

F. 簡單題

Description

通常題目叫做簡單題的題目都不太簡單，就跟說自己很弱的競賽國手一樣其實一點也不弱。但，畢竟這題是行政出的，算法班的行政是整個算法班演算法能力相對最差的人，怎麼可能出難題呢？還是說，這其實是一種反向操作，透過行政的口中說出的簡單題，實際上是難到要騙光你所有的比賽時間來寫這題？又或者，疑心病很重的你，發現以上的言論都只是我們的心理戰，其實這題是真正的簡單題，但卻用這樣的戰術來不戰而屈人之兵，讓你不敢動手寫這題？

既然是簡單題，那我們當然是要求你的程式要寫得夠簡單。因為不是英文的字母太不簡單了，所以你的程式碼只能包含 `ascii` 字元。再來，對於 `C` 和 `C++` 的每個保留字 (keyword)，你最多只能使用一次，不然你的程式就太複雜了。另外，`macro` 也會讓你的程式複雜到難以閱讀，因此我們當然也不允許你使用 `#define` 這個複雜的東西。精準地來說，你上傳的程式碼必須同時滿足下列所有條件：

- 不可以出現非 `ascii` 字元。
- 不可以出現 `define` 這個字串。
- 以下每個字串最多只能出現一次：

`requires, alignas, break, enum, catch, false, char, long, int, co_await,`

`concept, co_yield, public, goto, synchronized, sizeof, bool, restrict,`

`explicit, and_eq, private, nullptr, throw, reflexpr, extern, case, double,`

`export, friend, noexcept, alignof, constexpr, not_eq, auto, xor_eq, template,`

`typeid, atomic_commit, void, for, compl, operator, volatile, virtual, asm,`

`bitand, thread_local, union, continue, delete, else, atomic_cancel, using,`

`struct, float, dynamic_cast, new, constexpr, decltype, unsigned, while,`

`try, mutable, namespace, static_assert, register, typename, true, return,`

`bitor, inline, switch, constexpr, default, typedef, protected, short, class,`

`if, this, const_cast`

Input

輸入首行有兩個正整數 V, r, m ，代表這棵 perfect binary tree 的節點編號從 1 到 $V - 1$ ，且根節點的編號是 $r \circ m$ 會決定你要用哪種方式輸出（詳情請見下一個 section）。

接下來會有 $V - 2$ 行，每行會有兩個正整數 a_i, b_i ，代表 a_i 是 b_i 的父節點。

- $V \in \{2^n | n \in \mathbb{N} \wedge 1 \leq n \leq 10\}$
- $1 \leq r \leq V - 1$

- $1 \leq a_i \leq V - 1$
- $1 \leq b_i \leq V - 1$
- $m \in \{0, 1, 2\}$

Output

請輸出一行 $V - 1$ 個用空白隔開的正整數，用 prefix, infix, postfix 的其中一種方式將整棵樹表達出來。

請注意，如果一個節點有兩個子節點，則編號小的節點會在編號大的節點的左邊。

- 若 $m = 0$ ，請用 prefix 輸出。prefix 的意思是對於每個節點，請用「中左右」的方式遍歷。
- 若 $m = 1$ ，請用 infix 輸出。infix 的意思是對於每個節點，請用「左中右」的方式遍歷。
- 若 $m = 2$ ，請用 postfix 輸出。postfix 的意思是對於每個節點，請用「左右中」的方式遍歷。

注意，本題輸出為嚴格比對，不可以有任何行尾空白或多餘的換行！（行內多餘的空白當然也不可以）。因為這題已經很簡單了，所以我們會嚴格一點，還請見諒 (*◡◡*)

Sample 1

Input	Output
8 1 1 4 3 1 4 1 7 7 5 4 2 7 6	2 4 3 1 5 7 6

配分

在一個子任務的「測試資料範圍」的敘述中，如果存在沒有提到範圍的變數，則此變數的範圍為 Input 所描述的範圍。注意：本題非聯集給分，意即分數取自最高分那次，而非每次分數的聯集。

子任務編號	子任務配分	測試資料範圍
1	0%	範例測試資料
2	33%	$m = 0$
3	34%	$m = 1$
4	33%	$m = 2$

Hint

這題真的很簡單啦，先寫這題啦！