

Programiranje 2 — drugi kolokvij

27. maj 2024

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

Razdelitev točk po nalogah: 20, 40, 40.

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

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

```
typedef struct Vozlisce Vozlisce;
struct Vozlisce {
    Vozlisce* naslednje; // kazalec na naslednje vozlišče v seznamu (NULL, če ga ni)
    Vozlisce* prejsnje;  // kazalec na prejšnje vozlišče v seznamu (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 `prejsnje` pa lahko ima poljubno vrednost. Funkcijo napišite tako, da bo za vsako vozlišče pravilno nastavila komponento `prejsnje`, komponente `naslednje` pa naj ne spremeni.

- ② Vhodna besedilna datoteka v formatu CSV vsebuje od 1 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 so podani medsebojno različni nazivi stolpcev.

Na primer, sledeča datoteka (`vhod01.csv`) vsebuje pet stolpcev ($m = 5$), 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
```

Napišite program (`naloga2.c`), ki za vhodno datoteko s podanim imenom izdela izhodno datoteko s podanim imenom in vanjo prepíše stolpce s podanimi nazivi. Ločilo med stolpci naj bo vejica.

Ime vhodne datoteke, ime izhodne datoteke, število m in zaporedje nazivov stolpcev so (v tem vrstnem redu) podani kot argumenti ukazne vrstice pri zagonu programa. Nazivi stolpcev, ki naj jih program prepíše, so veljavni, njihovo število pa znaša med 1 in m .

Če bi v obravnavanem primeru program prevedli in pognali kot

```
gcc -o naloga2 naloga2.c
./naloga2 vhod01.csv rezultat01.csv 5 kraj starost priimek
```

bi moral program izdelati datoteko `rezultat01.csv` s sledečo vsebino:

```
kraj,starost,priimek
Celje,60,Novak
Ljubljana,50,Mihevc
Ljubljana,30,Novak
Kranj,40,Debeljak
Ljubljana,70,Breznik
```

Namig: morda vam bo koristila funkcija `atoi`.

- ③ Na šahovnici velikosti $n \times n$ ($n \in [1, 20]$) so na poljih $(v_1, s_1), (v_2, s_2), \dots, (v_k, s_k)$ ($v_i, s_i \in [0, n-1]$ za $i = 1, \dots, k$) postavljene ovire. Polje $(0, 0)$ je vedno prazno, skupno število praznih polj (tj. polj brez ovir) pa ne presega 25. Napišite program (`naloga3.c`), ki izpiše najdaljšo pot, ki jo lahko na tej šahovnici opravi skakač, če prične na polju $(0, 0)$ in če na nobeno polje ne sme stopiti več kot enkrat.

Spomnimo se, da lahko skakač s polja (v, s) skoči na vsa polja (v', s') , za katera velja $|v' - v| \cdot |s' - s| = 2$. Na polje z oviro seveda ne more skočiti.

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 . Na standardni izhod izpišite zaporedje polj, ki tvorijo najdaljšo skakačevo pot (začetnega polja, torej polja $(0, 0)$, ne izpišite). Vsako polje izpišite v svojo vrstico, in sicer v obliki v/s , kjer je v indeks vrstice, s pa indeks stolpca. Ni pomembno, katero od najdaljših poti izpišete, če jih je morda več.

V sledečem primeru lahko skakač opravi pot dolžine 7, prikazano na desni sliki.

test01.in: test01.out:

5
10
0 2
1 3
1 4
2 1
2 2
2 4
3 1
4 1
4 2
4 4

1/2
0/4
2/3
1/1
3/2
2/0
0/1

	0	1	2	3	4
0	0	7	●		2
1		4	1	●	●
2	6	●	●	3	●
3		●	5		
4		●	●		●

Namig: bi vam lahko pomagali sledeči definiciji?

```
int DVR[] = {1, 2, 2, 1, -1, -2, -2, -1};    // vrstični premiki
int DST[] = {2, 1, -1, -2, -2, -1, 1, 2};    // stolpčni premiki
```