZAPRI(B)	Zapre semafor na strani B.
TIPALO(M)	Ima vrednost DA, če je na mostu kak avto.
TIPALO(B)	Ima vrednost NE, če na strani B ni vozil.

3. Definirani imamo naslednji funkciji (n je naravno število):

$$s(n) = \begin{cases} n & \underline{\text{če}}\ 0<n<10; \\ s(n\ /\ 10) + n\ \underline{\text{mod}}\ 10 & \underline{\text{če}}\ n>9; \end{cases}$$

Deljenje je celoštevilčno; n mod 10 pa pomeni ostanek pri deljenju n z 10.

$$p(n) = \begin{cases} n & \underline{\text{če}}\ 0<n<9; \\ 0 & \underline{\text{če}}\ n=9; \\ p(s(n)) & \underline{\text{če}}\ n>9; \end{cases}$$

a) Izračunaj s(15324) in p(15324).
b) Kaj računa funkcija s (razloži).
Neobvezno:
c) *Dokaži, da za vsako naravno število n* velja p(n) je ostanek pri deljenju n z 9.

4. Neki programer sumljivih kvalitet nam je prinesel naslednji program:

```
C     program obrne podatke
      INTEGER T(100),Z
      INTEGER I,J,N
      READ(2,1)N
1     FORMAT(I5)
      READ(2,2)(T(I),I=1,N)
2     FORMAT(16I5)
      J=N
      DO 3 I=1,N
        Z=T(I)
        T(I)=T(J)
        T(J)=Z
        J=J-1
3     CONTINUE
      WRITE(3,4)(T(I),I=1,N)
4     FORMAT(1X,10I10)
      CALL EXIT
      END
```

```
program o(input,output);
  var n;i;j;z:integer;
     t:array[1..100]of
            integer;
  begin  {obrni podatke}
    readln(n);
    for i:=1 to n do
      read(t[i]);
    j:=n;
    for i:=1 to n do
      begin  z:=t[i];
        t[i]:=t[j];
        t[ ]:=z;
        j:=j-1
      end;
    for i:=1 to n do
      write(t[i])
  end.
```

Napiši, kaj program izpiše, če dobi naslednje podatke:

```
10
9   8   7   6   5   4   3   2   1   0
```

Ali lahko uganeš, kaj je programer hotel napisati in program popraviš?

5. Programi, ki so zapisani v nekem programskem jeziku (fortran ali pascal) se lahko pri danih podatkih ustavijo, ali pa tudi ne. Dokaži, da ne obstoja program (imenujmo ga T) zapisan v istem programskem jeziku, ki bi za vsak program in njegove podatke izračunal, ali se program ustavi ali ne.

```
                    +-----+
program      ----->|     |             se ustavi
                    |  T  |----->
podatki zanj ----->|     |             se ne ustavi
                    +-----+
```

Nasveta:

A) Predpostavi, da bi imeli tak program T. S pretvorbami, ki so izvedljive, ga spremeni v drugačen program, ki gotovo ne obstoja. Če so pretvorbe zanesljivo izvedljive, potem T ne obstoja.

B) Program, ki naj bi se po enem razmisleku ustavil in se po drugačnem razmisleku ne bi, zanesljivo ne obstoja.

III. Naloge za učence po drugem letu pouka računalništva.

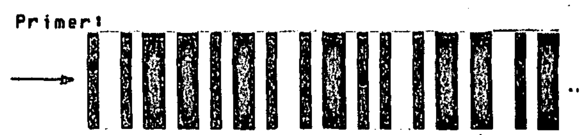
(Pogoji so isti kot za tekmovalce po prvem letu pouka.)

1. n otrok se hoče loviti. Poznajo izštevanko z m besedami. Napiši program, ki pove, kateri od otrok lovi. Otroke oštevilčimo s številkami od 1 do n in začnemo izštevati pri prvem otroku. Lovi tisti, ki zadnji ostane v krogu.

2. Neko informacijo imamo natisnjeno na papirju v posebni kodi. Na papirju so izmenično črni in beli pasovi. Tanki pasovi (tako črni kot beli) pomenijo ničlo, debeli pa enico. Debeli pasovi so dvakrat debelejši od tankih. Napiši postopek, ki bo izpisal zaporedje ničel in enic, ki je zakodirano na papirju. S čitalnikom se premikamo po papirju s konstantno hitrostjo. Za ugotovitev hitrosti imamo pred informacijo na papirju en tanek črn pas. Za čitanje imamo na voljo naslednji funkciji:

BARVA	pove, kakšna barva je trenutno pod glavo čitalnika.
ČAS	*nam pove čas v milisekundah, ki je preteкel od začetka programa.*

Primer:



1 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 ...

