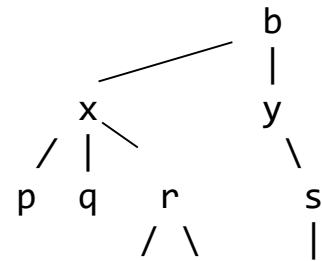


**Задача 2.** Задачата да се реши на един от езиките C/C++ или Java. В началото на решението си посочете кой език сте избрали.

Троично дърво от латински букви ще наричаме следната структура:

- Празното дърво е троично дърво от латински букви;
- Ако  $T_1$ ,  $T_2$  и  $T_3$  са троични дървета от латински букви, а  $x$  е латинска буква, то наредената четворка  $\langle x, T_1, T_2, T_3 \rangle$  също е троично дърво от латински букви.

а) Да се напише функция **readLast**, която по дадено троично дърво от латински букви намира думата, която може да се прочете отляво надясно в последното му ниво. Ниво на троично дърво от латински букви наричаме списък от възли в дървото, които са равноотдалечени от корена.



**Пример:** за дървото вдясно функцията **readLast** трябва да връща думата “cat”.

Представянето на дървото е по Ваш избор. Опишете избраното от Вас представяне.

б) Да се напише функция **serialize**, която по дадени: (1) троично дърво от латински букви и (2) низ, описващ път до файл, записва дървото в текстов файл. Текстовото представяне на троично дърво да бъде следното:

- Текстовото представяне на празното дърво е звездичка (“\*”);
- Текстовото представяне на троичното дърво  $\langle x, T_1, T_2, T_3 \rangle$  е “(x T1 T2 T3)”, където  $T_1$ ,  $T_2$  и  $T_3$  са текстовите представяния съответно на  $T_1$ ,  $T_2$  и  $T_3$ .

**Пример:** Текстовото представяне на дървото горе е:

(b (x (p \* \* \*) (q \* \* \*) (r (c \* \* \*) \* (a \* \* \*))) (y \* \* (s \* (t \* \* \* \*) \*) \*)

За реализацията на функциите **readLast** и **serialize** е позволено използването на стандартните за съответния език библиотечни функции.