

Programiranje 2 — prvi izpitni rok

16. junij 2022

- ① V datoteki `naloga1.h` sta podani sledeči deklaraciji:

```
#define MAXFON 31

typedef struct {
    char* ime;
    char telefon[MAXFON];    // telefonska številka
} Oseba;
```

V datoteki `naloga1.c` dopolnite sledečo funkcijo:

```
void vstaviPredpono(char* predpona, Oseba** osebe, int stOseb)
```

Kazalec `predpona` kaže na začetek niza, ki podaja klicno predpono države (npr. +386 za Slovenijo), kazalec `osebe` pa na začetek tabele `stOseb` ($1 \leq \text{stOseb} \leq 10^4$) kazalcev na strukture tipa `Oseba`, ki predstavljajo posamezne osebe. Funkcija naj na začetek telefonske številke vsake osebe vstavi podano klicno predpono (npr. 1234567 \rightarrow +3861234567).

Dolžina telefonske številke brez predpone ni večja od 20, dolžina predpone pa ni večja od 10. V 40% skritih testnih primerov je predpona vedno enaka +386, v polovici *od teh* primerov pa je telefonska številka brez predpone vedno dolga natanko 8 znakov.

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

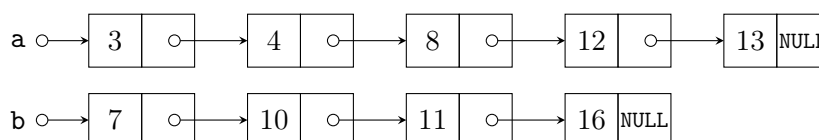
```
typedef struct _Vozlisce {
    int podatek;
    struct _Vozlisce* naslednje;
} Vozlisce;
```

V datoteki `naloga2.c` dopolnite funkcijo

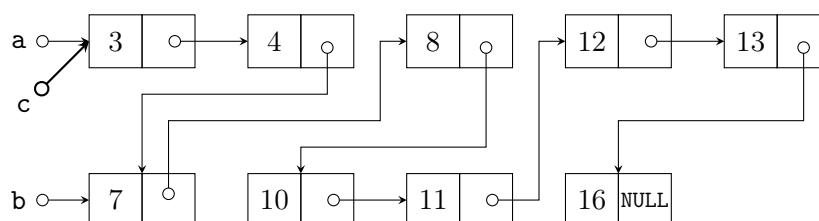
`Vozlisce* zlij(Vozlisce* a, Vozlisce* b),`

tako da bo znila podana naraščajoče urejena neprazna povezana seznama (kazalec `a` kaže na prvo vozlišče prvega, kazalec `b` pa na prvo vozlišče drugega seznama) v nov naraščajoče urejen povezani seznam. Novi seznam mora biti sestavljen iz *obstoječih* vozlišč obeh seznamov. Funkcija naj vrne kazalec na začetek novega seznama (ta kazalec bo torej enak bodisi kazalcu `a` bodisi kazalcu `b`).

Na primer, program `test01.c` preverja, ali se seznama



po klicu `Vozlisce* c = zlij(a, b)` zlijeta v seznam



Noben seznam ne vsebuje več kot 10^5 vozlišč. Vse komponente `podatek` so med seboj različne.

V 26% skritih testnih primerov velja, da je zadnji element prvega seznama manjši od prvega elementa drugega seznama. Če boste seznam zgradili iz novih vozlišč (namesto da bi na novo povezali obstoječa vozlišča), boste za pripadajoči testni primer prejeli 0,5 točke.

- ③ Napišite program `naloga3.c`, ki s standardnega vhoda prebere število $d \in [2, 50]$, niz S , sestavljen iz d medsebojno različnih alfanumeričnih znakov, ter število $n \in [1, \log_d 10^6]$, na standardni izhod pa izpiše vse nize dolžine n , ki so sestavljeni iz znakov niza S in v katerih nobeden od znakov ne nastopa več kot trikrat. Nize lahko izpišete v poljubnem vrstnem redu, zagotovite le, da bo vsak v svoji vrstici in da boste vsakega izpisali natanko enkrat. Na primer:

test01.in:

2 ab 5

test01.out:

aaabb
aabab
aabba
aabbb
abaab
ababa
ababb
abbaa
abbab
abbba
baaab
baaba
baabb
babaa
babab
babba
bbaaa
bbaab
bbaba
bbbba

Če boste izpisali *vse* nize dolžine n , sestavljene iz znakov niza S , boste za pripadajoči testni primer prejeli 0,3 točke.