

Izpit pri predmetu Programiranje 2

17. junij 2019

Dopolnite in oddajte datoteke `naloga1.c`, `naloga2.c` in `naloga3.c`.

- ① (30 točk) V prvi vrstici vhoda je zapisano ime vhodne (dvojiške) datoteke, v drugi je podano celo število $n \in [1, 10^3]$, v tretji pa je navedeno ime izhodne (besedilne) datoteke. Imeni sta sestavljeni iz največ 20 znakov z ASCII-kodami od 33 do 126.

Vhodna datoteka podaja opis slike z n pikami in vsebuje $3n$ bajtov. Prvi trije bajti po vrsti podajajo komponente R , G in B za prvo piko, naslednji trije podajajo komponente R , G in B za drugo piko itd. Napišite program, ki v podano izhodno datoteko izpiše tri vrstice. V prvo naj izpiše število pik z vrednostmi $R > 0$, $G = 0$ in $B = 0$, v drugo število pik z vrednostmi $R = 0$, $G > 0$ in $B = 0$, v tretjo pa število pik z vrednostmi $R = 0$, $G = 0$ in $B > 0$.

V testnih primerih J1–J5 (S1–S25) za vse pike velja $G = B = 0$.

Testni primer J6:

test06.in:

```
vhod06.bin
7
rezultat06.txt
```

vhod06.bin:

```
Zaporedje bajtov:
0, 95, 0,
0, 0, 0,
66, 10, 0,
0, 0, 60,
0, 37, 0,
0, 205, 208,
0, 5, 0
```

izhod06.txt:

```
0
3
1
```

- ② (30 točk) V datoteki `naloga2.h` sta podani sledeči deklaraciji:

```
typedef struct Oseba Oseba;
struct Oseba {    // oseba s podanim imenom in starostjo
    char* ime;    // zaporedje do 20 črk angleške abecede
    int starost;
};
```

Napišite funkcijo `void uredi(Oseba** osebe, int n)`, ki tabelo $n \in \{1, 1000\}$ kazalcev na strukture tipa `Oseba` leksikografsko uredi po imenih oseb, pri čemer ne sme spremeniti medsebojnega vrstnega reda oseb z enakim imenom. Kazalec osebe kaže na začetek tabele.

V okvirčku na desni je prikazana začetna in končna razporeditev struktur (oz. kazalcev nanje) v tabeli, izdelani v testnem primeru J9.

V testnih primerih J1–J8 in S1–S40 so imena vseh oseb med seboj različna. V primerih J1–J4 in S1–S20 se prve črke vseh imen med seboj razlikujejo.

Začetna razp.:	Končna razp.:
{"Bojan", 30}	{"Ana", 25}
{"Ana", 25}	{"Ana", 20}
{"Bojan", 40}	{"Bojan", 30}
{"Cene", 15}	{"Bojan", 40}
{"Bojan", 20}	{"Bojan", 20}
{"Ana", 20}	{"Cene", 15}

- ③ (40 točk) Napišite program, ki prebere števili $m \in [1, 30]$ in $n \in [1, 30]$ ter opis labirinta, sestavljenega iz $m \times n$ celic (m vrstic, n stolpcev), in izpiše pot od zgornje leve do spodnje desne celice. V vseh testnih primerih obstaja natanko ena pot.

V prvi vrstici vhoda sta zapisani s presledkom ločeni števili m in n , nato pa sledi m vrstic, od katerih vsaka vsebuje n s presledkom ločenih zaporedij štirih števk 1 in 0. Zaporedja opisujejo posamezne celice labirinta: prva števka zaporedja podaja prisotnost leve stene celice (1: prisotna; 0: odsotna), druga prisotnost zgornje stene, tretja prisotnost desne, četrta pa prisotnost spodnje stene celice. Opisi sosednjih celic so med seboj usklajeni. Labirint je v celoti obdan s stenami.

Pot izpišite kot zaporedje števk 0 (levo), 1 (navzgor), 2 (desno) in 3 (navzdol) brez vmesnih presledkov.

V testnih primerih J1–J3 in S1–S15 so stene prisotne povsod razen vzdolž edine poti.

Slike labirintov (v besedilni obliki) si lahko ogledate v datotekah `test01.lab`, ..., `test10.lab`.

Testni primer J4:

`test04.in:`

```
4 3
1101 0110 1110
1110 1001 0010
1000 0101 0011
1001 0101 0111
```

`test04.out:`

```
232300322
```

`test04.lab:`

```
+---+---+---+
|           | |
+---+   +   +
|   |           |
+   +---+   +
|           |
+   +---+---+
|           |
+---+---+---+
```