

Naprogramujte v jazyku C funkci `int evaluate(const char *formula)`, která vyhodnotí danou formuli skládající se z kulatých závorek, číslic (0 až 9) a operátorů (+ pro sčítání, \* pro násobení). Parametr `formula` obsahuje algebraický výraz, který je vždy ve tvaru:

1. libovolný řetězec číslic;
2. levá závorka, libovolný řetězec číslic, pravá závorka;
3. levá závorka, libovolný algebraický výraz, libovolný operátor, libovolný algebraický výraz a pravá závorka;
4. libovolný algebraický výraz, libovolný operátor, libovolný algebraický výraz.

Algebraické výrazy tedy mohou, ale nemusí být uzávorkovány. Pro zvýšení přehlednosti mohou algebraické výrazy obsahovat také mezery, které budou při vyhodnocování zcela ignorovány. Násobení má při vyhodnocování výrazů samozřejmě přednost před sčítáním.

Při řešení úlohy doporučuji využít princip rekurze, který nám na základě definice algebraického výrazu (viz výše) dovoluje:

1. Odstranit případné závorky kolem výrazu.
2. Výraz skládající se z číslic, převést na celé číslo a to vrátit jako výsledek.
3. Najít operátor (který není uvnitř žádných závorek; pokud jich je více vzít ten s nižší prioritou), rekurzivně vyhodnotit první algebraický podvýraz (od operátoru vlevo), poté druhý algebraický podvýraz (od operátoru vpravo) a na jejich výsledky aplikovat matematickou operaci odpovídající nalezenému operátoru.

#### **Příklady použití:**

```
evaluate("12");
evaluate("( 12 )");
evaluate("( ( 12 ) )");
evaluate("(1+2)");
evaluate("(1 + (2) )");
evaluate(" ( 1 + 2 ) ");
evaluate("( 1 + (2*3) )");
evaluate("( (1+2) * (2+3) )");
evaluate(" ( ( 1+(2 *3) ) +4 )");
evaluate("10*2+2*3");
evaluate("10 * (2+3)");
```

#### **Výsledky příkladů:**

12  
12  
12  
3  
3  
3  
7  
15  
11  
26  
60