



23. siječnja 2012. od 14:30 do 16:30

Infokup 2012

Školsko natjecanje / Osnovna škola (7. i 8. razred razred)
Algoritmi (Basic/Pascal/C/C++)

OBJAŠNJENJA ZADATAKA

Sponzori



Medijski pokrovitelji



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

udruga mladih programera
dump



Zadatak: F1

Rješenje ovog zadatka se na kraju svodi na oduzimanje dva broja. Mali problem predstavlja činjenica da se za učitavanje brojeva treba koristiti niz te kako odrediti koja dva broja treba oduzeti. Jedan od načina je sljedeći:

```
i:=0;  
ponavljaj  
    i:=i+1;  
    učitaj(niz[i]);  
sve dok nije niz[i]=0;
```

```
// nakon učitavanja na i-toj poziciji u nizu (niz se popunjavao od  
prve pozicije) se nalazi vrijednost nula. Sada je potrebno odrediti  
na kojim se pozicijama nalaze zadani zaostatci te paziti da dobivena  
vrijednost bude pozitivna.
```

```
učitaj(B1);  
učitaj(B2);
```

```
ispiši(abs(niz[i-(B1-1)]-niz[i-(B2-1)]));
```

```
// da se niz popunjavao od nulte pozicije, tada u određivanju  
zaostatka nije potrebno oduzimati jedan.
```

Potrebno znanje: naredba ponavljanja, nizovi

Kategorija: ad hoc (bez upotrebe posebnih algoritama)



Zadatak: Tenis

Tenis je jedan zadatak koji ne zahtjeva neku posebnu ideju jer se sve svodi na simuliranje uvjeta koji pišu u zadatku. Na ovom zadatku su razna nepotpuna rješenja (ali točna u tom dijelu) mogla donijeti dosta bodova. Općenito, kod ovakvih zadataka kada ne znate optimalno rješenje trebate težiti osvajanju što više bodova osmišljavanjem raznih nepotpunih rješenja.

Posebno su bile kritične situacije kada je trebalo pratiti prednost i izjednačenje kao i situacija pri ispisu kada je trebalo bodove pretvoriti u teniski rječnik (15, 30, 40).

Jedno od mogućih cjelovitih rješenje je:

```
//prvo definiramo niz koji će nam olakšati prebacivanje poena u  
tenisku notaciju. Može i bez njega, ali se onda sve komplicira.  
poen[0]:=0; poen[1]:=15; poen[2]:=30; poen[3]:=40;
```

```
učitaj(A_gem); učitaj(B_gem);  
A_poen:=0; B_poen:=0;
```

```
učitaj (n);  
za i:=1 do n radi  
{  
    učitaj(z);  
    ako je z='A' tada  
    {  
        povećaj(A_poen,1);  
  
        ako je (A_poen>=4) i (A_poen-B_poen>=2) tada  
        {  
            povećaj(A_gem,1);  
            A_poen:=0;  
            B_poen:=0;  
        };  
  
        ako je (A_poen>=4) i (A_poen-B_poen=1) tada  
        {  
            A_poen:=4;  
            B_poen:=3;  
        };  
    }  
    inače  
    {  
        // slično kao u dijelu za poena igrača „A”  
    };  
};  
  
ispiši(A_gem,':',B_gem,' ');
```



```
ako je (A_poen<=3) i (B_poen<=3) tada  
  ispiši (poen[A_poen],':',poen[B_poen])  
inače  
  ako je (A_poen=4) i (B_poen=3) tada  
    ispiši ('AD-A')  
  inače  
    ako je (A_poen=3) i (B_poen=4) tada  
      ispiši ('AD-B')  
    inače  
      ispiši ('deuce');
```

Potrebno znanje: naredba odlučivanja (if), naredba ponavljanja (for)

Kategorija: simulacija

Zadatak: Restoran

Restoran je dvostruki zadatak koji se može riješiti istom idejom. Iz omjera zadanih test primjera je očito za je dio za „subotu“ lakši od dijela za „nedjelju“.

Subota

Kreiramo niz s 24 komponente (za svaki sat u danu rezerviramo jednu). Nije nužno imati 24 komponente ali olakšava posao. Redom učitavamo račune te u nizu na poziciji koja odgovara satu u kome je izdan račun vrijednost povećamo za vrijednost koja piše na i-tom izdanom računu. Nakon toga još samo moramo pronaći minimalnu vrijednost (oprez, vrijednost mora biti veća od nule) u onom dijelu niza koji odgovara radnom vremenu (od 9. pozicije do 14. pozicije). Kako format ispis nije bitan tada je relativno lako ispisati traženi sat.

Nedjelja

U prvom dijelu, ideja za nedjelju je ista kao za subotu. Samo sada trebamo kreirati niz duljine 24*60, tj. jednu komponentu za svaku minutu. Nakon učitavanja podataka, treba pronaći minimalnu zbrojenu vrijednost svakih 15 minuta u slijedu. Kada se to pronađe, sam ispis nije više problem.

Potrebno znanje: naredba ponavljanja (for), nizovi, algoritam traženja minimuma

Kategorija: ad hoc