

Задачи

1. Напишете функция, която прочита файл израз от скоби и валидира дали те са правилно подредени. За израз с правилно подредени скоби се счита израз, в който за всяка отваряща скоба съответства затваряща такава. Поддържат се следните видове скоби – ‘(’, ‘)’, ‘{’, ‘}’, ‘[’, ‘]’.
2. Напишете програма, която пресмята стойността на инфиксни изрази.
Напр.: $(1 + ((2 + 3) * (4 * 5)))$
3. Създайте клас **VendingMachine**, който съдържа **4** слота за продукти (по един за всеки вид поддържан продукт). Нашият **VendingMachine** трябва да поддържа следните команди:
 - a. **hasEmpty()** – проверява дали има поне един празен слот
 - b. **addItem(<slotNumber>, <*Items>)** – добавя няколко продукта в посочената колонка
 - c. **consume(<slotNumber>)** – взима последно добавеният **Item** от съответната колонкаПоддържаните продукти ще се представят от class **Item** с единствена данна price.
Производните класове са следните:
 - a. **class Cola** със следното поле
 - **flavor** {cherry, lemon, vanilla, zero, raspberry, ginger}
 - b. **class Water** със следното поле
 - **bottleSize** {halfLitre, oneLitre, litreAndHalf}
 - c. **class Croissant** със следното поле
 - **flavor** {cherry, coconut, strawberry, hazelnut}
 - d. **class BakeRolls** със следното поле
 - **flavor** {salt, pizza, garlic, tomato, cheeseAndSpinach}
4. Веднъж след поредният турбо як купон, Ванката трябвало да проверява домашни по математика. Но този път било различно, математическите термове почнали да се разместват по един доста интересен начин. Например $4 + 2 * 3$ се размества като $4\ 2\ 3\ *\ +$. Термовете на Ванката съдържали само операциите $+$, $-$, $/$, $*$. Помогнете на Ванката да провери домашните, като напишете функция, която приема такъв терм и връща резултата от оценката му.
5. Даден ви е *stack*, от вас се иска да може да кажете кой е максималния елемент във всеки един момент (може да е на върха на стека, но след като бъде поп-нат новия максимален елемент ще е един от останалите).
6. Трябва да създадете опростен текстов редактор. Първаначално редакторът съдържа празният string *S*. Трябва да поддържате следните 4 операции:
 - a. **append(W)** – добавя string *W* към *S*
 - b. **delete(K)** – изтрива последните *K* символи от *S*
 - c. **print(k)** – принтира *k*-тия символ от *S*
 - d. **undo** - върни *S* в състоянието преди извършване на операция от тип 1 или 2
7. В една градина има *n* на брой цветя. Всяко цвете е третирано с определено количество пестициди. В края на всеки ден, ако някое цвете е било третирано с повече пестициди от съседното му вляво, то умира.
В началото се подава *n* - броя на цветята. На следващият ред *n* числа, разделени с интервал - количеството пестициди, с които е третирано *i*-тото цвете. Вашата програма

трябва да изведе броя на дните, след които никое цвете не умира (няма цветя с повече пестициди от левия си съсед).

Пример: Нека първоначално количеството пестициди за всяко цвете е $p = [3, 6, 2, 7, 5]$. След първия ден умират второто и четвъртото цвете ($p[1]$, $p[3]$) и остават следните - $p = [3, 2, 5]$. След втория ден третото цвете от първоначалния масив($p[2]$) умира, т.е. остават $p = [3, 5]$. Вече няма цветя с повече пестициди от левия си съсед, следователно отговора е 2.

Вход

5

3 6 2 7 5

Изход

2

Вход

7

6 5 8 4 7 10 9

Изход

2

8. Напишете алгоритъм, който да помогне на любимият ви учител да оцени домашните по математика. Алгоритъмът трябва по дадени равенства (състоящи се от математически изрази, разделени със знаци за равенство) да даде оценка на коректността на равенството.

Примери:

"3 + 2 = 5" => "1 / 1"

Има само една стъпка и тя е изпълнена правилно.

"7 - 3 * 2 + 1 = 4 * 2 + 1 = 8 + 1 = 9" => "2 / 3"

Първата стъпка е грешна, но следващите 2 не са, поради тази причина пишем 2 от 3 точки.