

Programiranje 2 — drugi kolokvij

22. maj 2019

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

- ① (15 točk) Vsaka vrstica podane vhodne datoteke je sestavljena iz besed, ločenih s presledkom, pri čemer je vsaka beseda neprazno zaporedje znakov z ASCII-kodami od 33 do 126. Datoteka vsebuje od 1 do 1000 vrstic, vsaka od njih pa vsebuje od 0 do 1000 znakov (brez upoštevanja znaka `\n`) in se niti ne prične niti ne konča s presledkom.

Napišite program, ki ugotovi, koliko besed vsebuje vrstica z največ besedami, in rezultat zapiše v podano izhodno datoteko.

V prvi vrstici standardnega vhoda je podano ime vhodne, v drugi pa ime izhodne datoteke. Imeni vsebujeta od 1 do 100 znakov iz množice `{A-Z, a-z, 0-9, ., -, _}`.

V testnih primerih J1–J5 (S1–S25) so vse besede sestavljene iz enega samega znaka.

Namig: datoteko lahko berete s funkcijo `fgets`, `getline` ali celo `fgetc`.

Testni primer J6:

test06.in:

```
vhod06.txt
rezultat06.txt
```

vhod06.txt:

```
java C C++
python
scala haskell lisp
SQL PHP perl ruby
swift go
```

rezultat06.txt:

```
4
```

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

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

Napišite funkcijo `void izloci(Vozlisce* zacetek)`, ki iz povezanega seznama, na čigar začetno vozlišče kaže kazalec `zacetek`, po vrsti izloči sledeča vozlišča:

- vozlišče, ki je za k_1 korakov oddaljeno od začetnega vozlišča, pri čemer je k_1 komponenta `podatek` v začetnem vozlišču;
- vozlišče, ki je za k_2 korakov oddaljeno od naslednika pravkar izločenega vozlišča, pri čemer je k_2 komponenta `podatek` v omenjenem nasledniku;
- vozlišče, ki je za k_3 korakov oddaljeno od naslednika pravkar izločenega vozlišča, pri čemer je k_3 komponenta `podatek` v omenjenem nasledniku;
- ...

Funkcija naj se zaključi, ko vozlišče, ki bi ga bilo treba izločiti, ne obstaja. Vse komponente `podatek` so pozitivna števila.

Testni primeri J1–J3 (S1–S15) so pripravljeni tako, da bo funkcija izločila natanko eno vozlišče.

Sledeča slika se nanaša na testni primer J4. Vrednosti k_1, k_2, \dots so obkrožene, vozlišča, ki jih je treba izločiti, pa prekržana:



- ③ (20 točk) Napišite program, ki izpiše vse aciklične poti med podanima vozliščema v podanem usmerjenem grafu. Vozlišča v grafu so označena z indeksi $0, 1, \dots, n-1$, kjer je n število vozlišč.

V prvi vrstici vhoda je zapisano število $n \in [2, 10]$, naslednjih n vrstic pa določa matriko sosednosti; vsaka od njih vsebuje po n enic in ničel, ločenih s presledkom. Povezava od vozlišča i do vozlišča j obstaja natanko tedaj, ko je element matrike v vrstici z indeksom i in stolpcu z indeksom j enak 1. V zadnji vrstici vhoda sta zapisani s presledkom ločeni števili $u \in [0, n-1]$ in $v \in [0, n-1] \setminus \{u\}$, ki določata začetno in končno vozlišče poti.

Vsako pot izpišite v svoji vrstici, in sicer kot zaporedje indeksov vozlišč brez kakršnihkoli ločil. Poti naj bodo izpisane v leksikografskem vrstnem redu; na primer, pot 327 naj se izpiše pred potjo 3407 ($3 = 3, 2 < 4$), ta pa pred potjo 347 ($3 = 3, 4 = 4, 0 < 7$).

V testnih primerih J1–J7 (S1–S35) je graf acikličen. V primerih J1–J5 (S1–S25) obstaja med začetnim in končnim vozliščem natanko ena pot. V primerih J1–J3 (S1–S15) ima vsako vozlišče kvečjemu eno izhodno povezavo.

Testni primer J8 (vhod/izhod):

```

6
0 1 0 0 1 0
1 0 0 1 0 0
1 0 0 0 0 0
0 0 0 0 0 0
0 0 1 0 0 1
0 1 0 1 0 1
2 3

```

```

2013
204513
20453

```

Sledeča slika prikazuje graf za ta testni primer:

