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

Tento príklad aj všetky ďalšie zadania tento semester budú používať kompilátor jazyka C++, pri kompilovaní teda použite príkaz g++ namiesto gcc.

Vašou úlohou je do **priloženej kostry** naprogramovať funkciu `kTyUzol`, ktorá dostane smerník na uzol spájaného zoznamu a pozíciu k a vráti smerník na uzol, ktorý je o k pozícií v zozname ďalej. Teda ak k je nula, vráti vstupný smerník, ak k je jedna, vráti jeho následníka atď. Ak za zadaným uzlom už nasleduje menej ako k uzlov, funkcia vráti NULL. Vstupný smerník môže byť aj NULL, vtedy funkcia vráti NULL pre ľubovoľné k. Vo vašej funkcii zoznam nemeňte ani nevytvárajte žiadne nové uzly.

Priložená kostra zo vstupu načíta niekoľko celých nezáporných čísel a uloží ich do zoznamu v pôvodnom poradí. Za posledným číslom je na vstupe číslo -1, ktoré sa do zoznamu neuloží. Zoznam môže byť aj prázdny, ak vstup obsahuje iba číslo -1. Potom program opakovane načíta požadované číslo k a zavolá vašu funkciu, pričom vždy začína od miesta, ktoré vaša funkcia vrátila v predchádzajúcom volaní. Za posledným číslom k nasleduje opäť číslo -1. Po každom volaní vašej funkcie program vypíše číslo uložené v aktuálnom uzle alebo slovo NULL, ak funkcia vrátila NULL.

Úplne na konci kostra vypíše vstupný zoznam a uvoľní pamäť. Za vypísaným zoznamom vypíše číslo -1.

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

Príklad vstupu:
```
6 4 5 2 1 -1
2 0 1 5 1 -1
```

Príklad výstupu:
```
5 5 2 NULL NULL
6 4 5 2 1 -1
```

Príklad vstupu:
```
-1
0 1 -1
```

Príklad výstupu:
```
NULL NULL
-1
```

Príklad vstupu:
```
1 2 3 -1
0 0 4 -1
```

Príklad výstupu:
```
1 1 NULL
1 2 3 -1
```

Pripomíname, že číslo -1 sa do zoznamov neukladá, používa sa len pri načítavaní a výpise 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 pridanie úplne prvého prvku a pridanie ďalších, používame techniku dvojitého smerníku typu `node **`. Skúste si na papieri nakresliť, ako prebieha vo funkcii pridanie prvých dvoch prvkov do prázdneho zoznamu.