

Programiranje 2 — prvi kolokvij (skupina B)

29. maj 2023

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

- ① Napišite program, ki podano vhodno datoteko dobesedno prepiše v podano izhodno datoteko, le da v vsaki vrstici pobriše vse njene morebitne začetne presledke. (To so res samo presledki; datoteka ne vsebuje tabulatorjev.) 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` (presledki so prikazani z znaki `␣`):

`vhod01.txt`:

```
␣␣␣0␣Vrba!
srecna,␣␣␣draga␣␣␣␣
␣␣␣␣␣␣␣␣␣
␣␣␣␣␣␣␣vas␣␣domaca!␣
```

`izhod01.txt`:

```
0␣Vrba!
srecna,␣␣␣draga␣␣␣␣
vas␣␣domaca!␣
```

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

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

V datoteki `naloga2.c` dopolnite funkcijo

```
void izlociDuplikate(Vozlisce* zacetek),
```

ki podani povezani seznam spremeni tako, da v vsaki množici vozlišč, pri katerih kazalci `p` kažejo na isto spremenljivko, ohrani samo prvo vozlišče, vsa ostala pa izloči. Kazalec `zacetek` kaže na prvo vozlišče izhodiščnega seznama.

Testni program `test01.c` zgradi seznam s 6 vozlišči, pri čemer

- kazalca `p` v vozliščih z indeksoma 0 in 2 kažeta na isto spremenljivko;
- kazalci `p` v vozliščih z indeksi 1, 3 in 5 kažejo na isto spremenljivko.

V tem primeru mora funkcija ohraniti samo vozlišča z indeksi 0, 1 in 4, vsa ostala pa mora izločiti.

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 seznama v nobenem testnem primeru ne presega 1000.

- ③ Podan je usmerjen graf z n vozlišči, označenimi z $0, 1, \dots, n-1$, in m uteženimi povezavami. Graf nima niti zank niti vzporednih povezav. Napišite program, ki na standardni izhod izpiše vse aciklične poti od vozlišča 0 do vozlišča $n-1$, pri katerih vsota uteži ne presega vrednosti K . (Pot je aciklična, če se v njej nobeno vozlišče ne ponovi.) Poti lahko izpišete v poljubnem vrstnem redu.

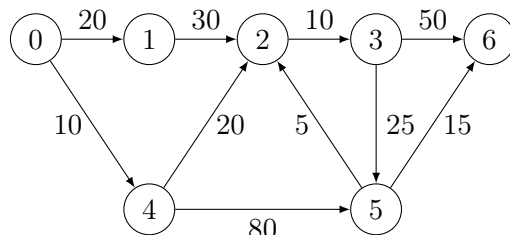
V prvi vrstici standardnega vhoda sta podani števili $n \in [1, 10]$ in $m \in [n-1, n(n-1)]$, v drugi pa število $K \in [1, 10^4]$. Naslednjih m vrstic podaja posamezne povezave; vsaka je opredeljena z izvirnim vozliščem, ciljnim vozliščem in utežjo z intervala $[1, 10^3]$. Vsa vhodna števila so cela. Sledi primer:

test01.in:

```
7 10
100
0 1 20
0 4 10
1 2 30
2 3 10
3 5 25
3 6 50
4 2 20
4 5 80
5 2 5
5 6 15
```

test01.out:

```
0->1->2->3->5->6
0->4->2->3->5->6
0->4->2->3->6
```



Za delne točke:

- V 50% testnih primerov je graf acikličen.