Programovanie (1) v C/C++ 2026/27
Cvičenia 10, príklad 3
Zmaž

V tejto úlohe budeme pracovať s **obojsmerne spájaným zoznamom**, v ktorom si každý uzol pamätá smerník `next` na ďalší uzol a smerník `prev` na predchádzajúci. Zoznam je navyše **kruhový**, to znamená, že položka `next` v poslednom prvku ukazuje na prvý prvok a položka `prev` v prvom prvku ukazuje na posledný.

Vašou úlohou je do **priloženej kostry** doprogramovať funkciu `deleteNode`, ktorá dostane smerník na uzol takéhoto zoznamu a zmaže uzol, ktorý nasleduje v zozname za ním. Môžete predpokladať, že aj po zmazaní v zozname zostane aspoň jeden uzol. Vaša funkcia by mala uvoľniť pamäť alokovanú pre tento uzol a tiež správne nastaviť smerníky `next` a `prev` v susedných uzloch.

Funkcia `main` v kostre na začiatku vytvorí zoznam s jedným uzlom obsahujúcim hodnotu -1. Potom načítava a vykonáva postupnosť príkazov zadaných ako jednopísmenové skratky uvedené nižšie. Celý čas zoznam obsahuje aspoň jeden uzol a funkcia `main` si pamätá smerník na jeden z uzlov.

- Príkazy `'f'` a `'b'` (forward a backward) posunú aktuálny smerník o jeden uzol dopredu resp. dozadu.
- Príkazy `'p'` a `'r'` (print a print reverse) vypíšu zoznam začínajúc od aktuálneho prvku, pričom príkaz `'p'` vypisuje smerom dopredu (po smerníkoch `next`) a príkaz `'r'` smerom dozadu (po smerníkoch `prev`).
- Príkaz `'i'` (insert) vloží nový uzol do zoznamu za aktuálny uzol. Zo vstupu sa načíta celé číslo, ktoré sa do uzla uloží ako dáta.
- Príkaz `'d'` (delete) zmaže uzol, ktorý nasleduje za aktuálnym uzlom zoznamu pomocou vašej funkcie.
- Príkaz `'e'` (end) ukončí postupnosť príkazov.

Po poslednom príkaze program uvoľní pamäť zvyšných prvkov zoznamu, pričom opäť používa vašu funkciu. Kým vaša funkcia nefunguje, môžete uvoľnenie pamäte dočasne zakomentovať, v odovzdanom programe ho však nechajte funkčné.

Upozornenie: Nemeňte už hotové časti programu, programy budeme dodatočne kontrolovať.

Príklad vstupu:

```
p
i 1 i 2 p
f f i 3 p
b p
d p
r
e
```

Príklad výstupu:

```
-1
-1 2 1
1 3 -1 2
2 1 3 -1
2 3 -1
2 -1 3
```

Zoznam predstavujúci druhý riadok výstupu:

