21. siječnja 2013.

Infokup 2013

Školsko natjecanje / Osnovna škola (5. i 6. i 7. i 8. razred)

Algoritmi (Basic/Pascal/C/C++)

OBJAŠNJENJA ZADATAKA

Agencija za odgoj i obrazovanje
Education and Teacher Training Agency

MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

5.1. Zadatak: Smart

Autor: Nikola Dmitrović[1]

Smart je zadatak namijenjen učenicima petih razreda koji se prvi put susreću s programiranjem u natjecateljskom okruženju. Rješenje zadatka se sastoji od učitavanja šest znamenki s ulaza te ispisivanja svake druge učitane znamenke. Opis rješenja je vrlo jednostavan.

```
učitaj(a);
učitaj(b);
učitaj(c);
učitaj(d);
učitaj(e);
učitaj(f);
ispiši(b);
ispiši(d);
ispiši(f);
```

Potrebno znanje: naredba učitavanja, naredba ispisa

Kategorija: ad hoc

5.2. Zadatak: Popust

Autor: Nikola Dmitrović

Rješenje zadatka se krije u rečenici iz teksta koja glasi: „Naime, obećali su da će svaki račun izdan u toj trgovini **umanjiti za zbroj sata i minute** u kojima je taj račun izdan“. Doslovno prevođenje ove rečenice u odabrani programski jezik nosi 4 od 5 test primjera. Za sve bodove treba uočiti još jednu rečenicu unutar teksta koja glasi: “Ako vrijednost računa nakon umanjenja bude negativna, tada se za novu vrijednost uzima nula.“

```
učitaj(C);
učitaj(S);
učitaj(M);
Cnova:=C-(S+M);
ako je Cnova>=0 tada
      ispiši(Cnova)
inače
      ispiši(0);
```

Potrebno znanje: naredba učitavanja, naredba ispisa, računska operacija oduzimanja, osnovni oblik naredbe odlučivanja

Kategorija: ad hoc

[1] član Državnog povjerenstva, voditelj natjecanja u kategoriji primjena algoritama (programski jezik Basic/Pascal/C/C++), profesor informatike u XV. gimnaziji, Zagreb

5.3 Zadatak: Peking — Autor: Nikola Dmitrović

Prilikom rješavanja ovog zadatka treba paziti na pravilno osmišljavanje višestruke naredbe odlučivanja, na pravilno postavljanje granica pojedinih segmenata te točno prepisivanje/kopiranje zadane vrijednosti koja se spominje u zadatku.

```
učitaj(IKZ);
ako je IKZ<=50 tada
      ispiši('dobra kvaliteta zraka')
inače
      ako je IKZ<=100 tada
            ispiši('umjerena kvaliteta zraka')
      inače
            ako je IKZ<=150 tada
                  ispiši('zrak nezdrav za osjetljive grupe')
            inače
                  ako je IKZ<=200 tada
                        ispiši('nezdrav zrak')
                  inače
                        ako je IKZ<=300 tada
                              ispiši('vrlo nezdrav zrak')
                        inače
                              ispiši('opasan zrak');
```

Potrebno znanje: naredba učitavanja, naredba ispisa, višestruka naredba odlučivanja

Kategorija: ad hoc

6.1. Zadatak: Napolitanke — Autor: Nikola Dmitrović

Napolitanke su zadatak koji je prvenstveno namijenjen učenicima šestih razreda koji se prvi put susreću s programiranjem u natjecateljskom okruženju te ostalim učenicima za zagrijavanje. Rješenje se svodi na osnovno korištenje računskih operacija.

```
učitaj(R);
učitaj(S);
učitaj(K);
učitaj(D);
učitaj(V);
napolitanki:=R*S*K-D-V
ispiši(napolitanki);
```

Potrebno znanje: naredba učitavanja, naredba ispisa, osnovne računske operacije množenja i oduzimanja

Kategorija: ad hoc

6.2. Zadatak: Peking Autor: Nikola Dmitrović

Vidi pojašnjenje pod 5.3.

6.3. Zadatak: Joker Autor: Nikola Dmitrović

Kako je broj vrijednosti koje se zadaju i koje se koriste unaprijed zadan, zadatak možemo riješiti „pješice“, posebno računajući svaku vrijednost znamenke jedinica (6 brojeva nije prevelik broj za taj postupak) te od njih kreirati broj na opisani način u tekstu. Nakon toga, ispis traženih vrijednosti nije poseban problem.

