

PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 27 (1999/2000)

Številka 6

Strani 340-341

Martin Juvan:

LOKALNE VOLITVE

Ključne besede: računalništvo, programiranje, volilni imeniki, urejanje, programski jezik C.

Elektronska verzija: <http://www.presek.si/27/1423-Juvan.pdf>

© 2000 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

LOKALNE VOLITVE

Nedavno so se v Kašlju odločili, da krajevno skupnost razdelijo na dva dela: Spodnji Kašelj in Zgornji Kašelj. Seveda je treba po razdelitvi opraviti nove lokalne volitve. Žal pa sta volilna imenika za oba Kašlja nekoliko neurejena.

In tako so se v odboru za pripravo volitev spomnili Vas, kašelskega programerskega vseznalca. Poiskati morate napake v volilnih imenikih.

Vsaka oseba ima enolično določeno štirimestno identifikacijsko številko. Vhodni podatki so tri (neurejena) zaporedja števil. Vsako se konča z vrednostjo -1 . Prvo zaporedje predstavlja seznam volivcev, torej gre za seznam identifikacijskih števil tistih oseb, ki lahko volijo. Paziti morate, da se v tem seznamu nobeno število ne pojavi večkrat; če se pojavi, izpišite opozorilo (za vsakega volivca samo enkrat):

Volivec xxxx je prijavljen večkrat.

Drugo in tretje zaporedje predstavljata nova volilna imenika za Spodnji oziroma Zgornji Kašelj. Tu je treba preveriti več stvari. Najprej je treba ugotoviti, ali imajo osebe iz teh dveh imenikov sploh volilno pravico, ali se torej nahajajo v prvem seznamu. Če najdete osebo brez volilne pravice, izpišite obvestilo (za vsako osebo v vsakem imeniku samo enkrat):

Oseba xxxx iz Spodnjega Kašlja nima volilne pravice.

oziroma

Oseba xxxx iz Zgornjega Kašlja nima volilne pravice.

Če se taka oseba nahaja v obeh imenikih, izpišite obe obvestili. Za osebe, ki imajo volilno pravico, je treba preveriti, da niso hkrati prijavljene v obeh Kašljih. Če so, morate izpisati:

Volivec xxxx je prijavljen v Spodnjem in Zgornjem Kašlju.

Za volivce, ki so le v enem od obeh imenikov, je treba preveriti, da so vanj vpisani le enkrat. Obvestilo o odkriti napaki je podobno kot pri seznamu oseb z volilno pravico:

Volivec xxxx je v Spodnjem Kašlju prijavljen večkrat.

Če gre za volivca iz Zgornjega Kašlja, v izpisu besedico Spodnjem zamenjamo z Zgornjem. Nazadnje je treba še preveriti, da je vsak volivec

vpisan v volilni imenik za Spodnji ali za Zgornji Kašelj. Če odkrijete volivca, ki ni vpisan v nobenem Kašlju, morate izpisati:

Volivec xxxx ni vpisan v noben volilni imenik.

Za konec si oglejmo še primer. Recimo, da imamo na vhodu naslednja zaporedja števil:

0002	1234	3784	1234	8652	9399	1980	2213	0999	-1
8652	4321	8652	4321	3784	8652	2213	9399	-1	
1980	1234	3784	7777	1234	4321	7777	3784	0002	-1

Program za preverjanje pravilnosti volilnih imenikov mora izpisati naslednja obvestila:

Volivec 1234 je prijavljen večkrat.

Oseba 4321 iz Spodnjega Kašlja nima volilne pravice.

Oseba 7777 iz Zgornjega Kašlja nima volilne pravice.

Oseba 4321 iz Zgornjega Kašlja nima volilne pravice.

Volivec 3784 je vpisan v Spodnjem in Zgornjem Kašlju.

Volivec 8652 je v Spodnjem Kašlju prijavljen večkrat.

Volivec 1234 je v Zgornjem Kašlju prijavljen večkrat.

Volivec 0999 ni vpisan v noben volilni imenik

Volitve so pred vrati, zato kar hitro na delo.

