Programovanie (1) v C/C++ 2024/25
Tréningový príklad na skúšku 2b
Slová

Toto je cvičný príklad pre druhý príklad na skúške. Odporúčame prečítať si zadanie až keď budete úlohu skutočne cvične riešiť.

Dopíšte chýbajúce časti programu, ktorý hľadá zadaný reťazec v zadanom slovníku slov, pričom ho zaujímajú všetky slová zo slovníka, vo vnútri ktorých sa reťazec nachádza. Napríklad reťazec `les` sa nachádza v samotnom slove `les`, ale aj v slove `pleso` a mnohých ďalších, ale nenachádza sa v slove `lieska`, lebo vyžadujeme, aby výskyt tvoril súvislý úsek.

Na vstupe je najskôr slovník: zoznam slov pozostávajúcich z malých písmen anglickej abecedy. Tieto slová sú oddelené medzerami alebo koncami riadkov. Za posledným slovom slovníka ide špeciálne slovo `END`. Potom program načítava po jednom reťazce pozostávajúce z malých písmen anglickej abecedy a pre každý reťazec vypíše zoznam slov zo slovníka, v ktorých sa daný reťazec nachádza. Nájdené slová vypisujte v abecednom poradí na jeden riadok oddelené medzerami, vo formáte ako v príklade nižšie. Za posledné nájdené slovo už medzeru nepíšte a ak žiadne slovo nebolo nájdené, riadok končí dvojbodkou. Reťazce, ktoré treba vyhľadávať, sú tiež na vstupe oddelené koncami riadkov alebo medzerami a za posledným z nich ide slovo `END`. Môžete predpokladať, že žiadne slovo sa v slovníku neopakuje a že všetky slová v slovníku aj reťazce, ktoré treba vyhľadávať, majú dĺžku najviac 100.

Tento príklad by sa dal riešiť pomerne priamočiaro, ak by sme si slová v slovníku uložili priamo do poľa, ale v tejto úlohe budeme mať slová v slovníku uložené v prefixovom strome. Do priloženej **kostry** programu doprogramujte funkcie alebo časti funkcií `createNode`, `endsWith`, `writeWord` a `search`. Program uloží slovník do prefixového stromu, pričom každý vrchol si pamätá aj smerník na rodiča, svoju hĺbku v strome a znak na hrane vedúcej z rodiča do vrchola. Správne nastavenie smerníka na rodiča a hĺbky doprogramujte do funkcie `createNode`.

Pri samotnom hľadaní vo funkcii `search` prejdite stromom a v každom vrchole v zistite, či náhodou reťazec tvorený znakmi na ceste z koreňa do vrchola v nekončí na hľadaný reťazec pomocou funkcie `endsWith` a ak áno, môžete vypísať všetky slovníkové slová v podstrome s koreňom vo v pomocou funkcie `writeSubtreeWords`. Vďaka smerníku na rodiča a znakom uloženým vo vrcholoch môžeme jednoduchým pochodovaním smerom nahor zistiť, akému reťazcu daný vrchol prislúcha, čo sa môže zísť vo funkciách `endsWith` a `writeWord`.

Každá funkcia má v hlavičke napísané, čo má robiť; tento popis dodržujte. Dátové štruktúry majú tiež v definícii popísaný význam ich jednotlivých zložiek, ktorý by mal váš program dodržiavať. Naprogramujte všetky požadované funkcie, aj keby ste ich v programe nevyužili. Podľa potreby môžete naprogramovať a použiť aj ďalšie funkcie. V programe uvoľnite všetku pamäť, ktorú alokujete.

Príklad vstupu:

```
les prales pleso lieska lesny END
les p END
```

Príklad výstupu:

```
Vyskyty retazca "les": les lesny pleso prales
Vyskyty retazca "p": pleso prales
```

Príklad vstupu:

```
aaa bab
babab aab baab
END
ab
bb
a
END
```

Príklad výstupu:

```
Vyskyty retazca "ab": aab baab bab babab
Vyskyty retazca "bb":
Vyskyty retazca "a": aaa aab baab bab babab
```