

Programovanie (1) v C/C++ 2022/23

Tréningový príklad na skúšku 2c

Premenné

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

Priložená **kostra** programu pracuje s aritmetickými výrazmi reprezentovanými vo forme binárneho stromu. Vašou úlohou bude dopísať niektoré chýbajúce časti. Aritmetické výrazy v tejto úlohe obsahujú celočíselné konštanty, premennú x a operátory $+$ a $*$. Typ `node` reprezentuje jeden vrchol stromu. Má dva smerníky na deti (`left` a `right`), ktoré majú hodnotu `NULL` v prípade konštanty a premennej. Položka `op` typu `char` určuje typ vrchola. Môže obsahovať znamienko `'+'` alebo `'*'`, alebo znak `'x'` označujúci, že ide o vrchol pre premennú, alebo znak medzera `' '` označujúci, že ide o konštantu. Položka `val` sa používa iba vo vrcholoch pre konštanty a obsahuje samotnú hodnotu konštanty. Vašou úlohou je doprogramovať nasledujúce časti programu.

- Naprogramujte funkciu `containsX`, ktorá dostane vrchol v a zistí, či sa niekde v strome s koreňom v nachádza premenná x .
- Naprogramujte funkciu `simplify`, ktorá dostane vrchol v a upraví strom s koreňom v v tomto vrchole tak, že každý podstrom, ktorý neobsahuje premennú x , nahradí konštantou, ktorá zodpovedá hodnote jeho výrazu. Napríklad vo výraze $((1+2)*x)$ podvýraz $(1+2)$ neobsahuje premennú a teda ho môžeme nahradiť konštantou 3, čím dostávame zjednodušený výraz $(3*x)$. Niekedy sa pomocou `simplify` môže zmeniť aj koreň stromu a preto funkcia vráti smerník na koreň stromu po zjednodušení. (Vrátený smerník teda môže ukazovať buď na pôvodný vrchol v alebo na nejaký novovytvorený vrchol.) Vo funkcii `simplify` sa vám môže hodiť funkcia `containsX`, dá sa to však aj bez nej. Odalokujte všetky vrcholy, ktoré už nie sú potrebné.
- Opravte funkciu `readPrefix`, ktorá z konzoly číta výraz v prefixovej notácii a vytvára z neho strom. Funkcia funguje správne, ak sú všetky konštanty vo výraze nezáporné. Ak však niektorá konštantá začína znamienkom `-`, funkcia toto znamienko odignoruje a načíta iba absolútnu hodnotu. Opravte túto chybu.

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. Podľa potreby môžete naprogramovať a použiť aj ďalšie funkcie, ale pokiaľ možno nemeňte už hotové časti programu.

V programe je už hotové načítanie a výpis dát, ktoré nemeňte (okrem opravy načítania záporných čísel). Na prvom riadku vstupu je počet výrazov a na každom z ďalších riadkov jeden výraz v prefixovej notácii. Jednotlivé operátory, konštanty a premenné sú vo výraze oddelené medzerami. Program vypisuje na konzolu načítaný výraz v infixovej notácii, to či obsahuje premennú a napokon ten istý výraz po zjednodušení.

Pomôcka: V programe je hotových niekoľko funkcií na prácu so výrazmi, ktoré môžete použiť. Môžu sa vám zísť napríklad funkcie `destroyTree`, `evaluate` a `createNum`.

Príklad vstupu:

```
5
* + 1 2 x
+ 1 * 3 + 1 1
+ + + 1 2 x + x + 3 4
+ * x 0 0
* + -1 -2 3
```

Príklad výstupu:

```
Povodny vyraz: ((1 + 2) * x)
Vyras obsahuje premmenu x
Po volani funkcie simplify: (3 * x)
```

```
Povodny vyraz: (1 + (3 * (1 + 1)))
Vyras neobsahuje premennu x
Po volani funkcie simplify: 7
```

```
Povodny vyraz: (((1 + 2) + x) + (x + (3 + 4)))
Vyras obsahuje premmenu x
Po volani funkcie simplify: ((3 + x) + (x + 7))
```

```
Povodny vyraz: ((x * 0) + 0)
Vyras obsahuje premmenu x
Po volani funkcie simplify: ((x * 0) + 0)
```

```
Povodny vyraz: (((-1) + (-2)) * 3)
Vyras neobsahuje premennu x
Po volani funkcie simplify: (-9)
```