Potencijalni problem pri rješavanju ovog zadatka je činjenica da se s ulaza zadaje sedam brojeva, a koristi samo prvih šest. Taj problem se na gore opisani način vrlo lako može izbjeći. Rješenje ćemo opisati u sljedećem obliku.

```
učitaj(a); aj:=a mod 10; // mod je operator ostatka pri djeljenju
učitaj(b); bj:=b mod 10;
učitaj(c); cj:=c mod 10;
učitaj(d); dj:=d mod 10;
učitaj(e); ej:=e mod 10;
učitaj(f); fj:=f mod 10;
učitaj(g);
joker:=aj*100000+bj*10000+cj*1000+dj*100+ej*10+fj;
ispiši(aj+bj+cj+dj+ej+fj);
ispiši(joker mod 101);
```

Zadatak možemo rješiti na kraći i elegantniji način primjenom algoritma za kreiranje broja uzastopnim dodavanje novih znamenki broju s desne strane. Naime, ako želimo na postojeći broj, s desne strane dodati novu znamenku, tada postojeći broj trebamo pomnožiti s deset i zbrojiti s novom znamenkom.

```
zbj:=0; joker:=0; // inicijaliziramo zbroj_jedinica i joker broj na nula
za i:=1 do 6 radi
{
    učitaj(broj);
    j:=broj mod 10;
    zbj:=zbj+j;
    joker:=joker*10+j;
};
učitaj(sedmi); // učitavanje sedmog broja
ispiši(zbj);
ispiši(joker mod 101);
```

Potrebno znanje: naredba učitavanja, naredba ispisa, operator ostatka pri djeljenju, operator množenja (dodatno: naredba ponavljanja, algoritam za kreiranje broja)

Kategorija: ad hoc

7.1. Zadatak: Superkuvar Autori: Sanja Grabusin[2]/Nikola Dmitrović

Superkuvar je zadatak koji je prvenstveno namijenjen učenicima sedmih razreda koji se prvi put susreću s programiranjem u natjecateljskom okruženju te ostalim učenicima za zagrijavanje. Rješenje se sastoji od zbrajanja brojeva u prva tri retka ulaza, zbrajanja brojeva u sljedeća tri retka te uspoređivanja dobivenih zbrojeva.

```
za i:=1 do 3 radi
{
      učitaj(a,b,c);
      OcEdi:=OcEdi+a+b+c;
}
za i:=1 do 3 radi
{
      učitaj(a,b,c);
      OcTin:=OcTin+a+b+c;
}
ako je OcTin>OcEdi tada
  ispiši('Tin ',OcTin)
inače
    ako je OcTin<OcEdi tada
       ispiši('Edi ',OcEdi)
    inače
        ispiši('Nema pobjednika ', OcTin);
```

Potrebno znanje: naredba učitavanja, naredba ispisa, naredba odlučivanja, naredba ponavljanja.

Kategorija: ad hoc

[2] član Državnog povjerenstva, profesorica informatike u Gimnaziji Požega, Požega

7.2. Zadatak: Melman

Autor: Nikola Dmitrović

Cijeli zadatak se može sažeti na jednu rečenicu iz teksta koja kaže da toplomjer „pamti najveću izmjerenu temperaturu nakon prvog mjerenja i točno vrijeme kada je ta temperatura izmjerena“.

Zadatak se sastoji od dva dijela. U prvom dijelu treba odrediti najvišu izmjerenu temperaturu u zadanih n mjerenja za što je dovoljno pronaći maksimalnu vrijednost među n učitanih prirodnih brojeva.

Drugi dio, određivanje sata i minute kada je izmjerena najviša temperatura zahtjeva malo više posla. Prvo, treba istovremeno s određivanjem maksimuma odrediti redni broj mjerenja (X) u kome se izmjerila najviša temperatura. Kada se to odredi, potrebno je na početno vrijeme prvog mjerenja dodati X puta po pola sata. To možemo postići tako da početno vrijeme pretvorimo u broj minuta proteklih od početka dana, tomu dodamo X*30 minuta te zatim ponovno odredimo konkretan sat i minutu.

Rješenje možemo opisati na sljedeći način.

```
učitaj(S); učitaj(M); // trenutak prvog mjerenja
učitaj(N); // broj narednih N mjerenja
max:=1; X:=1;
za i:=1 do N radi //tražimo najveću temperaturu nakon prvog mjerenja
{
      učitaj(T);
      ako je T>max tada
      {
            max:=T;
            X:=i;
      };
};
vrijeme:=S*60+M+X*30; // određivanje ukupno vremena izraženog u minutama
ispiši(max);
ispiši(vrijeme div 60); // broj sati u ukupnom broju minuta
ispiši(vrijeme mod 60); // ostatak minuta
```

Potrebno znanje: naredba učitavanja, naredba ispisa, naredba ponavljanja, određivanje maksimalne vrijednosti

Kategorija: ad hoc

7.3. Zadatak: Luka

Autor: Nikola Dmitrović

Zamislite da ste na mjestu Luke Modrića. Naravno, s puno, puno manje novca na računu! Što bi učinili da vam se takva situacija stvarno dogodila? Jedna od ideja koja bi vam pala na pamet je da između svih knjiga prvo potražite najtanju knjigu crvenih korica te je stavite na policu na prvo mjesto. Zatim bi potražili najtanju knjigu bijelih korica te nju postavili na drugo mjesto, a zatim bi pronašli najtanju knjigu plavih koricu koja bi svoje mjesto na polici pronašla na trećoj poziciji. Ovih nekoliko koraka bi nastavili ponavljati sve dok ne bi sve knjige složili na policu. Sada nam samo ostaje da to pravilno nakodiramo.

```
// boja je niz znakova u koji ćemo spremati boju, a debljina niz prirodnih brojeva maksimalne duljine 33 komponente u koji ćemo spremati debljinu knjige s oznakom „i“;
//knjige je niz u koji ćemo na odgovarajuću poziciju zapisati oznaku knjige koja će doći na tu poziciju
učitaj(n);

za i:=1 do n radi
      učitaj(boja[i],debljina[i]);

za i:=1 do n radi
{
      //biramo boju u ovisnosti o trenutnom mjestu na polici
      ako je i mod 3=1 tada znak:='C';
      ako je i mod 3=2 tada znak:='B';
      ako je i mod 3=0 tada znak:='P';
      min:=100;
      za j:=1 do n radi //tražimo najtanju knjigu u „znak“ boji
            ako je (boja[j]=znak) i (debljina[j]<min) tada
                  {
                        min:=debljina[j];
                        gdje:=j;
                  };
      debljina[gdje]:=101; //ova knjiga se više ne gleda
      boja[gdje]:='-';     // ova knjiga se više ne gleda
      knjige[i]:=gdje; //postavi knjigu na policu
};
za i:=1 do n radi
      ispiši(knjige[i]);
```

Potrebno znanje: naredba učitavanja, naredba ispisa, naredba ponavljanja, algoritam traženja minimuma

Kategorija: simulacija

8.1. Zadatak: Bubnjevi

Autor: Matija Milišić[3]

U ovom zadatku potrebno je naći udaljenost od najbližeg susjeda. Za to trebamo varijablu *R* u kojoj ćemo zapamtiti traženi rezultat. Prolazimo kroz popis katova naredbom ponavljanja. Kod svakog kata s popisa izračunamo udaljenost od Ivanovog. Udaljenost je apsolutna vrijednost razlike brojeva tih katova. Sadržaj varijable *R* mijenjamo onda kada udaljenost nekog kata bude manja od udaljenosti *R*. To radimo naredbom odlučivanja. Kako bi ovo rješenje radilo, potrebno je na početku programa pravilno namjestiti početnu vrijednost varijable *R*. Ta vrijednost mora biti veća od konačnog rezultata. Dovoljno je postaviti *R* na vrijednost 15 (broj katova) jer je konačni rezultat uvijek manji od tog broja.

Potrebno znanje: naredba odlučivanja, naredba ponavljanja

Kategorija: ad hoc

8.2. Zadatak: Blago

Autor: Antun Razum[4]

Ovaj zadatak riješit ćemo tako da ćemo se kretati po matrici znakova upravo kako je opisano u zadatku. Za to će nam biti potrebne tri pomoćne varijable koje će nam služiti za pamćenje trenutnog retka i stupca u kojem se nalazimo te količinu blaga koju smo do tada skupili. Sav kod za kretanje po matrici nalazit će se unutar for petlje koja će prolaziti po nizu znakova u koji smo spremili upute za kretanje. Na početku petlje provjeravamo nalazi li se na trenutnom polju broj. Ako se nalazi povećamo brojač petlje za taj broj koji se nalazi na polju. Odmah nakon toga moramo provjeriti jesmo li možda s brojačem prešli *K* jer ako jesmo to označava kraj i moramo izaći iz petlje kako ne bi čitali znakove koji se ne nalaze u nizu uputa. Zatim pogledamo koji se znak nalazi u nizu uputa na mjestu brojača te sukladno očitanom znaku promijenimo trenutni redak odnosno stupac u kojem se nalazimo. Nakon toga preostaje nam provjeriti nalazi li se na trenutnom polju blago. Ukoliko se nalazi, brojač kojim pamtimo koliko smo blaga do sad skupili povećamo za jedan. Na posljetku, izvan petlje, ispišemo koliko smo skupili blaga, što je rješenje zadatka.

Potrebno znanje: naredba odlučivanja, naredba ponavljanja, nizovi znakova (string), matrice znakova

Kategorija: simulacija

[3] student Fakulteta elektrotehnike i računarstva, osvajač brončane medalje na IOI 2011., srebrne medalje na CEOI 2011. te dviju brončanih medalja na IMO 2011. i IMO 2012, dvostruki državni prvak iz informatike (2. i 4. razred)

[4] student Fakulteta elektrotehnike i računarstva, osvajač srebrnih medalja na IOI 2012. i CEOI 2012., dvostruki državni prvak iz informatike (1. i 3. razred)

8.3. Zadatak: Bilbo

Autor: Nikola Dmitrović

Najteži zadatak na ovoj razini natjecanja. Osnovni problem u ovom zadatku je kako kreirati dva troznamenkasta broja od 6 ponuđenih znamenki. Treba pronaći način kako odrediti znamenku stotica, desetica i jedinica za oba broja, a da pri tome ne ponovimo ni jednu znamenku. Jedno parcijalno rješenje možemo opisati na sljedeći način. Najveću zadanu znamenku postavimo na mjesto stotice većeg broja, sljedeću po veličini na mjesto stotice drugog broja i tako za sva preostala mjesta u oba broja. Npr., ako su zadane znamenke 4, 2, 0, 2, 0 i 1 tada su brojevi 420 i 210.

U nastavku je opisan jedan opći način raspoređivanja znamenki, kreiranja i određivanja traženih brojeva. Posebno treba paziti da su oba broja troznamenkasta i da je veći od dva broja najveći takav.

```
učitaj(a); // a je niz u koji zapišemo 6 zadanih znamenki
max:=0; tko1:=100; tko2:=100;
za i:=1 do 6 radi
  za ii:=1 do 6 radi
    za iii:=1 do 6 radi
      za iiii:=1 do 6 radi
        za iiiii:=1 do 6 radi
          za iiiiii:=1 do 6 radi
          {
            // u obzir uzimamo samo onaj poredak u kome se svaka zadana znamenka
            // pojavljuje točno jednom
            za f:=1 do 6 radi
                  odabir[f]:=0;
            odabir[i]:=1;odabir[ii]:=1;
            odabir[iii]:=1;odabir[iiii]:=1;
            odabir[iiiii]:=1; odabir[iiiiii]:=1;
            koliko:=0;
            za f:=1 do 6 radi
                  koliko:=koliko+odabir[f];

            ako je koliko=6 tada // ako je koliko 6, tada nije bilo ponavljanja
            {
                  n1:=a[iiiiii]*100+a[iiiii]*10+a[iiii];
                  n2:=a[iii]*100+a[ii]*10+a[i];

                  ako je (n1>=100) i (n2>=100) i (n1+n2<1000) tada
                  {
                        ako je (n1+n2>max) tada
                        {
                              max:=n1+n2;
                              tko1:=n1;
                              tko2:=n1;
                        };
                        ako je (n1+n2=max) i (n1>tko1) tada
```

```
                        {
                              tko1:=n1;
                              tko2:=n1;
                        };
                  };
            };
};
ako je max<>0 tada
{
      ispiši(max);
      ispiši (tko1);
      ispiši (tko2);
}
inače
      ispiši('Bilbo');
```

Potrebno znanje: višestruka naredba ponavljanja, nizovi, algoritam određivanja permutacija niza brojeva

Kategorija: ad hoc