Kakšen bi bil algoritem, ki bi se prilagajal spremenljivi realni hitrosti odčitavanja?

3. Za nenegativna števila n imamo definirane funkcije f, g in h takole:

$$f(n) = \begin{cases} 0 & \text{če } n=0 \\ 1 & \text{če } n=1 \\ f(n-1)+f(n-2) & \text{če } n>1; \end{cases}$$

$$h(a,b,n) = \begin{cases} a & \text{če } n=0; \\ b & \text{če } n=1; \\ h(b,a+b,n-1) & \text{če } n>1; \end{cases}$$

$$g(n) = h(0,1,n)$$

a) Izračunaj f(5) in g(5).
b) Pokaži, da za vsak a, b, c, d in e velja: h(a,b,e)+h(c,d,e)=h(a+c,b+d,e) .
c) Pokaži, da za vsak nenegativen cel n velja f(n)=g(n).

4. Neki programer sumljivih kvalitet nam je prinesel naslednji program:

```
      INTEGER A(10,10),I,J,Z
      READ(2,1)((A(I,J),
     -        J=1,10),I=1,10)
1     FORMAT(10I5)
      DO 3 I=1,10
        DO 2 J=1,10
          Z=A(I,J)
          A(I,J)=A(J,I)
          A(J,I)=Z
2       CONTINUE
3     CONTINUE
      WRITE(3,4)((A(I,J),
     -        J=1,10),I=1,10)
4     FORMAT(1X,10I10)
      CALL EXIT
      END
```

```
program t(input,output);
  var i,j,z:integer;
      a:array[1..10,1..10]
          of integer;
  begin
    for i:=1 to 10 do
      for j:=1 to 10 do
        read(a[i,j]);
    for i:=1 to 10 do
      for j:=1 to 10 do
        begin z:=a[i,j];
              a[i,j]:=a[j,i];
              a[j,i]:=z
        end;
    for i:=1 to 10 do
      begin
        for j:=1 to 10 do
          write(a[i,j]);
        writeln
      end
  end.
```

Izmisli si primerne podatke za ta program in zapiši podatke in rezultate. Ali lahko uganeš, kaj je imel programer v mislih in popraviš program tako, kot misliš, da bi moral delovati?

5. Ista naloga kot 5. naloga za tekmovalce po 1. letu pouka.

IV. Rezultati prvih petnajstih tekmovalcev v vsaki skupini

po prvem letu pouka računalništva

Mesto	Št. točk	Tekmovalec
01	111	Mirjam Lešnik, I. gimnazija Ljubljana - Bežigrad
02	104	Anton Verbovšek, I. gimnazija Ljubljana - Bežigrad
03	103	Uroš Kunaver, I. gimnazija Ljubljana - Bežigrad
04	089	Marjan Horvatič, Gimnazija Novo mesto
05	075	Ester Zimic, Šc "Vojvodina" - gimnazija Tolmin
06	074	Bojan Cestnik, I. gimnazija Ljubljana - Bežigrad
07	073	Andrej Brodnik, I. gimnazija Ljubljana - Bežigrad
07	073	Tomi Dolenc, I. gimnazija Ljubljana - Bežigrad
09	072	Dario Medoš, Gimnazija Koper
10	069	Nada Žagar, Gimnazija in ekonomska šola, Trbovlje
11	065	Gorazd Planinšič, I. gimnazija Ljubljana - Bežigrad
12	061	Jana Padežnik, Gimnazija Miloša Zidanška - Maribor
13	059	Miloš Požar, Gimnazija Nova Gorica
14	057	Simona Jaklič, I. gimnazija Ljubljana - Bežigrad
14	057	Metod Purgar, Center srednjih šol - Jesenice

