



ZVEZA ZA TEHNIČNO KULTURO SLOVENIJE

11. FESTIVAL RAČUNALNIŠTVA

20. državno tekmovanje osnovnošolcev
v znanju računalništva
Ljubljana, 2. 4. 2005

PASCAL – 1. skupina (7. razred - 9 letna OŠ)

Rešitve nalog za tekmovalce

1. naloga (maksimalno 25 točk):

Napiši program, ki preveri veljavnost registrske tablice za avto.

Veljavne so tablice, ki ustrezajo naslednjim pogojem:

- prva dva znaka tablice sta oznaka mesta, torej LJ, MB, CE, GO, NM, KP, KK, KR, PO, SG ali MS
- tretji znak je presledek (namesto grba)
- presledku sledi najmanj 3 in največ 6 znakov; ti znaki so lahko velike tiskane angleške brke in številke, tretji od teh znakov pa je lahko tudi črtica (-)

Program prebere tablico in izpiše samo VELJAVNA ali NEVELJAVNA.

Primer vnosa:

LJ 35-AF8

Izpis:

VELJAVNA

Rešitev: (Mitja Trampuš)

```
program tablice;
const mesta = 'LJ.MB.CE.GO.NM.KP.KK.KR.PO.SG.MS';
var tablica : string;
    OK : boolean;
    i : integer;

begin
  write('Vpisi registrsko tablico: ');
  readln(tablica);

  OK := pos( copy(tablica,1,2), mesta ) mod 3 = 1;

  OK := OK and (tablica[3]=' ');

  OK := OK and (length(tablica)>3+2) and (length(tablica)<3+7);

  for i := 4 to length(tablica) do
    if not (
      (tablica[i] in ['A'..'Z']) or
      (tablica[i] in ['1'..'9']) or
      ( (tablica[i]='-') and (i=6) )
    ) then OK := false;

  if OK then writeln('VELJAVNA') else writeln('NEVELJAVNA');
end.
```

2. naloga (maksimalno 25 točk):

Matija zelo rad igra košarko in tudi nastopa na tekmah, zato si želi program, ki bi mu vodil statistiko metov za dve in tri točke. Napiši ga.

Program naj prebere zaporedje Matijevih metov v obliki zaporedja števil -3, -2, 2 in 3. Pri tem 2 pomeni zadet met za 2 točki, -2 pa zgrešenega in podobno za mete za 3 točke. Vnos zaporedja se konča z vnosom nesmiselnega števila (torej ne -2, 2, -3 ali 3). Na koncu naj program izpiše statistiko v obliki, kot jo kaže spodnji primer. Odstotke pri tem zaokroži na celo število.

Matija na vsaki tekmi vsaj enkrat vrže za 2 in vsaj enkrat za 3 točke.

Primer vnosa:

```
2
2
-3
3
-2
2
-2
2
0
```

Izpis:

Meti za 2 točki: 4/6 (67%)

Meti za 3 točke: 1/2 (50%)

Opomba:

prva vrstica izpisa pomeni, da je Matija metal za dve točki šestkrat, zadel pa štirikrat, kar je 67% vseh metov za 2 točki. Podobno je v drugi vrstici.

Rešitev: (Mitja Trampuš)

```
program kosarka;
var meti : array[-3..3] of integer;
    i : integer;

begin
    while true do begin
        readln(i);
        if (abs(i)<>2) and (abs(i)<>3) then break;
        inc(meti[i]);
    end;

    writeln('Meti za 2 točki: ',meti[2], '/', meti[2]+meti[-2], '
(' , round(meti[2]/(meti[2]+meti[-2])*100), '%) ');
    writeln('Meti za 3 točke: ',meti[3], '/', meti[3]+meti[-3], '
(' , round(meti[3]/(meti[3]+meti[-3])*100), '%) ');
end.
```

3. naloga (maksimalno 25 točk):

Na veliki kokošji farmi Yatta imajo velike težave z nestandardnimi velikostmi škatel za jajca. Zgodi se jim na primer, da imajo 4125 jajc, ki jih je treba zapakirati v škatle, od katerih ima vsaka prostora za 13 jajc.

Napiši jim program, ki bo vprašal za število jajc, ki jih je potrebno zapakirati, ter za število jajc, ki grejo v eno škatlo, ter nato izpisal dva podatka: koliko škatel bo Yatta tega dne potrebovala za pakiranje vseh jajc in koliko praznih prostorčkov bo ostalo v zadnji škatli. Ni namreč nujno, da se vse lepo izide in je jajc ravno toliko, da do konca zapolnijo tudi zadnjo škatlo. Primer vnosa in izpisa:

```
Koliko jajc? 14
Koliko jih gre v eno skatlo? 3
Porabili bodo 5 skatel, vendar bo v zadnji skatli 1 prostorckov praznih.
```

Če so števila izbrana tako, da ne bo ostajalo praznih prostorčkov v škatlah, naj program ne izpiše tistega o praznih prostorčkih. Primer takega vnosa in izpisa:

```
Koliko jajc? 15
Koliko jih gre v eno skatlo? 5
Porabili bodo 3 skatel.
```

Rešitev: (Mitja Trampuš)

```
Program jajca;
var n,v : integer;
begin
  write('Koliko jajc? '); readln(n);
  write('Koliko jih gre v eno skatlo? '); readln(v);
  write('Porabili bodo ', (n+v-1) div v, ' skatel');
  if n mod v <> 0 then
    writeln(', vendar bo v zadnji skatli ',v-(n mod v), ' prostorckov praznih.')
  else
    writeln('.');
end.
```

4. naloga (maksimalno 25 točk):

Peter mora priti iz službe domov, vendar mu je že skoraj zmanjkalo bencina.

Med službo in domom ima več bencinskih črpalk; pomagaj mu odločiti se, na katero naj gre natočiti, da bo čim prej doma.

Program naj najprej prebere, koliko kilometrov lahko Peter še prevozi brez dotakanja bencina. V naslednji vrstici prebere, koliko je črpalk.

Nato v vsaki naslednji vrstici prebere dve števili: prvo je razdalja (v km) med službo in črpalko, drugo med črpalko in domom.

Program naj izpiše stavek oblike 'Peter bo najhitreje doma, če se ustavi na črpalki št. 5'

Seveda se lahko zgodi, da je kakšna črpalka predaleč od službe in se do nje sploh ni mogoče pripeljati s preostalim bencinom.

Na tako črpalko se mu v nobenem primeru ne splača iti.

Lahko se tudi zgodi, da so vse črpalke predaleč ali pa jih sploh ni. V tem primeru izpiši samo 'Vse črpalke so predaleč'.

Primer vnosa:

```
20
4
30 1
7 45
2 90
78 16
```

Izpis:

Peter bo najhitreje doma, če se ustavi na crpalki st. 2

Rešitev: (Mitja Trampuš)

```
program bencin;
var najPumpa, najRazd, ostanek, st, a, b, i : integer;

begin

  readln(ostanek); readln(st);

  najPumpa := 0; najRazd:=32000;
  for i := 1 to st do begin
    readln(a,b);
    if (a<=ostanek) and (a+b<najRazd) then begin
      najPumpa:=i;
      najRazd:=a+b;
    end;
  end;

  if najPumpa=0 then
    writeln('Vse crpalke so predalec')
  else
    writeln('Peter bo najhitreje doma, ce se ustavi na crpalki st. ',najPumpa);
  end.
```



ZVEZA ZA TEHNIČNO KULTURO SLOVENIJE

11. FESTIVAL RAČUNALNIŠTVA

20. državno tekmovanje osnovnošolcev
v znanju računalništva
Ljubljana, 2. 4. 2005

PASCAL – 2. skupina (8. in 9. razred - 9 letna OŠ)

Rešitve nalog za tekmovalce

1. naloga (maksimalno 25 točk):

Napiši program, ki bo oponašal žreb pri loteriji. Program najprej prebere 10 različnih števil (vsako v svoji vrstici; pravilnosti vnosa ni treba preverjati), nato pa jih naključno izžreba pet ter jih izpiše na ekran. Vsaka številka je lahko izžrebana le enkrat.

Primer vnosa:

```
2
23
645
43
74
23
84
85
54
247
```

Izpis:

```
54
23
247
74
645
```

Rešitev: (Mitja Trampuš)

```
program loterija;
var
  i,p,l : integer;
  a: array[1..10] of integer;

begin
  l:=10;

  for i := 1 to l do readln(a[i]);

  writeln; writeln('Izzrebane številke:');
  randomize;
  for i := 1 to 5 do begin
    p := random(l)+1;
    write(a[p], ' ');
    a[p]:=a[l]; dec(l);
  end;
end.
```

2. naloga (maksimalno 25 točk):

Slabo znano dejstvo je, da so najinteligentnejša bitja na svetu podgane. Ravno kar se pripravljajo, da bi izdale knjigo, v kateri bi bilo zajeto vse njihovo dosedanje znanje. Encyclopaedia Podgannica, kot se bo knjiga imenovala, bo imela ogromno strani (vendar manj kot 30 000), zato podgane zanima, koliko katerih števk bodo potrebovale, da strani knjige oštevilčijo. Na prvi in nato na vsaki sedemnajsti naslednji strani bo celostranska slika poslancev, ministrov, ravnateljev, mirovnikov in predstavnikov civilne družbe. Podgane številke strani na teh straneh ne bodo zapisale.

Napiši program, ki vpraša, koliko strani bo imela nova knjiga, nato pa zapiše, koliko posameznih števk bodo podgane rabile za oštevilčevanje strani knjige.

Primer vnosa:

11

Izpis:

```
1-krat stevka 0
3-krat stevka 1
1-krat stevka 2
1-krat stevka 3
1-krat stevka 4
1-krat stevka 5
1-krat stevka 6
1-krat stevka 7
1-krat stevka 8
1-krat stevka 9
```

(Opomba: oštevilčiti bo treba strani 2,3,4,5,...,10,11, ne pa prve, ker bo tam slika)

Rešitev: (Mitja Trampuš)

```
program Podgannica;
var
  stStrani, stran, i : longint;
  stevke : array[0..9] of longint;
begin
  writeln('Koliko strani bo imela enciklopedija?');
  readln(stStrani);

  for stran := 1 to stStrani do begin
    if stran mod 17 = 1 then continue else i:=stran;
    while i<>0 do begin
      inc(stevke[i mod 10]);
      i:=i div 10;
    end;
  end;

  for i := 0 to 9 do
    writeln(stevke[i], '-krat stevka ', i);
  readln;
end.
```

3. naloga (maksimalno 25 točk):

V tropskem gozdu raste redka rastlina, Bitus Computii, čisto pri tleh in le na sončnih predelih. Napiši program, ki prebere opis tropskega gozda in nato prikaže, kje ima Bitus Computii ugodne pogoje za rast. Opis gozda prebereš iz datoteke gozd.in.

V prvi vrstici piše, kako širok je gozd, v drugi pa, kako visok. Vse naslednje vrstice so dolge do 70 znakov in so sestavljene iz znakov # in presledkov. Tam, kjer najdeš znak #, raste veja drevesa, zato svetloba tam ne more do tal (sonce sveti navpično navzdol) in Bitus pod to vejo ne more rasti, saj potrebuje svetlobo. Ni nujno, da so vse vrstice enake dolžine, kot je širina gozda.

Rezultat izpiši na zaslona v dveh vrsticah. Tam, kjer Bitus lahko raste, ga nariši (O nad znakom |), tam kjer ne more, pa nariši samo piko v spodnjo vrsto. Za boljše razumevanje glej primer.

Primer podatkov:

50

7

```

      x               xxxxx  x
      xxx             xxxxxxx
    xx  xxxxx        xxxxx  x  xx
  xxxx xxxxxxxx      x      xxxxx      x
xxxxxxxxxxxxxxxxxxx  xxx      xxx      xxx
xxxxxxxxxxxxxxxxxxx  xxxxx      x      xxxxx
    xx  xxx      x      x      x
```

Izpis:

```

                000      00                0000      000
.....|...|...|.....|...|...|.....|...|...|...|
```

Opomba: skrajno desno drevo se konča že v 47. stolpcu, gozd pa je širine 50, zato lahko na desni rastejo še 3 Bitusi.

Rešitev: (Mitja Trampuš)

```
program gozd;
var data : text;
    sirina, visina, i, j : integer;
    svetlo : array[1..70] of boolean;
    vrsta : string;

begin
    assign(data, 'w:\naloge\gozd.in'); reset(data);
    readln(data, sirina); readln(data, visina);

    {na zacetku je po vsej sirini svetlo}
    for i := 1 to sirina do svetlo[i] := true;

    for i := 1 to visina do begin
        readln(data, vrsta);
        {ce v kaksnem stolpcu najdemo vejo, potem si to oznacimo}
        for j := 1 to sirina do
            svetlo[j] := svetlo[j] and ( (length(vrsta)<j) or (vrsta[j]=' ') );
        end;

        {narisemo rezultat}
        writeln;
        for i := 1 to sirina do if svetlo[i] then write('O') else write(' ');
        writeln;
        for i := 1 to sirina do if svetlo[i] then write('|') else write('.');
    end.
```

4. naloga (maksimalno 25 točk):

Arabska pisava je znana po tem, da ne poteka od leve proti desni kot naša, ampak od desne proti levi. Napiši program, ki bo pomagal Arabcem pisati na ta način.

Program naj bere s tipkovnice črke in jih izpisuje na zaslon s skrajnega desnega roba proti levemu. Ko tega doseže, naj skoči v naslednjo vrsto. Poleg črk naj program prepozna še pritisk tipke ENTER (koda 13), ki povzroči skok na desni rob naslednje vrstice, in escape (koda 27), ki povzroči izhod iz programa.

Arabci nikoli ne pritiskajo tipk backspace, delete, puščic in podobnega, zato jih tvojemu programu ni treba prepoznati. Lahko upoštevaš, da bo besedila največ za 20 vrstic.

Rešitev: (Mitja Trampuš)

```
program arabci;
uses crt;
var x,y : integer;
    c : char;

begin

  x:=80; y:=1;
  repeat
    c:=readkey;
    if c=#27 then exit;
    if c=#13 then begin
      inc(y);
      x:=80;
    end else begin
      gotoxy(x,y); write(c);
      dec(x); if x=0 then begin x:=80; inc(y); end;
    end;
  until false;

end.
```