Домашна работа 3

курс Структури от данни и програмиране
за специалност Информатика
зимен семестър 2019/2020 г.

Правила

Следните правила описват процеса по реализирането и предаването на домашните по СДП.

1. Срок за предаване на Домашна работа 3: **23:59 ч на 08.01.2020 г**.
2. По домашната се работи самостоятелно (т.е. не се допуска работа в екипи)
3. Не губите нищо ако предадете частично направена домашна! По-добре се опитайте да решите поне една задача, отколкото да си кажете, че не можете!
4. Плагиатство от колеги и от други източници води до анулиране на работата.
5. Предаването става чрез прикачване на ZIP архив към съответното задание в Moodle, който съдържа всички файлове, необходими за компилирането на задачите от домашната.
6. Основните критерии при оценяването на домашните ще бъдат:
    - успешно изпълнение на поставеното условие;
    - използването на най-подходящите структури от данни;
7. Другите критерии при оценяването са:
    - добро стилизиране и форматиране на кода;
    - сложности;
    - следване на добри практики за писане на код;
    - спазване на ООП парадигмата;

Задача

Нека имаме следния вид аритметичен израз представен в прав полски запис, т.е.:

<израз> ::= <цяло число> | <променлива> | <операция> <израз> <израз>
<променлива> ::= a | b | c | d | … | y | z
<операция> ::= + | - | *

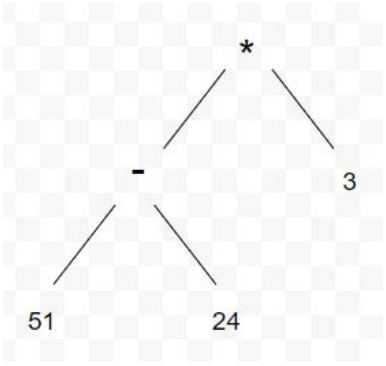
При правия полски запис операторът се намира точно преди двата операнда, за които трябва да се приложи (а при обратния полски запис операторът се намира точно след тях), например:

- **51+24** се записва като **+ 51 24** в прав полски запис и **51 24 +** в обратен полски запис
- **(51+24)*3** се записва като *** + 51 24 3** в прав полски запис и **51 24 + 3 *** в обратен полски запис
- **51+24*3** се записва като **+ 51 * 24 3** в прав полски запис и **51 24 3 * +** в обратен полски запис
- **(51+24)*3-48*79** се записва като **- * + 51 24 3 * 48 79** в прав полски запис и **51 24 + 3 * 48 79 * -** в обратен полски запис
- **(a+31)*b** се записва като *** + a 31 b** в прав полски запис и **a 31 + b *** в обратен полски запис

Заб.: при представянето на правия и обратния полски запис *като низ* между всяко цяло число, променлива и операция има разделител от едно празно място.

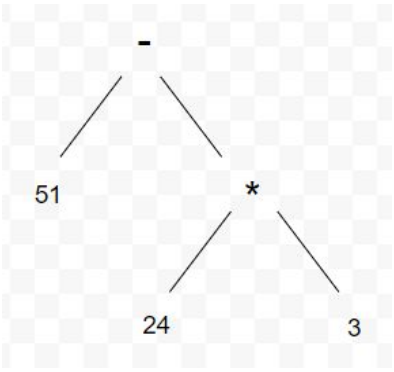
а) да се реализира външна функция, която строи двоично дърво на аритметичен израз, който е подаден *като низ в прав полски запис*. В рамките на това дърво операциите се явяват корен на цялото дърво или на поддърво, левият операнд се съдържа в лявото поддърво на съответния си оператор, а десният операнд - в дясното.

За * - 51 24 3 едно възможно дърво на израза би било:



,

а за - 51 * 24 3 едно възможно дърво на израза би било:



б) с помощта на дървото от точка а) изведете същия аритметичен израз

- в обратен полски запис като оставяте празно място между всяко число, променлива и операция подобно на записа в прав полски запис
- в нормален (инфиксен) запис като оградите в скоби всяка група от два операнда и операция, например за горните два примера това означава съответно ((51-24) *3) и (51-(24*3))

в) с помощта на дървото от точка а) пресметнете стойността на аритметичния израз (*без да копирате целия израз в друга структура от вида на стек, опашка, списък и т.н.*). При наличие на променлива в израза, поискайте от потребителя да въведе цяло число и използвайте това число за стойност на променливата.

Забележки:

1. За целите на домашното трябва в main функцията да прочетете от файл *поне 3* израза записани в прав полски запис като всеки израз трябва да е на отделен ред в рамките на файла. За всеки от тези изрази трябва да се приложат функциите от а), б) и в).
2. Във функциите, които дефинирате, можете да използвате обекти от тип дърво (например BinTree<T> или ваша собствена реализация) или указатели към корен на дървото (например TNode<T> *), където TNode представлява:

```
template <typename T>
struct TNode
{
        T data;
        TNode<T> *left, *right;
}
```

3. Можете да реализирате подточки б) и в) независимо от а) стига да генерирате пример, който да удовлетворява входните данни за подточки б) и в).
4. **Една променлива може да се среща повече от веднъж в рамките на даден аритметичен израз и в такъв случай стойността ѝ трябва да се въведе само веднъж от потребителя в подточка в).**

5. Можете да приемете, че изразът, който се подава в подточка а) е валиден.