

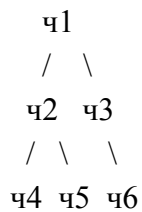
### Зад. 1

Да се напише програма, която проверява дали дадено двоично дърво е двоично наредено дърво.

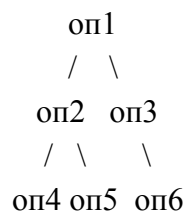
### Зад. 2

Дадени са две двоични дървета с еднаква структура – дърво от числа и дърво от операции, представени със символите +, - и \*. По-долу са дадени примери за такива дървета.

Дърво от числа:



Дърво от операции:



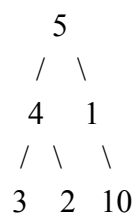
Да се напише функция, която пресмята сумата:

$$\sum_{i=1}^n S_i$$

където  $n$  е броят на върховете в кое да е от дърветата,  $S_i = Ч_i \text{ ОП}_i \text{ } C_i$ ,  
а  $C_i$  е броят на върховете в поддървото с корен върха  $Ч_i$  (включително върха  $Ч_i$ ).

Пример:

Дърво от числа:



Дърво от операции:

```
      *
     / \
    +  -
   / \ \
  +  * -
```

Резултат:  $(5 * 6) + (4 + 3) + (1 - 2) + (3 + 1) + (2 * 1) + (10 - 1) = 51$

### Зад. 3

Имаме двоично дърво, незадължително пълно. Да се напише програма, която проверява дали дървото е огледално-симетрично.

Примери:

1)  
1  
/ \ -> false  
2 3

2)  
1  
/ \  
2 2 -> true  
/ \  
3 3

3)  
1  
/ \  
2 2 -> true  
/ \ / \  
3 7 7 3  
/ \  
5 5

### Зад. 4

Дадено е двоично дърво и ниво  $k$  в дървото. Да се провери дали стойностите във върховете на дървото, намиращи се на ниво  $k$ , образуват палиндром (нивата се броят от нула, т.е. коренът на дървото се намира на ниво 0, неговите преки наследници на ниво 1 и т. нат.).

Примери:

1)

```
  1
 / \
2   3
1
```

2)

```
    1
   / \
  3   7
 / \  / \
1  2 2  1
-> true
```

2