

Работата с паметта е ваша отговорност. Посочените структури и типовете на елементите са задължителни. При оценяването на решенията ще се вземе предвид подредеността на решението и спазването на добрите практики при писането на код. Кодът ТРЯБВА да се компилира и да е придружен от главна функция, която да демонстрира работата му.

Задача 1. Дадено е двоично наредено дърво (дърво за търсене), което представя йерархия от служители в компания.

Структура на дървото:

- Всеки възел в дървото съдържа информация за служител - име, низ с максимална дължина 64 символа, дни трудов стаж, заплатата. Двама служители се различават по тяхното име и трудов стаж.
- Служителите трябва да бъдат **подредени в дървото по техния трудов стаж**. Ако двама служители са с еднакъв трудов стаж, за наредбата да се използва лексикографската наредба на имената.

Двоичното дърво се представя чрез указател към неговия корен, представен чрез следната структура:

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

- A. **(0.75 т.)** Да се реализира булева функция за добавяне на служител в йерархията. Ако служителят е вече добавен в йерархията, да се изведе подходящо съобщение за грешка и да се върне лъжа.
- B. **(0.75 т.)** Да се реализира функция, която намира медианата на стажа в компанията.
Бележка: Медиана е средната стойност в сортирана редица 1 2 3 4 5 => 3 или 0 2 4 50 => 3
** За реализация на функцията НЕ е позволено използване на масиви и други структури от данни.*
- C. **(1 т.)** Всяко ниво в дървото, започвайки от корена, отговаря на един разрез от йерархията на компанията. Да се напише функция, която намира средната заплата на даден разрез от йерархията. Нивото на корена е 1. Да се валидира, че на подаденото ниво има служители и ако такива не съществуват да се хвърли изключение (exception).

Да се напише функция, която определя разрезът с най-висока средна заплата в дървото.

Пример:

Средна заплата за разрез 2: Bob (3, 4500) и Charlie (7, 6000) => 5250

Средна заплата за разрез 5: "Няма такова ниво в йерархията"

```
Alice (5, 5000)
 /      \
Bob (3, 4500)  Charlie (7, 6000)
 \
David (5, 4800)
```

Разрезът с най-висока средна заплата: 2.

Задача 2. Нека булев израз има следния вид:

- t, което е True;
- f, което е False;
- ! (expr), което е логическо отрицание на вътрешния израз expr;
- & (expr1, expr2, ...), което е логическо И на 2 или повече израза expr1, expr2, ...;
- | (expr1, expr2, ...), което е логическо ИЛИ на 2 или повече израза expr1, expr2, ...;

- A. **(1 т.)** Да се напише функция, която по подаден булев израз връща неговия резултат. За изчислението се позволява използването единствено на структура от данни стек. Позволено е използване на помощни функции. Примери:

- !(f) => true
- |(f, t) => true
- &(t, f) => false
- |(&(t, f, t), !(t)) => false

- B. **(0.5 т.)** Дадена е опашка от булеви изрази от описания вид. Да се напише функция, която намира колко от изразите в опашката се оценяват със стойност истина. Функцията НЕ трябва да разрушава опашката и НЕ е позволено използване на допълнителни структури от данни.