

## Izpit pri predmetu Programiranje 2 — 27. junij 2018

Vse naloge so enakovredne.

Rešitve posameznih nalog oddajte v datotekah `naloga1.c`, `naloga2.c` in `naloga3.c`.

- ① Napišite program, ki prebere število  $n$  in zaporedje  $n$  znakov  $+$  in  $-$  in izpiše navzdol zaokroženo povprečje razdalj med zaporednima znakoma  $+$ .

### Vhod:

Vhod je sestavljen iz ene same vrstice, ta pa vsebuje celo število  $n \in [1, 1000]$ , presledek in zaporedje  $n$  znakov  $+$  in  $-$ . Zaporedje vsebuje najmanj dva znaka  $+$ .

### Izhod:

Izpišite iskano povprečje.

### Testni primer J1 (vhod/izhod):

28	-----+--+-----+-----+---
3	

Razdalja med prvim in drugim znakom  $+$  znaša 3, med drugim in tretjim 6, med tretjim in četrtem 5, med četrtem in petim pa 1; iskano povprečje je potemtakem  $\lfloor 15 / 4 \rfloor = 3$ .

- ② Podana je sledeča deklaracija:

```
typedef struct _Vozlisce { // vozlišče povezanega seznama
    int podatek;           // podatek v vozlišču
    struct _Vozlisce* n;   // kazalec na naslednika (NULL, če ga ni)
    struct _Vozlisce* nn;  // kazalec na naslednika naslednika (NULL, če ga ni)
} Vozlisce;
```

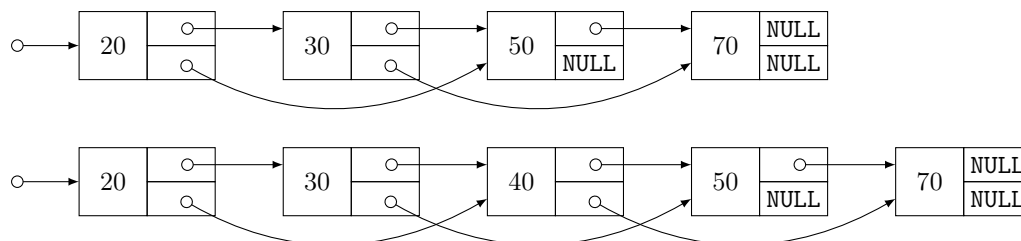
Napišite funkcijo

`Vozlisce* vstaviUrejeno(Vozlisce* zacetek, int element)`

ki ustvari vozlišče s podatkom `element` in ga vstavi na ustrezno mesto v naraščajoče urejeni seznam, čigar prvo vozlišče se nahaja na naslovu `zacetek`. (Seznam je lahko tudi prazen; v tem primeru ima kazalec `zacetek` vrednost `NULL`.) Funkcija mora seveda posodobiti komponente `n` in `nn`, vrne pa naj kazalec na začetek obogatene seznama.

V testnih primerih J1–J6 in S1–S30 vsebuje seznam najmanj 4 vozlišča, `element`, ki ga vstavljamo, pa je večji od drugega in manjši od predzadnjega elementa seznama. V primerih J1–J3 in S1–S15 vsebuje seznam natanko 4 vozlišča.

Sledeča slika prikazuje začetni in končni seznam v testnem primeru J1:



- ③ Parkirnina stane 1 evro, parkomat pa sprejema kovance za 1 evro in 2 evra. Če stranka vstavi kovanec za 2 evra, parkomat pa ne premore nobenega kovanca za 1 evro, nastanejo težave. Če parkomat na začetku vsebuje, denimo, dva kovanca za 1 evro, se to zgodi pri zaporedju vstavljanj 22122: ko zadnja stranka vstavi kovanec za 2 evra, ji parkomat ostanka ne more vrniti.

Napišite program, ki prebere števili  $n$  in  $k$  in izpiše število vseh zaporedij vstavljanja  $n$  kovancev, pri katerih **ne** pride do težav, če je v parkomatu na začetku  $k$  kovancev po 1 evro.

**Vhod:**

Na vhodu sta podani celi števili  $n \in [1, 20]$  in  $k \in [0, n]$ , ločeni s presledkom. V testnih primerih J1–J3 in S1–S15 velja  $n \leq 4$ .

**Izhod:**

Izpišite iskano število zaporedij.

**Testni primer J4 (vhod/izhod):**

---

6 2

---

50

---

Parkomat »preživi« vsa zaporedja razen 12222x, 21222x, 22122x in 222xxx ('x' lahko predstavlja enico ali dvojko).