

Programiranje 2 — izpit 3

23. avgust 2024

Vsa števila, ki nastopajo v besedilu in testnih primerih, so cela.

Razdelitev točk po nalogah: 30, 35, 35.

Oddajte datoteke `naloga1.c`, `naloga2.c` in `naloga3.c`.

- ① V datoteki `naloga1.h` je podana deklaracija strukture za predstavitev vozlišča povezanega seznama:

```
typedef struct Vozlisce Vozlisce;
struct Vozlisce {
    Vozlisce* naslednje; // kazalec na naslednje vozlišče v seznamu (NULL, če ga ni)
    Vozlisce* nn;        // kazalec na naslednika naslednjega vozlišča (NULL, če ga ni)
};
```

V datoteki `naloga1.c` dopolnite sledečo funkcijo:

```
void nastavi(Vozlisce* zacetek)
```

Funkcija se v vseh testnih primerih pokliče s kazalcem na začetno vozlišče nepraznega povezanega seznama z največ 1000 vozlišči. V tem seznamu je za vsako vozlišče pravilno nastavljena samo komponenta `naslednje`, komponenta `nn` pa lahko ima poljubno vrednost. Funkcijo napišite tako, da bo za vsako vozlišče pravilno nastavila komponento `nn`, komponente `naslednje` pa naj ne spremeni.

- ② Vhodna besedilna datoteka v formatu CSV vsebuje od 2 do 100 vrstic, vsaka vrstica pa vsebuje enako število ($m \in [1, 100]$) nepraznih nizov, ločenih s po enim presledkom. Vsak niz je sestavljen iz največ 100 ASCII-znakov iz množice `{'A', ..., 'Z', 'a', ..., 'z', '0', ..., '9', '_'}`. Prvi nizi v vrsticah datoteke tvorijo prvi stolpec, drugi nizi tvorijo drugi stolpec itd. V prvi vrstici (glava) so podani medsebojno različni nazivi stolpcev. Eden od stolpcev se imenuje `starost`, vsebuje pa cela števila z intervala `[0, 120]`.

Na primer, sledeča datoteka (`vhod19.csv`) vsebuje pet stolpcev, njihovi nazivi pa so `ime`, `priimek`, `starost`, `spol` in `kraj`:

```
ime priimek starost spol kraj
Janez Novak 60 M Celje
Maja Mihevc 50 Z Ljubljana
Miha Novak 30 M Ljubljana
Tina Debeljak 40 Z Kranj
Miha Breznik 70 M Ljubljana
Tina Novak 20 Z Kranj
Miha Debeljak 80 M Celje
Maja Novak 10 Z Celje
Janez Breznik 90 M Ljubljana
Janez Debeljak 100 M Kranj
```

Napišite program (`naloga2.c`), ki za vhodno datoteko s podanim imenom in za podani naziv stolpca ($C \neq \text{starost}$) izdela izhodno datoteko s podanim imenom, ki vsebuje dva stolpca, ločena z vejico. V prvem stolpcu naj bodo zapisani vsi medsebojno različni nizi,

ki nastopajo v stolpcu C . Nizi naj bodo zapisani v zaporedju njihovih prvih pojavitev. V drugem stolpcu izhodne datoteke naj bodo zapisana navzdol zaokrožena povprečja pripadajočih starosti: če je v prvem stolpcu neke vrstice zapisan niz S , naj bo v drugem stolpcu tiste vrstice zapisano navzdol zaokroženo povprečje vrednosti stolpca **starost** v tistih vrsticah vhodne datoteke, v katerih se v stolpcu C nahaja niz S . Izhodna datoteka naj bo brez glave.

Ime vhodne datoteke, ime izhodne datoteke in naziv C so (v tem vrstnem redu) podani kot argumenti ukazne vrstice pri zagonu programa.

Če bi v obravnavanem primeru program prevedli in pognali kot

```
gcc -o naloga2 naloga2.c
./naloga2 vhod19.csv rezultat19.csv kraj
```

bi program moral izdelati datoteko **rezultat19.csv** s sledečo vsebino:

```
Celje,50
Ljubljana,60
Kranj,53
```

V skritih testnih primerih 1–30 (60%) imamo samo dva stolpca, pri čemer je prvi vedno stolpec C , drugi pa **starost**. V skritih testnih primerih 1–15 in 31–45 (60%) se noben niz v stolpcu C ne ponovi (če izvzamemo glavo, bo izhodna datoteka v teh primerih torej imela natanko toliko vrstic kot vhodna).

Namig: morda vam bo koristila funkcija **atoi**.

- ③ Na šahovnici velikosti $n \times n$ ($n \in [1, 20]$) so na poljih $(v_1, s_1), (v_2, s_2), \dots, (v_k, s_k)$ ($k \in [\max(0, n^2 - 20), n^2]$; $v_i, s_i \in [0, n - 1]$ za $i = 1, \dots, k$) postavljene ovire. Napišite program (**naloga3.c**), ki izpiše največje število skakačev, ki jih je mogoče sočasno postaviti na polja brez ovir, ne da bi se katerikoli par skakačev med seboj napadal. Skakača na poljih (v, s) in (v', s') se napadata natanko v primeru, ko velja $|v' - v| \cdot |s' - s| = 2$.

V prvi vrstici standardnega vhoda je podano število n , v drugi k , nato pa sledi k vrstic, od katerih i -ta (za $i = 1, \dots, k$) vsebuje števili v_i in s_i .

Oglejmo si primer:

test01.in: test01.out:

4
5
0 2
1 3
2 1
2 2
3 1

8

			
			
			
			