

Programiranje 2 — izpit 1

11. junij 2025

Vsa števila, ki nastopajo v besedilu in testnih primerih, so cela.

Razdelitev točk po nalogah: 20, 20, 30, 30.

Oddajte datoteke `naloga1.c`, `naloga2.c`, `naloga3.c` in `naloga4.c`.

- ① V tej nalogi bomo vaš program poganjali z zaporedjem argumentov, ki bo sestavljeno iz stikal z morebitnimi pripadajočimi vrednostmi. Vsako stikalo bo zavzemalo eno od sledečih oblik:

- `--stikalo`

Za vsako stikalo takšne oblike naj program na standardni izhod izpiše vrstico `stikalo True`.

- `-stikalo vrednost`

Za vsako stikalo takšne oblike naj program na standardni izhod izpiše vrstico `stikalo vrednost`.

Na primer, v prvem testnem primeru program pokličemo takole:

```
./naloga1 -velikost 20 --srafiraj -polnilo rdeca
```

Vaš program naj na standardni izhod izpiše sledeče:

```
velikost 20
srafiraj True
polnilo rdeca
```

Tako imena kot vrednosti stikal so zaporedja najmanj ene in največ 20 črk in števč. Število stikal znaša med 1 in 100. Imena stikal se ne podvajajo.

V 30% testnih primerov so vsa stikala prve oblike. V nadaljnjih 30% testnih primerov so vsa stikala druge oblike.

- ② Elemente dvojiške matrice s $h \in [1, 1000]$ vrsticami in $w \in [1, 1000]$ stolpci, pri čemer je zmnožek hw deljiv z 8, smo po vrsticah prepisali v dvojiško datoteko. Vsakemu elementu ustreza en bit.

Napišite program, ki s standardnega vhoda prebere ime dvojiške datoteke, števila h , w in $n \in [1, 100]$ (vse to je zapisano v prvi vrstici) ter n parov vrstičnega in stolpčnega indeksa (vsak je zapisan v svoji vrstici), na standardni izhod pa izpiše elemente matrice na podanih indeksih. Indeks je število med 0 in $h - 1$ oziroma $w - 1$. Ime datoteke je zaporedje najmanj ene in največ 20 črk, števč, pik in podčrtajev.

Sledi vsebina standardnega vhoda, pričakovana vsebina standardnega izhoda, vsebina vhodne datoteke (po bajtih) in pripadajoča matrika za prvi testni primer.

test01.in:

```
matrika01.bin 4 6 3
3 5
0 4
2 1
```

test01.out:

```
0
0
1
```

matrika01.bin:

```
01100010
10110101
10111000
```

Matrika:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & \underline{0} & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & \underline{1} & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & \underline{0} \end{bmatrix}$$

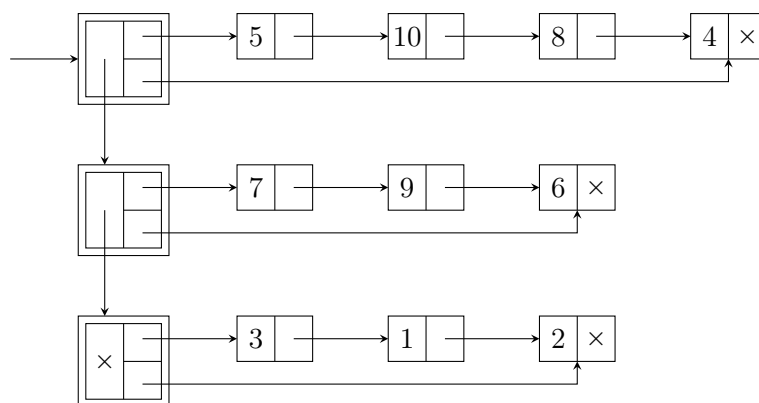
V 30% testnih primerov velja $h = 1$.

- ③ Dvonivojska konstrukcija je sestavljena iz zunanega povezanega seznama in množice notranjih povezanih seznamov. Vsako vozlišče zunanega seznama vsebuje kazalec na začetek in kazalec na konec notranjega seznama:

```
struct Zunanje {      // vozlišče zunanjega seznama
    Notranje* prvo;   // prvo vozlišče pripadajočega notranjega seznama
    Notranje* zadnje; // zadnje vozlišče pripadajočega notranjega seznama
    Zunanje* dol;     // naslednje vozlišče zunanjega seznama (NULL, če ga ni)
};

struct Notranje {     // vozlišče notranjega seznama
    int podatek;       // podatek, ki ga hrani vozlišče (element seznama)
    Notranje* desno;   // naslednje vozlišče (NULL, če ga ni)
};
```

Sledeča slika prikazuje primer dvonivojske konstrukcije. Vozlišča zunanega seznama (tj. vozlišča tipa `Zunanje`) so dvojno uokvirjena, znak \times pa predstavlja vrednost NULL.



V datoteki `naloga3.c` dopolnite funkcijo

```
Zunanje* porazdeli(Notranje* zacetek, int k),
```

ki iz *obstojećih* (!) vozlišč podanega nepraznega povezanega seznama zgradi dvonivojsko konstrukcijo s k notranjimi seznamami. Prvo vozlišče naj doda v prvi notranji seznam, drugo v drugi, ..., k -to v k -ti, naslednje spet v prvi itd. Funkcija naj vrne kazalec na prvo vozlišče zunanega seznama. Gornjo konstrukcijo dobimo s seznamom 5, 7, 3, 10, 9, 1, 8, 6, 2, 4 in vrednostjo $k = 3$ (`test01.c`).

V vseh testnih primerih velja $k \in [1, 1000]$ in $n \in [k, 1000]$ (n je dolžina vhodnega seznama). V 40% testnih primerov bomo sprejeli tudi rešitev, ki dvonivojsko konstrukcijo zgradi iz kopij vozlišč vhodnega seznama.

- ④ Napišite program, ki s standardnega vhoda prebere števili $m \in [0, 100]$ in $n \in [0, 100 - m]$, na standardni izhod pa izpiše, na koliko načinov lahko m belih in n črnih kock razporedimo v vrsto, tako da nikjer ne bosta zaporedoma nastopali več kot dve beli kocki in da nikjer ne bodo zaporedoma nastopale več kot tri črne kocke. Primer:

test01.in: test01.out:

3 4

28

Med $\binom{7}{3} = 35$ možnimi razporeditvami kock je sledečih 7 nedopustnih: BBBČČČČ, BBČČČČB, BČČČČBB, ČČČČBBB, ČČČBBBČ, ČČBBBČČ in ČBBBČČČ.

V 40% testnih primerov velja $m + n \leq 20$, v nadaljnjih 30% pa $m + n \leq 30$. V vseh testnih primerih je rezultat manjši od 2^{63} .