po drugem letu pouka računalništva

Mesto	Št. točk	Tekmovalec
01	083	Mark Pleško, VII. gimnazija Vič - Ljubljana
02	081	Kazimir Gomilšek, Gimnazija Miloša Zidanška - Maribor
03	077	Matjaž Lampe, I. gimnazija Ljubljana - Bežigrad
04	069	Cveto Gregorc, I. gimnazija Ljubljana - Bežigrad
05	064	Milan Bizant, Gimnazija Ljubljana - Šentvid
06	054	Darko Hanžel, I. gimnazija Ljubljana - Bežigrad
07	053	Janez Bonča, I. gimnazija Ljubljana - Bežigrad
08	051	Srečko Starič, Gimnazija Novo mesto
09	047	Rado Juvan, Gimnazija-ekonomska šola, Trbovlje
10	046	Marko Ahčan, VII. gimnazija Vič - Ljubljana
10	046	Branko Premzel, Tehniška elektro, strojna in tekstilna šola Maribor
12	043	Cveto Brkič, Gimnazija Novo mesto
13	042	Borut Stariha, Prva gimnazija, Maribor
14	039	Nada Ličen, Šolski center Idrija - gimnazija Jurija Vege
14	039	Miran Ulbin, Gimnazija Miloša Zidanška - Maribor

V. Rešitve nalog za učence po prvem letu pouka računalništva

1. Program za izpis tabele najprej v fortranu, nato pa v pascalu:

```
C     program tab
        INTEGER X(21),Y(21),I,P
C       Sestavimo prvo vrstico
        DO 1 I=1,21
          X(I)=0
1       CONTINUE
        X(11)=1
C       Dokler ni prvo število v vrsti različno
C       od 0 ponavljamo
```

```
2         CONTINUE
C           Izpis vrstice
            WRITE(3,3)(X(I),I=1,21)
3           FORMAT(21I6)
C           P je prvo število v izpisani vrsti
            P=X(1)
C           Izračunamo novo vrsto ...
            Y(1)=X(1)+X(2)
            Y(21)=X(20)+X(21)
            DO 4 I=2,20
              Y(I)=X(I-1)+X(I)+X(I+1)
4           CONTINUE
C           ... in jo prepišemo v staro
            DO 5 I=1,21
              X(I)=Y(I)
5           CONTINUE
          IF(P.EQ.0)GO TO 2
        CALL EXIT
        END
```

```
program tab(output);
  const mx=21;
  var x,y:array[1..mx]of integer;
      i:integer;
  begin
    { sestavimo prvo vrstico }
    for i:=1 to mx do
      x[i]:=0;
    x[mx div 2 + 1]:=1;
    { Dokler ni prvo število v vrsti različno
       od 0 ponavljamo }
    page(output);
    repeat
      { izpisemo vrsto  }
      for i:=1 to mx do
        write(x[i]:6);
      writeln;
      { p je prvo stevilo v ze izpisani vrsti }
      p:=x[1];
      { izracunamo novo vrsto ... }
      y[1]:=x[1]+x[2];
      y[mx]:=x[mx-1]+x[mx];
      for i:=2 to mx-1 do
        y[i]:=x[i-1]+x[i]+x[i+1];
      { ... in jo prepisemo v staro }
      for i:=1 to mx do
        x[i]:=y[i]
    until p<>0;
  end.
```

2. Rešitev zapišemo (skoraj) v pascalu.
   Iz pascala se postopek tako jasno vidi, da je zapis v slovenščini nepotreben.

```
program most(output);
  type prisoten=(DA,NE);
       točka=(A,B,M);

  { ukazi za krmiljenje naprav na mostu }
  procedure odpri(x: točka); extern;
  procedure zapri(x: točka); extern;
  function tipalo(x: točka): prisoten; extern;

  procedure prehod(x,y: točka);
    { če je na strani x kakšen avto, potem
       enega spusti čez most. }
    begin
      repeat until tipalo(M)=NE;
      if tipalo(x)=DA then
        begin
          zapri(y);
          odpri(x);
          repeat until tipalo(M)=DA
        end
    end;

  begin {most}
    zapri(A); zapri(B);
    repeat
      prehod(A,B);
      prehod(B,A)
    until false
  end {most}.
```

3.

a) s(15324) = s(1532)+4 = s(153)+2+4 = s(15)+3+2+4 = s(1)+5+3+2+4 = 1+5+3+2+4 = 15

p(15324) = p(s(15324)) = p(15) = p(s(15)) = p(6) = 6

b) s izračuna vsoto cifer (v desetiškem zapisu) svojega argumenta. Utemeljitev: vsota cifer enomestnega števila (t. j. števila, ki je manjše od 10) je to število samo.

Vsoto cifer večmestnega števila pa dobimo tako, da zadnji cifri prištejemo vsoto cifer tega števila brez zadnje cifre. n mod 10 je očitno zadnja cifra števila n v desetiškem sestavu, n / 10 (celoštevilčno) pa je število n brez zadnje cifre.

c) Trditev očitno velja za n<10. Vzemimo poljuben k>9. Naj trditev velja za vse n<k. Poglejmo, če velja tudi za n=k. Ker je k>9, velja

p(k) = p(s(k)).

Ker je k>9, je s(k) (vsota cifer v številu k) gotovo manjša od k, zato lahko uporabimo hipotezo, da trditev velja za vse n<k. p(s(k)) je torej ostanek pri deljenju s(k) z 9. Pokažimo, da imata s(k) in k pri deljenju z 9 isti ostanek. Če so a[i], i=0, 1, ... m cifre števila k, potem velja

$$k = a[0]*10^{0} + a[1]*10^{1} + \dots + a[m]*10^{m}$$

in

$$s(k) = a[0] + a[1] + \dots + a[m].$$

k in s(k) imata enak ostanek pri deljenju z 9, če je njuna razlika deljiva z 9.

$$k-s(k) = a[0]*(10^{0}-1) + a[1]*(10^{1}-1) + \dots + a[m-1]*(10^{m-1}-1) + a[m]*(10^{m}-1)$$

Vsa števila oblike $(10^{i}-1)$ so očitno deljiva z 9, torej je tudi razlika k-s(k) deljiva z 9, zato pa imata k in s(k) isti ostanek pri deljenju z 9. Indukcija opravi svoje in trditev je dokazana.

4. Program izpiše tisto, kar je prečital:

9   8   7   6   5   4   3   2   1   0

Iz strukture programa se vidi, da je programer verjetno hotel izpisati zaporedje v obrnjenem vrstnem redu. V ta namen bi morala teči zanka, ki elemente zaporedja menja, le do polovice zaporedja. Popravljen program bi bil:

```
...               |   ...
M=N/2             |   for i:=1 to n div 2 do
DO 3 I=1,M        |     begin z:=t[i];
  Z=T(I)          |       t[i]:=t[j];
  T(I)=T(J)       |       t[j]:=z;
...               |   ...
```

(Druga ugibanja so seveda prav tako dobra rešitev, le program moramo pravilno popraviti.)

5. Predpostavimo, da T obstaja. Naj bo T funkcija; njena argumenta sta program in podatki zanj, njena vrednost pa je true, če se program ustavi in false, če se ne ustavi. Sestavimo s T naslednji program:

```
procedure Q(prog,podat);
  begin
    while T(prog,podat) do;
    write("OK");
  end;
```

Vprašajmo se, kaj se zgodi, če poskušamo izračunati

Q(Q,p1),

kjer je p1 nek program, ki ne potrebuje podatkov. Če bi se Q ustavil, bi se to lahko zgodilo samo če je T(Q,p1) = false, kar pomeni, da bi moral T trditi, da se Q ne ustavi. Če pa se Q ne bi ustavil, se to lahko zgodi samo, če je T(Q,p1) = true, kar pomeni, da bi T moral trditi, da se Q ne ustavi. Niti prvo niti drugo se ne more zgoditi, zato tak Q ne more obstajati. Ker pa v Q vse razen T obstaja, T ne obstaja. Q. E. D.

VI. Rešitve nalog za učence po drugem letu pouka računalništva

1. Program za izštevanje najprej v fortranu nato pa v pascalu.

```
C     Program izštevanka
        INTEGER OTROCI(30),N,M,I,J,P
C       čitanje
        READ(2,1)N,M
1       FORMAT(2I2)
C       vsi otroci so v krogu
C       predpostavljamo 1<N<31, M>0
        DO 2 I=1,N
          OTROCI(I)=1
2       CONTINUE
C       začnemo s prvim
        P=0
C       N-1 jih mora izpasti
        DO 5 I=2,N
C         vsakokrat moramo šteti do M
          DO 4 J=1,M
C           izpadlih otrok ne smemo šteti
3           CONTINUE
              P=P+1
C             štejemo v krogu (za n-tim sledi
C                                   prvi otrok)
              IF(P.GT.N)P=1
            IF(OTROCI(P).EQ.0)GO TO 3
4         CONTINUE
C         otrok P izpade
          OTROCI(P)=0
5       CONTINUE
C       poiščemo edinega neizpadlega otroka
        P=1
6       IF(OTROCI(P).NE.0)GO TO 7
          P=P+1
        GO TO 6
7       CONTINUE
C       izpis
        WRITE(3,8)N,M,P
8       FORMAT('1število otrok',I10/
     -         ' dolžina izštevanke',I5/
     -         '0lovi otrok',I13)
      CALL EXIT
      END
```

```
program izstevanka(input,output);
  const mx=30;
  var m,n,i,j,p:integer;
      otroci:array[1..mx]of boolean;
  begin
    readln(n,m);  { čitanje }
    { vsi otroci so v krogu }
    { predpostavljamo 1<n<=mx, m>0
    for i:=1 to n do otroci[i]:=true;
    { začnemo s prvim }
    p:=0;
    { n-1 otrok mora izpasti }
    for i:=1 to n-1 do
      begin
        { vsakokrat moramo šteti do m }
        for j:=1 to m do
          { izpadlih otrok ne smemo šteti }
          repeat
            p:=p+1;
            { štejemo v krogu,
                  n-temu sledi prvi }
            if p>n then p:=1;
          until otroci[p];
        { otrok p izpade }
        otroci[p]:=false
      end;
    { poiščemo edinega neizpadlega otroka }
    p:=1;
    while not otroci[p] do p:=p+1;
    { izpis }
    page(output);
    writeln(" število otrok",n:10);
    writeln(" dolžina izštevanke",m:5);
    writeln;
    writeln(" lovi otrok",p:13)
  end.
```

2. Postopek zapišemo (skoraj) v pascalu, kar ne more škoditi preglednosti.

```
program barcode(output);
  type čb=(črna,bela);
  var t,t0,zakas: integer;
      b: čb;
      podat: 0..1;

  function barva: čb; extern;
  function čas: integer; extern;

  begin
    { ignoriramo belino vse do začetka }
    repeat until barva=črna;
    { izmerimo širino prvega črnega pasu }
    t:=čas;
    repeat until barva=bela;
    t0:=čas-t; { t0 je čas za prelet ničle }
    { pravo čitanje se začenja }
    repeat
      t:=čas;
      b:=barva;
      { počakamo, da čitalnik pride do
          spremembe barve }
      repeat until barva<>b;
      zakas:=čas-t;
      { zakas je čas potovanja čez zadnji pas }
      { odločimo se, ali je to 0 ali 1 }
      if zakas>1.5*t0 then
        begin
          podat:=1;
          { popravimo vzorčni čas }
          t0:=zakas/2
        end
      else
        begin
          podat:=0;
          { popravimo vzorčni čas }
          t0:=zakas
        end;
      write(podat)
    until false
  end.
```

Osnovni postopek, ki bi deloval le pri konstantni hitrosti, ne potrebuje popravljanja vzorčnega časa. V program bi lahko vgradili še test za konec podatkov (npr. če se barva zelo dolgo ne spremeni), vendar tega naloga ne zahteva.

3.

a) f(5) = f(4)+f(3) = f(3)+f(2)+f(3) =
f(2)+f(1)+f(2)+f(3) =
f(1)+f(0)+f(1)+f(2)+f(3) =
1+0+1+f(1)+f(0)+f(3) = 1+0+1+1+0+f(2)+f(1) =
1+0+1+1+0+f(1)+f(0)+f(1) = 1+0+1+1+0+1+0+1 =
5,

kar se seveda lažje izračuna, če zapišemo tabelo:

n	0	1	2	3	4	5	6	7...
f(n)	0	1	1	2	3	5	8	13...

<u>g(5)</u> = h(0,1,5) = h(1,1,4) = h(1,2,3) = h(2,3,2) = h(3,5,1) = <u>5</u>

b) Če so a, b, c in d, e poljubna nenegativna cela števila, je treba dokazati, da velja

f(a,b,e)+f(c,d,e) = f(a+c,b+d,e).

Trditev bomo dokazali z indukcijo:
- Pri e=0 in e=1 trditev očitno velja.
- Naj bo k poljubno naravno število večje od 1. Vzemimo, da trditev velja za vse e<k. Dokažimo, da tedaj velja tudi pri e=k.

h(a,b,k)+h(c,d,k) = h(b,a+b,k-1)+h(d,c+d,k-1) = h(b+d,a+b+c+d,k-1),

h(a+c,b+d,k) = h(b+d,a+c+b+d,k-1).

