

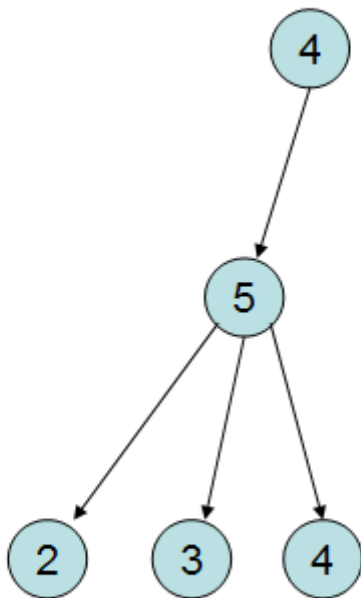
1. Напишете* реализация на дърво с произволен брой наследници на възел в дървото. Реализацията трябва да позволява зареждането на дървото от поток, където то е описано чрез следните операции:

- **nI** - въвежда числото като едноелементно дърво в стек от дървета
- **nM** - обединява най-горните n елемента от стека в дърво, което става нов връх на стека. Обединението става, като към корена на дървото на върха на стека се прибавят като наследници следващите $n-1$ дървета в стека.

тук с n е означено произволно цяло число, зададено като последователност от цифри. Като пример:

2I 3I 4I 5I 4M 4I 2M

задава дървото



(♣)Реализацията може (и даже, е желателно) да се базира на готова такава, която вие допълните само с функционалността за зареждане на дърво от поток.

2. Ypath заявка е низ от вида $a_1 \backslash a_2 \backslash a_3 \backslash \dots \backslash a_n$. По такава заявка и указател към връх от дърво, тя трябва да върне списък от стойности на върховете. Търсенето се извършва чрез навигация в дървото (т.е. преместване по върховете му) и избор на възли от него. Елементите $a_1, a_2, a_3, \dots, a_n$ са текстови низове и могат да приемат стойности от следните видове:

- *число* – тръгва се по наследника, равен на това число. Ако такъв няма – резултата е празния списък
- [*число*] - тръгва се по наследника, който е на позиция това число. Ако такъв няма – резултата е празния списък
- .. – тръгва се към родителя. Ако такъв няма – резултата е празния списък
- * > *число* – избират се всички наследници на текущия възел, по-големи от зададеното число. Ако такива няма – резултата е празния списък. За всеки наследник се изпълнява остатъка от Ypath заявката, като резултатите от тези подзаявки се обединяват в крайния резултат.
- * < *число* – избират се всички наследници на текущия възел, по-малки от зададеното число. Ако такива няма – резултата е празния списък. За всеки наследник се изпълнява остатъка от Ypath заявката, като резултатите от тези подзаявки се обединяват в крайния резултат.

Пример : за горното дърво и указател към елемента 5, заявката

..\[1]\[2] трябва да върне 3
2 трябва да върне 2
[3] трябва да върне 4
[3]\.. трябва да върне 5

Пример* : за горното дърво и указател към корена, заявката

5\>2 трябва да върне списъка 3,4
5\>2\.. трябва да върне списъка 5

Условията със * са незадължителни и представляват допълнителен бонус.

3. Напишете програма, която зарежда от стандартния си вход дърво описано във формата на т.1 след което по Ypath заявка зададена като аргумент на командния ред, извежда резултата от заявката на стандартния си изход.