

Programiranje 2 — drugi kolokvij

28. maj 2020

Dopolnite in oddajte datoteke `naloga1.c`, `naloga2.c` in `naloga3.c`.
Naloge so enakovredne.

① Napišite program

`naloga1` datoteka n k

ki na standardni izhod izpiše k -to besedo v n -ti vrstici besedilne datoteke `datoteka`. Besede znotraj vrstice so med seboj ločene s po enim presledkom. Datoteka vsebuje kvečjemu 10^4 znakov; vsi pripadajo osnovnemu naboru ASCII.

Velja $n \in [1, N]$ in $k \in [1, K_n]$, kjer je N število vrstic datoteke, K_n pa število besed v n -ti vrstici. V 50% testnih primerov velja $n = 1$.

Javni testni primer 6:

V tem primeru se program pokliče takole:

```
./naloga1 vrba.txt 2 3
```

Datoteka `vrba.txt`:

```
0 Vrba
srecna draga vas domaca
kjer hisa mojega stoji oceta
```

Pričakovani izpis:

```
vas
```

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

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

V datoteki `naloga2.c` napišite sledečo funkcijo:

```
Vozlisce* prepleti(Vozlisce** seznam, int n)
```

Kazalec `seznam` kaže na začetek tabele n kazalcev, od katerih vsak kaže na prvo vozlišče nekega nepraznega povezanega seznama. Naj V_{ij} označuje j -to vozlišče i -tega seznama. Funkcija naj vrne kazalec na prvo vozlišče novega povezanega seznama, v katerem si po vrsti sledijo *kopije* (pozor!) vozlišč $V_{11}, V_{21}, \dots, V_{n1}, V_{12}$ (če obstaja), V_{22} (če obstaja), \dots, V_{n2} (če obstaja), V_{13} (če obstaja), V_{23} (če obstaja), \dots, V_{n3} (če obstaja) itd.

Skupno število elementov v vhodnih seznamih ne presega 10^6 .

V 60% testnih primerov so vsi seznam enako dolgi. V polovici od teh primerov velja $n = 2$.

Javni testni primer 7:

V tem primeru se funkcija pokliče s seznam $\langle 5, 8, 3 \rangle, \langle 7 \rangle, \langle 4, 6, 1, 2 \rangle$ in $\langle 0, 9 \rangle$ ($n = 4$), vrne pa naj kazalec na prvo vozlišče seznama $\langle 5, 7, 4, 0, 8, 6, 9, 3, 1, 2 \rangle$.

- ③ Napišite program, ki s standardnega vhoda prebere podatek in opis drevesa in na standardni izhod izpiše pot od korena do vozlišča z iskanim podatkom. Drevo vsebuje natanko eno vozlišče z iskanim podatkom, skupno število vozlišč drevesa pa ne presega 1000.

Prva vrstica vhoda vsebuje podatek (celo število z intervala $[-10^9, 10^9]$), druga pa opis drevesa v obliki

$koren\ n\ poddrevo_1\ poddrevo_2\ \dots\ poddrevo_n$

pri čemer je *koren* podatek v korenu drevesa (celo število z intervala $[-10^9, 10^9]$), n število otrok korena, $poddrevo_1, \dots, poddrevo_n$ pa opisi posameznih poddreves korena. Ti opisi so zgrajeni na enak način kot opis celotnega drevesa.

Pot izpišite tako, da po vrsti navedete podatke v vozliščih od vključno korena do vključno vozlišča z iskanim podatkom. Vsak podatek izpišite v svoji vrstici.

V 50% testnih primerov ima vsako vozlišče bodisi dva otroka bodisi 0 otrok.

Javni testni primer 6 (vhod/izhod):

```
80
50 2 30 3 10 0 90 0 80 1 60 0 40 1 20 1 70 0
```

```
50
30
80
```

Sledeča slika prikazuje drevo, ki mu pripada opis na vhodu:

