D. DFS 與大波路巧克力

Description

小 B 參加茲迅芝崖營隊，營隊裡面有一份作業是完成一個在二元樹上 DFS 的填空題，就可以吃到大波路巧克力。作為世界冠軍等級的資訊高手，他超喜歡大波路巧克力。為了心心念念的大波路巧克力，他速速寫下了以下程式碼。

```
1 void dfs (node* root) {
2     if (root != nullptr) {
3         cout << root->id << "\n";
4         dfs(root->l);
5         dfs(root->r);
6     }
7 }
```

然後上傳之後 …… 小 B 沒有獲得大波路巧克力。反而，獲得了人生第一次的 MLE !?

你或許會想說：「小 B 根本沒有宣告任何變數啊，為什麼會 MLE ?」請回想 stack frame 的作業，遞迴會使用 system stack，因此小 B 貨真價實的拿到了 MLE。

在一番檢測之後，小 B 發現他的 DFS 函數只能宣告一個 int，再用更多空間就會 MLE，理所當然也不能用遞迴。

小 B 當然知道要如何只用一個 int 完成這題，但是他現在正忙著跟出題人員吵架，而你只想默默的拿到這題的分數，請在一個 int 以內完成 DFS 吧。

Implementation details

這題是一題填空題。若上傳的程式碼不符合要求就會獲得 WA，在此前提之下輸出當然也要是對的（廢話，不然就不用寫任何東西了）。

以下是 sample.cpp，你可以在 CMS → SPROUT-2023-ALGO-B-CONTEST-D → Statement → Attachments 下找到他。

```cpp
#include <iostream>
using namespace std;
const int N = 1e6 + 10;
void dfs (node* root) {
  int tmp;
  /*DO NOT MODIFY ABOVE*/
  // Write down your code here, remember not to declare any other variable and do recursion.
  /*DO NOT MODIFY BELOW*/
}
int e[N][3];
struct node {
  node *l, *r, *p;
  int id;
  node (node *p, int id): l(nullptr), r(nullptr), p(p), id(id) {}
};
node* build (int id, node* p) {
  if (id == 0) {
    return nullptr;
  }
  node* ret = new node(p, id);
  ret->l = build(e[id][1], ret);
  ret->r = build(e[id][2], ret);
  return ret;
}
int main () {
  int m, root;
```

```
27   cin >> m >> root;
28   for (int i = 0; i < m; i++) {
29     int a, b, c;
30     cin >> a >> b >> c;
31     e[a][c] = b;
32     e[b][0] = a;
33   }
34   dfs(build(root, nullptr));
35 }
```

第 6 行的 /* DO NOT MODIFY ABOVE */ 以上和第 8 行的 /* DO NOT MODIFY BELOW */ 以下都是不能修改的。身為臺灣人，以上和以下是包含的，所以那兩行註解也不能改。具體來說，如果你上傳的程式碼前 6 行和 sample.cpp 的前 6 行不同，或是後 28 行和 sample.cpp 的後 28 行不同，都會被判定為 WA。

同時，為了確保你只使用了一個 int，因此如果在 /* DO NOT MODIFY ABOVE */ 和 /* DO NOT MODIFY BELOW */ 中間的範圍內出現了任何變數宣告和遞迴，都會被判定為 WA。

同時，為了避免你使用奇怪的方法繞過檢查），在 /* DO NOT MODIFY ABOVE */ 和 /* DO NOT MODIFY BELOW */ 中間的範圍不能使用任何 #、typedef 或是組合語言（asm 指令），否則會被判定為 WA。

理所當然，include 其他 library 也會動用到額外的記憶體，為了方便起見（並防止你透過奇怪的方法繞過檢查），在 /* DO NOT MODIFY ABOVE */ 和 /* DO NOT MODIFY BELOW */ 中間的範圍不能使用任何 # 或是組合語言（asm 指令），否則會被判定為 WA。

如果你的程式碼符合以上的規範，那他還必須要能輸出一棵二元樹的前序才能獲得 AC。

以下的 Input 和 Output 是針對茲迅芝崖作業的原題，他可以幫助你更了解 sample.cpp 的運作，同時你或許需要裡面的線索。理論上你即使不看也可以完成這題。

