

ЗИМНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА
ВЕЛИКО ТЪРНОВО
27 февруари – 1 март 2015 г.
Група А, 11-12 клас

Задача А2. ФИРМА

Автори: Георги Георгиев и Йордан Чапъров

В световната мегафирма XXXXXL съществува строга йерархия на служителите, която естествено се представя с кореново дърво. Всички служители са номерирани уникално с последователни цели, неотрицателни числа, започвайки от 0. Номер 0 има шефът на фирмата, който се разполага в корена на дървовидната йерархия. Служител с номер j е пряко подчинен на служител с номер i (и съответно служител с номер i е непосредствен началник на служител с номер j), ако връх с номер j е син на връх с номер i в дървото на фирмената йерархия. Всеки служител, в момента на назначаването си, получава за номер следващото неизползвано естествено число. В този момент се определя и неговото място в йерархията на фирмата. Има две възможности: новоназначеният служител ще бъде подчинен на някого от фирмата, но няма да бъде началник на никого (с други думи ще „увисне“ като листо във фирменото дърво) или ще „разкъса“ някоя пряка връзка на подчинение и ще стане пряко подчинен на служител с номер i и пряк началник на служител с номер j (до този момент служител с номер j е бил пряко подчинен на служител с номер i). Не се допуска да се назначи някой, който да стане началник на шефа на фирмата – той е несменяем и непоклатимо си стои в корена на дървото. Фирмата XXXXXL само се разраства по гореописания начин – няма уволнения и реорганизации.

Счита се, че служител с номер k е началник на служител с номер p (макар и непряк), ако в дървото на фирмената йерархия връх с номер k е предшественик на връх с номер p .

Вие сте IT специалист в XXXXXL и на вас е възложена суперотговорната задача да поддържате информацията за дървовидната йерархична структура на фирмата и да давате информация за това кой на кого е началник. И така, при вас постъпват заявки от два основни вида: да отразите назначаването на нов служител и да отговорите на въпроса дали служител с номер k е началник на служител с номер p . За целта трябва да напишете четири функции, които ще бъдат викани от програма на журито:

void init(); - вика се точно веднъж в началото на програмата. След викане на **init()** в дървото има един връх (корен) с номер 0. Може да се използва за инициализиране на структури от данни.

void add_leaf(int parent); - добавя се ново листо към дървото. Родителят на листото е **parent**. Номерът на новодобавения връх е най-малкото свободно естествено число (първият добавен връх ще е номер 1, после 2 и т.н.)

void add_above(int node); - добавя се нов връх непосредствено „над“ върха **node**, по средата на реброто между **node** и родителя на **node**. Ако, преди операцията **add_above(node)**, върхът **node** е имал родител **A**, **add_above(node)** създава връх с номер **B** – най-малкото свободно естествено число. Структурата на дървото се изменя: родител на връх **node** вече е новодобавеният връх **B**, родител на **B** е първоначалният родител на **node** - върхът **A**. Върхът **B** няма други синове освен **node**. **add_above** не може да се вика на корена на дървото, т.е. **node > 0**.

bool is_par(int x, int y); - Функцията връща **true** тогава и само тогава, когато върхът **x** е предшественик на върха **y** (не е задължително да е непосредствен предшественик). Т.е. функцията връща **true** когато **x** е на пътя от корена - връх **0** до връх **y** и **x != y**.

Вие трябва да предадете към проверяващата система файл **business.cpp**, който съдържа четирите функции, глобалните структури от данни и допълнителни функции, които са необходими за решението. Той **не трябва** да съдържа функция **main**.

ЗИМНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА
ВЕЛИКО ТЪРНОВО
27 февруари – 1 март 2015 г.
Група А, 11-12 клас

Пример

Ето една примерна последователност от заявки, които постъпват от програмата на журито в рамките на един тест:

init();	Създава се връх номер 0 (шеф на фирмата)
add_leaf(0);	Добавя се връх номер 1, син на номер 0
add_leaf(0);	Добавя се връх номер 2, син на номер 0
is_par(0, 1);	Запитване: отговор true
is_par(1, 2);	Запитване: отговор false
add_above(2);	Добавя се връх номер 3, син на 0. Нов родител на 2.
is_par(3, 2);	Запитване: отговор true
is_par(0, 2);	Запитване: отговор true
is_par(2, 3);	Запитване: отговор false

Оценяване

Нека m е общия брой викания на функции **add_leaf**, **add_above** и **is_par**.

Подзадача 1 (10 точки) : $2 \leq m \leq 5\,000$;
Подзадача 2 (60 точки) : $5\,000 < m \leq 200\,000$;
Подзадача 3 (30 точки) : $200\,000 < m \leq 500\,000$.

Точките за дадена подзадача се получават, само ако програмата премине успешно всички тестове, предвидени за нея.

Указания

На състезателите се предоставя файл **grader.cpp** и един тест във файлове **business.00.in** и **business.00.sol**. Целта е да могат да си тестват написаните функции. Примерът в **business.00.in** съответства на описаната по-горе последователност от заявки. Състезателите могат да напишат и собствен **grader** или да разширят предоставения. Не е гарантирано, че същият файл **grader.cpp**, който е предоставен на състезателите, ще се използва от журито. Файлът **grader.cpp** трябва да се постави в същата папка, където е **business.cpp**.

Компилацията може да се извърши по два начина – или като се използва командния ред, изписан в проверяващата система (раздел „Условие“ за тази задача) или като се махне коментара от реда :

```
//include "business.cpp"
```

и се компилира директно в Codeblocks.

Форматът на файл **business.00.in** е следния:

На първия ред е броят на заявките от тип **add_leaf**, **add_above** и **is_par**.

Следват толкова реда колкото са заявките. На всеки ред първият символ означава типа на заявката : L(leaf) – добавяне на листо; N(node) – добавяне на междинен връх; A(ask) – запитване. След него идват числата, които трябва да се подадат като параметри на съответната функция.