S tem je trditev dokazana.

c) Dokazati moramo, da za vsak nenegativen cel n velja

f(n) = g(n).

Tudi to trditev bomo dokazovali z indukcijo. Veljavnost je očitna za n med 0 in 2. Spet naj bo k neko naravno število večje od 2 in predpostavimo, da trditev velja za vse n<k. Za dokaz, da trditev velja tudi pri n=k, bomo potrebovali trditev pod točko b) :

f(k) = f(k-1)+f(k-2) = h(0,1,k-1)+h(0,1,k-2) = h(1,1,k-2)+h(0,1,k-2) = h(1,2,k-2)

g(k) = h(0,1,k) = h(1,1,k-1) = h(1,2,k-2)

Prepričali smo se, da tako f kot g računata fibonaccijeva števila. Opazimo lahko, da je pri tem g v primerjavi z f mnogo učinkovitejša.

4. Program prečita celoštevilčno matriko velikosti 10x10. Izpiše jo nespremenjeno. Programer je hotel verjetno matriko transponirati, kar bi dosegel, če bi program popravil takole:

```
   . . .                    |   . . .
   DO 3 I=1,10              |   for i:=1 to 10 do
     DO 2 J=1,I             |     for j:=1 to i-1 do
       Z=A(I,J)             |       begin
   . . .                    |   . . .
```

Še lažje pa je odstraniti zanke in zamenjati indekse pri izpisu:

```
     . . .                    | . . .
     READ(2,1)((A(I,J),J=...  |read(a[i,j]);
1    FORMAT ...               |{ zanki za izpis }
     WRITE(3,4)((A(J,I),J=... |      write(a[j,i];
     . . .                    | . . .
```

(Veljavno rešitev dobimo tudi, če si mislimo, da je programer hotel kaj drugega, le program je treba pravilno popraviti.)

5. Glej 5. nalogo pri nalogah za učence po enem letu pouka.

<u>TABELA 1</u>: Število udeležencev na vseh treh tekmovanjih

	1977	1978	1979
po 1. letu	?	52	56
po 2. letu	?	27	36
skupaj	47	79	92

Opomba: V letu 1977 tekmovanje ni bilo ločeno na dve skupini.

<u>TABELA 2</u>: Število tekmovalcev in povprečen uspeh po šolah

Šola	Št. tekmovalcev in povprečno št. točk po 1. letu		po 2. letu		Št. prijavljenih tekmovalcev
Center srednjih šol - Jesenice	2	51.00	2	7.50	6
CSŠ Črnomelj - gimnazija splošne smeri	1	18.00	0	---	3
Ekonomska srednja šola Črnomelj	2	21.50	0	---	2
Elektrotehniška srednja šola Krško	1	19.00	0	---	2
Elektrotehniška šola v Ljubljani	7	24.29	0	---	20
Gimnazija "Boris Ziherl" Škofja loka	4	9.50	4	14.50	10
Gimnazija Kočevje	2	26.50	0	---	3
Gimnazija Koper	5	39.00	1	34.00	12
Gimnazija Ljubljana - Šentvid	2	35.00	2	41.50	8
Gimnazija Miloša Zidanška - Maribor	2	51.50	3	48.00	6
Gimnazija Nova Gorica	4	45.00	0	---	5
Gimnazija Novo mesto	3	54.33	4	33.50	10
Gimnazija Trbovlje	4	42.25	1	47.00	7
I. gimnazija Ljubljana - Bežigrad	8	82.50	5	57.40	13
Prva gimnazija Maribor	0	---	6	26.50	6
Šc Idrija - gimnazija Jurija Vege	0	---	1	23.00	1
Šc "Vojvodina" - gimnazija Tolmin	1	75.00	1	39.00	2
Tehniška el. str. in tekstilna šola Maribor	0	---	3	23.33	3
Tehniška strojna in elektro šola, Trbovlje	7	20.00	0	---	8
Tehniška tekstilna šola, Kranj	1	40.00	0	---	1
VII. gimnazija Ljubljana Vič	0	---	3	49.67	3
Skupaj	56	40.00	36	34.50	137
Število šol	17		13		21
Število gimnazij	11		11		14
Število tehniških srednjih šol	6		2		7