Input

第一行包含兩個變數 m 和 $root$，代表這棵二元樹的邊數以及根節點的編號。

接下來有 m 行，每行有三個變數 a, b, c。如果 $c = 1$，代表 b 是 a 的左子節點；如果 $c = 2$，代表 b 是 a 的右子節點。

- $1 \leq m \leq 10^6$。
- $1 \leq root, a, b \leq 10^6$。
- $c \in \{1, 2\}$

- 保證輸入構成一棵以 $root$ 為根節點的二元樹。

Output

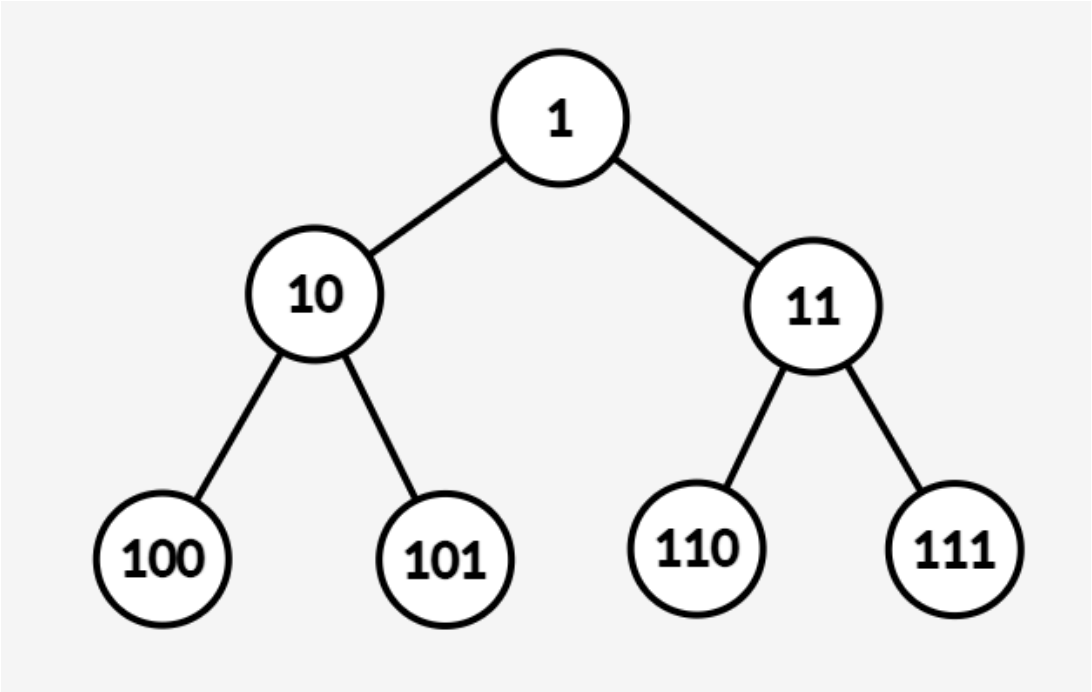
輸出 $m+1$ 行，代表這棵二元樹的前序。

Sample

Input	Output
6 1 1 10 1 1 11 2 10 100 1 10 101 2 11 110 1 11 111 2	1 10 100 101 11 110 111

Hint 1

Sample 的輸入所代表的二元樹如下：



Hint 2

具體來說，在 /* DO NOT MODIFY ABOVE */ 和 /* DO NOT MODIFY BELOW */ 中間的範圍如果出現以下關鍵字則會被判定為 WA。

- asm
- goto
- typedef
- delete、new
- auto、bool、class、const、double、enum、long、namespace、short、signed、struct、template、unsigned、using、void、byte
- char、char8_t、char16_t、char32_t
- float、__float128
- int、int8_t、int16_t、int32_t、int64_t、__int128、__int128_t、intmax_t、intptr_t
- uint8_t、uint16_t、uint32_t、uint64_t、uintmax_t、uintptr_t
- size_t、ssize_t、fpos_t、va_list、ptrdiff_t、nullptr_t、max_align_t、off_t、offsetof、type_info、bad_typeid、bad_cast、type_index、mode_t、FILE、mbstate_t、dev_t、id_t、uid_t、pid_t、stack_t、rlim_t、mcontext_t、ucontext_t
- int_fast8_t、int_fast16_t、int_fast32_t、int_fast64_t
- int_least8_t、int_least16_t、int_least32_t、int_least64_t
- uint_fast8_t、uint_fast16_t、uint_fast32_t、uint_fast64_t
- uint_least8_t、uint_least16_t、uint_least32_t、uint_least64_t
- node、build、dfs、main
- div_t、ldiv_t、lldiv_t
- time_t、tm、clock_t、clockid_t、timespec、timeval
- wchar_t、wint_t、wctrans_t、wctype_t
- pthread_t、pthread_attr_t、pthread_mutex_t、pthread_mutexattr_t、pthread_cond_t、pthread_condattr_t、pthread_rwlock_t、pthread_rwlockattr_t、pthread_override_t、pthread_once_t、pthread_key_t

- sig_atomic_t、sigset_t、sigval、sigevent、siginfo_t、sigaction
- string、wstring

如果你認為你要使用的某個關鍵字明明不會用到額外記憶體卻被禁用了，歡迎透過 CMS 進行提問，並盡量附上你認為他不會用到額外記憶體的說明。(當然 # 、asm 和 typedef 是特殊狀況，請不要嘗試為他們開脫。)

要注意的是，如果於比賽中發現了意想不到的變數宣告方式，出題者有權利調整 blacklist 並 rejudge。

Hint 3

真正的 build 程式碼被隱藏起來了，畢竟真的 build 是沒有遞迴的，改一改就能得到答案了，怎麼可能這麼簡單呢:D

Hint 4

首殺有機會獲得大波路巧克力喔 >< (限 204 實體領取 !)