

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

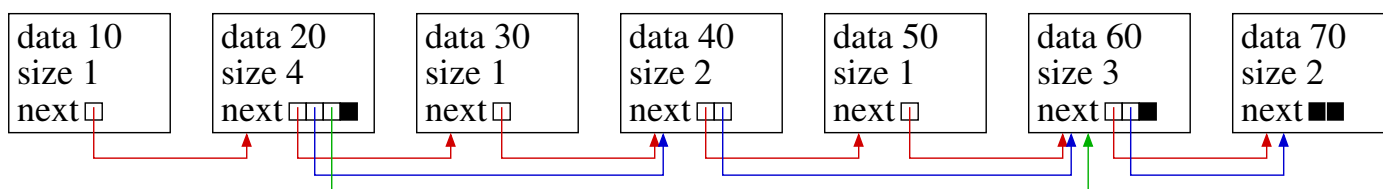
Cvičenia 9, príklad 3, bonus

Skratky 2

V tomto príklade budeme pokračovať v práci so zoznamom z príkladu 3. Prvá časť vstupu vyzerá rovnako a aj sa rovnako uloží do pamäte. Môžete predpokladať, že vstupný zoznam spĺňa nasledovné podmienky:

- Hodnoty uložené v položke `data` tvoria striktno rastúcu postupnosť, žiadne číslo sa neopakuje.
- Ak je v `next[i]` hodnota `NULL`, tak za ňou už v poli `next` budú nasledovať iba ďalšie hodnoty `NULL`.
- Ak `next[i]` ani `next[i+1]` nie sú `NULL`, tak buď ukazujú na to isté miesto v zozname, alebo `next[i+1]` ukazuje ďalej v zozname (na väčšie hodnoty) ako `next[i]`.

Tieto podmienky spĺňa aj príklad z minulého zadania:



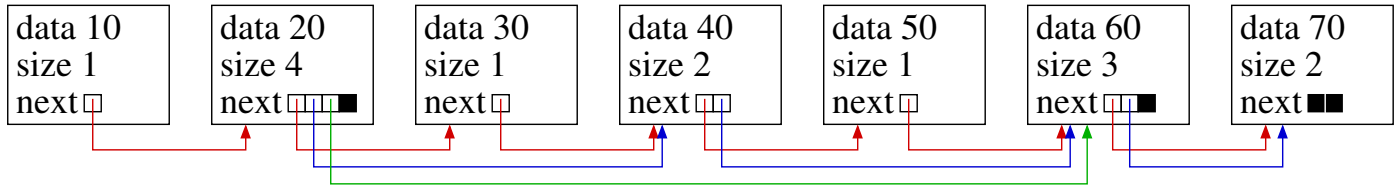
Vašou úlohou je naprogramovať vyhľadávanie zadaného čísla x v tomto zozname. Postupne sa budeme posúvať zľava doprava po zozname, pričom chceme použiť vždy čo najvýhodnejšiu skratku, ale tak, aby sme hľadanú hodnotu x nepreskočili. Ak sme v určitom uzle, môžu nastať tri prípady:

- Hodnota v uzle je rovná x . V tom prípade môžeme skončiť hľadanie s úspechom.
- Hodnota v uzle je väčšia ako x . V tom prípade hľadanie končí neúspechom, x sa v zozname nenachádza.
- Hodnota v uzle je menšia ako x . Potom medzi smerníkmi v poli `next` nájdite posledný (najpravejší), ktorý ukazuje na uzol s hodnotou menšou alebo rovnou x , do tohto uzla sa presuňte a pokračujte v hľadaní rovnakým spôsobom. Ak žiaden zo smerníkov v poli `next` nemá túto vlastnosť, prvok x sa v zozname nenachádza.

Vyhľadávanie napíšete do funkcie `contains`, ktorá dostane smerník na prvý uzol zoznamu a vráti logickú hodnotu indikujúcu, či sa prvok x podarilo nájsť. Funkcia by mala fungovať aj v prípade, že vstupný smerník je `NULL`, čo znamená, že zoznam je prázdny. Funkcia má na výstup vypísať hodnotu `data` vo všetkých uzloch, ktoré navštívi. Pred každú hodnotu vypíšte medzeru. Zvyšok výpisu je hotový vo funkcii `main`.

Priložená **kostra** načíta zoznam v rovnakom formáte ako v príklade 3. Potom postupne číta viacero hodnôt x a pre každú spustí vaše vyhľadávanie a vypíše jeho výsledok. Za poslednou hodnotu x je na vstupe opäť číslo `-1`. **Nemeňte už hotové časti programu.**

Príklady na ďalšej strane.



Príklad vstupu:

```
10 1 20 4 30 1 40 2 50 1 60 3 70 2 -1
10 5 70 90 35 -1
```

Príklad výstupu:

```
hľadame 10
prejdene uzly 10
najdene

hľadame 5
prejdene uzly 10
nenajdene

hľadame 70
prejdene uzly 10 20 60 70
najdene

hľadame 90
prejdene uzly 10 20 60 70
nenajdene

hľadame 35
prejdene uzly 10 20 30
nenajdene
```

Príklad vstupu:

```
-1
10 -1
```

Príklad výstupu:

```
hľadame 10
prejdene uzly
nenajdene
```