Martin Juvan

LOKALNE VOLITVE – Rešitev s str. 340

Navodila za preverjanje pravilnosti volilnih imenikov, podana v besedilu naloge, so nekoliko razvlečena in kar zapletena. Podatek, ki bistveno vpliva na zasnovo rešitve, pa je omenjen že na začetku: vsaka oseba je enolično opisana s štirimestno identifikacijsko številko. Potencialnih volivcev torej ni več kot 10000.

Rešitev bomo sprogramirali v programskem jeziku C, brez težav pa jo bo moč predelati tudi v pascal in druge sorodne jezike. Zmerno število volivcev nam omogoča, da v programu seznam volivcev in oba volilna imenika predstavimo s tremi tabelami `sez`, `ime1` in `ime2`, ki bodo vse imele po 10000 elementov. Če bi bilo potencialnih volivcev več, bi s tako zastavljeno rešitvijo lahko zašli v težave zaradi pomanjkanja pomnilnika. Vrednosti elementov tabel bodo le 0, 1 ali 2. Element z indeksom `i` bo imel vrednost 0, če osebe z identifikacijsko številko `i` ne bo v seznamu volivcev oziroma v pripadajočem volilnem imeniku, vrednost 1, če se bo v seznamu oziroma imeniku pojavila natanko enkrat, in vrednost 2, če se bo pojavila večkrat. Da ne bomo preveč potratni s pomnilnikom, bodo elementi tabele tipa `char` (v C-ju se tudi znaki obnašajo kot majhna števila).

Opišimo še, kako poteka preverjanje seznama volivcev in obeh volilnih imenikov. Najprej v zanki preberemo identifikacijske številke volilnih upravičencev. Uporabili bomo zanko `do`. Seznam številke se konča z vrednostjo `-1`, zato tedaj, ko preberemo to vrednost, z `break` končamo zanko. Identifikacijsko številko preberemo v spremenljivko `oseba`. Če ima `sez[oseba]` vrednost 0, to vrednost le popravimo na 1. Če ima `sez[oseba]` vrednost 1, vrednost povečamo na 2, hkrati pa tudi izpišemo obvestilo, da je volivec večkrat prijavljen. Kadar ima `sez[oseba]` vrednost 2, ne storimo ničesar (obvestilo o večkratni prijavi smo že izpisali pri spremembi vrednosti z 1 na 2, ponovljenih obvestil pa ne smemo izpisovati). Na enak način preberemo tudi podatke o osebah iz volilnih imenikov za Spodnji in Zgornji Kašelj. Razlika je le pri preverjanju in izpisovanju opozoril. Tokrat izpišemo opozorilo, ko prvič naletimo na osebo, ki ni na seznamu volivcev (vrednost `sez[oseba]` je enaka 0, pa tudi `ime1[oseba]` oziroma `ime2[oseba]` mora imeti vrednost 0).

Ko so vsi podatki prebrani, opravimo še preostala štiri preverjanja. Vsako od njih je sestavljeno iz zanke `for`, v kateri spremenljivka `oseba` preteče vse možne identifikacijske številke. Pogoji, ki jih pri posameznem prehodu preverjamo, so:

kaj preverjamo	sez[oseba]	ime1[oseba]	ime2[oseba]
oseba, prijavljena v obeh Kašljih	!= 0	!= 0	!= 0
oseba, večkrat prijavljena v Spodnjem Kašlju	!= 0	== 2	== 0
oseba, večkrat prijavljena v Zgornjem Kašlju	!= 0	== 0	== 2
volivci, ki niso vpisani v nobenem Kašlju	!= 0	== 0	== 0

Program v C-ju je nekoliko daljši, kot so običajno rešitve v Preseku, a programersko ni posebej zahteven:

```
#include <stdio.h>

#define MAX 10000 /* zgornja meja za število volivcev */
#define NIZ1 "Oseba %04d iz %s nima volilne pravice.\n"
#define NIZ2 "Volivec %04d je %s.\n"

int main(void)
{
    char sez[MAX] = {0}, ime1[MAX] = {0}, ime2[MAX] = {0};
    int oseba;

    do { /* Pregled seznama volivcev. */
        scanf("%d", &oseba);
        if (oseba == -1) break; /* konec seznama volivcev */
        if (sez[oseba] == 0 || sez[oseba] == 1) {
            if (sez[oseba] == 1) printf(NIZ2, oseba, "prijavljen večkrat");
            sez[oseba]++;
        }
    } while (1);

    do { /* Pregled volilnega imenika za Spodnji Kašelj. */
        scanf("%d", &oseba);
        if (oseba == -1) break; /* konec imenika */
        if (ime1[oseba] == 0 || ime1[oseba] == 1) {
            if (sez[oseba] == 0 && ime1[oseba] == 0)
                printf(NIZ1, oseba, "Spodnjega Kašlja");
            ime1[oseba]++;
        }
    } while (1);
}
```

```

do { /* Pregled volilnega imenika za Zgornji Kašelj. */
scanf("%d", &oseba);
if (oseba == -1) break; /* konec imenika */
if (ime2[oseba] == 0 || ime2[oseba] == 1) {
if (sez[oseba] == 0 && ime2[oseba] == 0)
printf(NIZ1, oseba, "Zgornjega Kašlja");
ime2[oseba]++;
}
} while (1);

/* Poiščemo osebe, ki so prijavljene v Spodnjem in Zgornjem Kašlju. */
for (oseba = 0; oseba < MAX; oseba++)
if (sez[oseba] != 0 && ime1[oseba] != 0 && ime2[oseba] != 0)
printf(NIZ2, oseba, "prijavljen v Spodnjem in Zgornjem Kašlju");

/* Iščemo večkratne prijave v imenik za Spodnji Kašelj. */
for (oseba = 0; oseba < MAX; oseba++)
if (sez[oseba] != 0 && ime1[oseba] == 2 && ime2[oseba] == 0)
printf(NIZ2, oseba, "v Spodnjem Kašlju prijavljen večkrat");

/* Iščemo večkratne prijave v imenik za Zgornji Kašelj. */
for (oseba = 0; oseba < MAX; oseba++)
if (sez[oseba] != 0 && ime2[oseba] == 2 && ime1[oseba] == 0)
printf(NIZ2, oseba, "v Zgornjem Kašlju prijavljen večkrat");

/* Iščemo volivce, ki niso vpisani ne v enem ne v drugem Kašlju. */
for (oseba = 0; oseba < MAX; oseba++)
if (sez[oseba] != 0 && ime1[oseba] == 0 && ime2[oseba] == 0)
printf("Volivec %04d ni vpisan v noben volilni imenik.\n", oseba);

return 0;
}

```

Ker ima revija Presek majhne strani, smo si pri vrsticah, ki vsebujejo daljše izpise, pomagali s simboličnima konstantama NIZ1 in NIZ2 in tako malce skrajšali te vrstice.

Program bere podatke s standardnega vhoda. Če ga želite preizkusiti z večjo količino podatkov, te najprej pripravite na datoteki, nato pa prevedeni program poženite iz ukazne vrstice in vhod preusmerite na datoteko (npr. volitve <vhod.dat).

Verjetno ste tudi opazili, da program prve tri vrste obvestil izpisuje urejene tako, kot so osebe navedene v vhodnih podatkih, nadaljnje vrste obvestil pa so urejene po naraščajočih identifikacijskih številkah. Če bi, na primer, želeli doseči, da bi bila obvestila o večkratni prijavi v Spodnjem Kašlju urejena enako, kot se večkrat prijavljene osebe pojavijo v volilnem imeniku, bi morali program kar precej spremeniti.

Za konec pa še izziv za tiste, ki jim programiranja ni nikoli dovolj. V tabelah hranimo le vrednosti 0, 1 in 2. Za zapis treh vrednosti zadoščata že 2 bita. Torej lahko podatke namesto v tri tabele zložimo v eno samo. Pri posameznem elementu tabele prva bita porabimo za vrednost, ki je sedaj v tabeli `sez`, naslednja dva za vrednost iz `ime1`, še nadaljnja dva pa za vrednost iz `ime2`. Seveda je treba tudi ustrezno prilagoditi program (dodatno računanje oziroma uporaba bitnih operacij).

Martin Juvan
