

Programiranje 2 — prvi kolokvij (skupina A)

29. maj 2023

Oddajte datoteke `naloga1.c`, `naloga2.c` in `naloga3.c`. Točkovanje: 30, 30, 40.

- ① Napišite program, ki podano vhodno datoteko dobesečno prepíše v podano izhodno datoteko, le da vsako pojavitev dvoznakovnega zaporedja "C", "c", "S", "s", "Z" oziroma "z" zamenja z znakom C, c, S, s, Z oziroma z. Ime vhodne datoteke je podano kot prvi, ime izhodne datoteke pa kot drugi argument ukazne vrstice. Vhodna datoteka vsebuje kvečjemu 10^5 znakov.

V sledečem primeru program pokličemo z argumentoma `vhod01.txt` in `izhod01.txt`:

`vhod01.txt`:

```
"Ze kot pobi" c sem se navdu"seval  
nad ""S"cuke pa ni, "s"cuca pa ne."
```

`izhod01.txt`:

```
Ze kot pobic sem se navduseval  
nad "Scuke pa ni, scuka pa ne."
```

- ② V datoteki `naloga2.h` je podana sledeča deklaracija:

```
typedef struct _Vozlisce Vozlisce;  
struct _Vozlisce {  
    int podatek;  
    Vozlisce* naslednje;  
};
```

V datoteki `naloga2.c` dopolnite funkcijo

```
void izlociSkupne(Vozlisce* a, Vozlisce* b, Vozlisce** na, Vozlisce** nb),
```

ki iz podanih dveh nepraznih povezanih seznamov izloči vozlišča s podatki, ki nastopajo v obeh seznamih. Kazalec `a` kaže na prvo vozlišče prvega seznama, kazalec `b` pa na prvo vozlišče drugega seznama. V nobenem od seznamov se ne zgodi, da bi se isti podatek pojavil več kot enkrat. Funkcija naj v spremenljivko, na katero kaže kazalec `na`, vpiše kazalec na (morebitni novi) začetek prvega seznama, v spremenljivko, na katero kaže kazalec `nb`, pa kazalec na (morebitni novi) začetek drugega seznama.

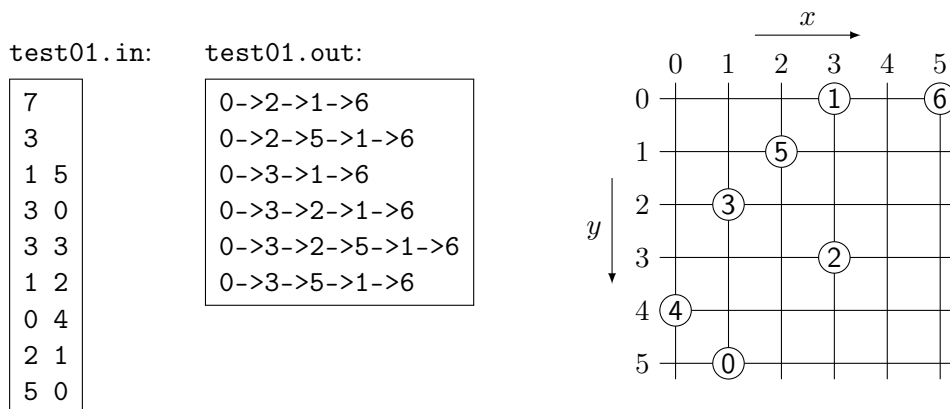
Testni program `test01.c` zgradi seznama [3, 7, 2, 9, 8, 6, 5] (prvo vozlišče ima podatek 3, drugo 7 itd.) in [4, 1, 5, 2, 3, 0]. Po klicu funkcije `izlociSkupne` mora od prvega seznama ostati [7, 9, 8, 6], od drugega pa [4, 1, 0].

Funkcija mora ustrezna vozlišča dejansko *izločiti*, ostala pa *ohraniti*; »alternative«, ki temeljijo na (denimo) izdelavi novega seznama ali prestavljanju podatkov med vozlišči, ne bodo prinesle nobene točke.

Število vozlišč v nobenem seznamu ne presega 1000.

- ③ Stojimo na kamnu na koordinatah (x_0, y_0) , radi pa bi prek neke podmnožice množice kamnov na koordinatah $(x_1, y_1), (x_2, y_2), \dots, (x_{n-2}, y_{n-2})$ prispeli do kamna na koordinatah (x_{n-1}, y_{n-1}) . V poštev pridejo le poti, v katerih noben korak ni daljši od vrednosti K , poleg tega pa bi se radi ciljnemu kamnu vseskozi približevali: po vsakem koraku bi radi bili od ciljnega kamna oddaljeni strogo manj kot prej. Napišite program, ki na standardni izhod izpiše vse poti, ki ustrezajo navedenim pogojem. Poti lahko izpišete v poljubnem vrstnem redu.

V prvi vrstici standardnega vhoda je podano število $n \in [2, 10]$, v drugi pa $K \in [1, 10^4]$. V naslednjih n vrsticah so podane koordinate (x_i, y_i) (za $i \in \{0, \dots, n-1\}$); vsaka koordinata pripada intervalu $[-10^3, 10^3]$. Vsa vhodna števila so cela. Sledi primer:



Pot $0 \rightarrow 3 \rightarrow 5 \rightarrow 6$, denimo, je neustrezna zaradi predolgega koraka (razdalja med kamnoma 5 in 6 znaša $\sqrt{10} > 3$), pot $0 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow 6$ pa zaradi kršitve zahteve po nenehnem približevanju (kamen 4 ni kamnu 6 nič bližje kot kamen 0).

Za delne točke:

- V 50% testnih primerov velja $y_i = 0$ za vsak $i \in \{0, \dots, n-1\}$, koordinate x pa pri potovanju od kamna 0 do kamna $n-1$ strogo naraščajo.