

Programovanie (1) v C/C++ 2023/24

Cvičenia 9, príklad 2

Nuly

Priložená **kostra** obsahuje niekoľko funkcií na prácu s jednosmerným spájaným zoznamom. Zoznam pozostáva z niekoľkých uzlov typu `node`. Navyše záznam typu `linkedList` obsahuje smerník na prvý uzol zoznamu (tento smerník bude `NULL`, ak je zoznam prázdny).

Na začiatku kostry zo vstupu načíta niekoľko celých nezáporných čísel a uloží ich do zoznamu v pôvodnom poradí. Za posledným číslom je na vstupe číslo `-1`. Zoznam môže byť aj prázdny, ak vstup obsahuje iba `-1`. Na konci programu zoznam vypíše a uvoľní pamäť.

Vašou úlohou je do kostry doprogramovať funkciu `vyhod`, ktorá zo zoznamu vyhodí všetky uzly, v ktorých je ako dáta uložená hodnota nula. Uvoľnite tiež pamäť, ktorú vyhodené uzly zaberajú. Na výstupe by teda správne mali byť iba kladné čísla zo vstupu, v pôvodnom poradí.

Upozornenie: Nemeňte už hotové časti programu. Testovač nekontroluje správne uvoľnenie pamäti, ale budeme to dodatočne kontrolovať. Ak si chcete uvoľnenie pamäte skontrolovať na vašom počítači, môžete použiť program `valgrind` (pre Linux) alebo `Dr. Memory` (pre Windows aj Linux). Návod na použitie programu `valgrind` nájdete na stránke predmetu.

Návod: Jedna možnosť je pomerne priamočiario prejsť zoznamom od začiatku po koniec a vyhadzovať čo treba. Ale pozor, aby ste vyhodili prvok zo zoznamu, potrebujete smerník na jeho predchodcu, lebo tomuto predchodcovi budete meniť položku `next`. Ak máme v premennej `v` smerník na uzol a chceme zmazať za ním nasledujúci uzol, môžeme použiť nasledujúci kód:

```
node *to_del = v->next;
v->next = v->next->next;
delete to_del;
```

Pozor, keď chceme zmazať nejaké uzly zo začiatku zoznamu, tieto nemajú predchodcu, ale namiesto toho treba meniť položku `first` v zázname typu `linkedList`.

Druhá možnosť je napísať pomocnú rekurzívnu funkciu, ktorá dostane smerník na prvý uzol starého zoznamu a vráti smerník na prvý uzol zoznamu po vymazaní núl. Výsledok tejto rekurzívnej funkcie sa potom vo funkcii `vyhod` uloží do položky `first` v zázname typu `linkedList`. V rekurzívnej verzii sa potrebujete zaoberať iba spracovaním prvého prvku zoznamu a na zvyšok zoznamu zavoláte rekurziu.

Príklad vstupu:

```
0 1 0 0 0 2 3 0 -1
```

Príklad výstupu:

```
1 2 3
```

Pripomíname, že číslo `-1` sa do zoznamu neukladá, používa sa len pri načítavaní na ukončenie zoznamu.

Pre zvedavých: Na prednáške sme videli iba pridávanie na začiatok zoznamu, ale v kostre pridávame načítané prvky na koniec zoznamu. Aby sme nemuseli vždy pred pridaním prvku hľadať posledný uzol zoznamu, pamätáme si ho v premennej. Aby sme nemuseli zvlášť programovať prídanie úplne prvého prvku a prídanie ďalších, používame techniku dvojitého smerníku typu `node **`. Skúste si na papieri nakresliť, ako prebieha vo funkcii prídanie prvých dvoch prvkov do prázdneho zoznamu.