

試場規則

違規事項

- 行動裝置未置於教室外、教室前後、監考老師桌上、個人電腦主機上，經監考老師發現。
- 於考試期間使用行動裝置。
- 配戴具通訊功能的穿戴裝置。
- 以任何方式使其他考生無法正常使用系統。
- 考試期間與監考老師以外之人交談。

上述行為被發現，且屢勸不聽者，將登記於考生簽到表，並在賽後系統測試時將總成績 $\times 0.0001$ 並四捨五入至個位。

賽制

- 本次競賽採 **OI** 制度，有部分分，無罰時，並取每筆提交的子題聯集為總分。
 - 例如：某題共有兩筆提交，第一筆通過子測資 $\{1, 2\}$ 、第二筆通過子測資 $\{2, 3\}$ ，則總分為第 $\{1, 2, 3\}$ 筆子測資的分數相加。
- 本次為封板賽，記分板將在比賽結束後公布。
- 競賽結束後會做一次 **System test**(系統測試)，所有成績以其為準。
- 提交的冷卻時間(CD time)為 15 秒，最後 30 分鐘不在此限。
- 對於每一題，使用者最多可以進行 100 筆提交。

系統使用說明

- 系統連結: <http://192.168.81.242>
- 競賽將在 **2025/07/18 9:00 A.M.** 開始，使用者有十分鐘的時間閱讀試場規則，確認讀畢後請按下系統上的開始鈕，以免影響競賽時間。
- 本次競賽時長共 **180** 分鐘。
- 最晚進場時間 **2025/07/18 10:00 A.M.**。
- 最早離場時間 **2025/07/18 10:00 A.M.**。
- 總題本在第**A**題的題目敘述頁面中。
- 使用者允許使用 **C/C++11/C++17** 提交程式碼。
- 若結果為 **Execution timed out (wall clock limit exceeded)**，則表示系統因為某筆提交繁忙中，請檢查你的程式碼是否有可能超過執行時間，並稍後再試。
- 對於每筆提交，請確認副檔名符合系統要求，詳見系統頁面。
- 如有題目問題，請使用系統提供的訊息詢問功能提問。
- 如有其他問題，如：上廁所、需要計算紙、系統使用問題等，請直接舉手向監考老師發問。

資源

- 賽後我們將會在一天內 **彰中資訊社群** 及 **HARC Discord** 中公告本次題解、總成績。
- 競賽後將擇期在 **HARC Discord** 上進行直播題解。
- 網址：
 - [彰中資訊社群](#)
 - [本次專案](#)
 - [HARC Discord](#)
 - [彰中資訊社Discord](#)

A. 庭院深深深幾許 *Garden*

time limit 1s
memory limit 256MB

Statement

眾所周知，將「庭院深深深幾許」重新排列後共有 $\frac{7!}{1!1!3!1!1!}$ 種不互相重複的可能組合。

而徐氏數學裏頭有一題：「現在給你 n 個字元，字元 **ASCII** 值為 a_i ， $1 \leq i \leq n$ 的字元會重複 c_i ， $1 \leq i \leq n$ 次，請你求出共有幾種不互相重複的可能組合」

現在 **ysh** 因為數學太弱，被 **yy** 出的徐氏數學難住了，於是請你 - - **C++電神** 寫出一個程式來幫助 **ysh** 計算。

Input

n
 $a_1\ c_1$
 $a_2\ c_2$
...
 $a_n\ c_n$

Output

$Ans \bmod (10^9 + 7)$

請輸出答案除以 $10^9 + 7$ 的餘數

Sample Input

```
5
1 1
2 1
3 3
4 1
5 1
```

Sample Output

```
840
```

Note

- $0 \leq n \leq 10^5$
- $1 \leq c_i \leq 10^6$, $1 \leq i \leq n$
- $0 \leq a_i \leq n$, $1 \leq i \leq n$

- $\sum_{k=1}^n c_k \leq 10^6$

Subtask

- **subtask1:** 10% $n \leq 10$, $\sum_{k=1}^n c_k \leq 20$
- **subtask2:** 20% $n \leq 100$, $\sum_{k=1}^n c_k \leq 1000$
- **subtask3:** 30% $n \leq 10^4$, $\sum_{k=1}^n c_k \leq 10^6$
- **subtask4:** 40% **As statement**

B. hamster冒險 *Hamster's Tree Adventure*

time limit 1s

memory limit 256MB

Statement

Hamster 是一隻聰明又勤奮的小倉鼠，牠住在一棵 **巨大二元樹形的樹林**中，樹林結構如下規則：

- 樹的根是編號 1 的樹洞。
- 每個樹洞編號為 i 符合以下條件：
 - 左子節點為 $2i$ (若 $2i \leq n$) 。
 - 右子節點為 $2i + 1$ (若 $2i + 1 \leq n$) 。

每個樹洞裡都有寶藏，其價值為 w_i 。

現在 Hamster 想請你幫忙回答 m 個問題。

每個問題給你一個樹洞編號 u ，請輸出以 u 為根的底下子樹內所有樹洞的寶藏總價值 (包括 u 自己) 。

Input

$n\ m$

w_1, w_2, \dots, w_n

q_1, q_2, \dots, q_m

第一行包含兩個整數 n, m ($1 \leq n, m \leq 10^5$)：

- n 是樹洞數 (節點數)
- m 是詢問次數

第二行包含 n 個整數 w_1, w_2, \dots, w_n ($-10^9 \leq w_i \leq 10^9$)：

第 i 個數字表示節點 i 的寶藏價值。

第三行包含 m 個整數 q_1, q_2, \dots, q_m ($1 \leq q_i \leq n$)：

表示 Hamster 要詢問的節點編號。

Output

Ans

對於每個詢問，輸出一行，表示以 q_i 為根的子樹內所有寶藏總價值 *Ans*

Note



編號1樹洞底下總寶藏 = $5+2+3+4+5+6+7 = 32$

Sample Input

```
7 3
1 2 3 4 5 6 7
1 2 3
```

Sample Output

```
28
11
16
```

Subtask

- **subtask1:** $10\% \cdot n, m \leq 100$
- **subtask2:** $90\% \cdot n, m \leq 10^5$

C. 彈幕遊戲 *Game*

time limit 1s

memory limit 256MB

Statement

Frisk最近愛上了一款彈幕遊戲，玩法是要在 5×5 的範圍內(座標為0~4)閃躲從四面八方飛來的子彈，角色與子彈大小皆為一格，角色一開始在中心的位置 $(2, 2)$ ，接下來角色先移動一格(上下左右，不包含斜向，可不移動)，子彈隨後移動一格(上下左右，可能包含斜向)，以此類推，因為Frisk對no hit的堅持，因此只要角色被子彈打中一次就算失敗。

而他的好朋友sans得知後，自訂了一個關卡給Frisk挑戰，Frisk毫不猶豫地接下了挑戰，但經多次嘗試後都過不了關，Frisk認為此關卡是無解的，sans則認為這是skill issue，你能寫程式幫他們判斷關卡是否有解嗎？

Input

第一行為 N ，表示此關的子彈數量

接下來有 N 行，每行包含 T, X, Y

T 代表子彈的移動模式

X, Y 代表關卡開始時子彈的座位置

T 所代表的子彈移動模式

T	X	Y
0	+	=
1	+	+
2	=	+
3	-	+
4	-	=
5	-	-
6	=	-
7	+	-

保證每顆子彈路徑必經 5×5 的範圍內

Output

若關卡有解，則輸出1

否則輸出0

Sample Input 1

```
8
0 -1 0
0 -1 1
```

```
0 -1 2
0 -1 3
0 -6 1
0 -6 2
0 -6 3
0 -6 4
```

Sample Output 1

```
1
```

Sample Input 2

```
8
4 5 0
4 5 1
4 5 2
4 5 3
4 9 1
4 9 2
4 9 3
4 9 4
```

Sample Output 2

```
0
```

Sample Input 3

```
9
6 0 5
6 1 5
6 2 5
5 4 4
4 5 2
3 4 0
2 0 -1
2 1 -1
2 2 -1
```

Sample Output 3

0

Sample Input 4

```
25
2 0 -1
2 1 -1
2 2 -1
6 3 5
6 4 5
2 0 -4
2 1 -4
2 2 -4
6 3 8
6 4 8
2 0 -7
2 1 -7
2 2 -7
6 3 11
6 4 11
2 0 -10
2 1 -10
2 2 -10
6 3 14
6 4 14
2 0 -13
2 1 -13
2 2 -13
6 3 17
6 4 17
```

Sample Output 4

1

Note

- $N \leq 8 \times 10^3$
- $0 \leq T \leq 7$
- $|X|, |Y| \leq 6000$
- $N, T, X, Y \in \mathbb{N}$

Subtask

- **task:** As statement

D. 紅黑樹 *Red Black Tree*

time limit 2s

memory limit 256MB

Statement

MelonWalker 有一棵包含 n 個節點的紅黑樹，節點編號從 1 到 n 。

節點 1 是樹的根節點，每個節點的顏色可以是紅色 (編號 0) 或黑色 (編號 1)。

每個節點 i 有一個顏色值 c_i ，其值為 0 或 1。

你可以執行以下操作任意次數：

- 選擇一個節點 v 。
- 反轉以 v 為根的子樹中所有節點的顏色 (將 0 變為 1，將 1 變為 0)。

你的目標是讓樹中所有節點的顏色都變為紅色 0。請找出達成此目標所需的最少操作次數。

在第一個範例中，初始顏色為 $[1, 1, 1]$ 。

對節點 1 執行一次操作，所有節點的顏色會反轉為 $[0, 0, 0]$ 。

在第二個範例中，初始顏色為 $[1, 0, 1, 0]$ 。

首先，對節點 1 執行一次操作，顏色變為 $[0, 1, 0, 1]$ 。

接著，對節點 2 執行一次操作，顏色變為 $[0, 0, 0, 0]$ 。

Input

```
n
c1, c2, ..., cn
u1 v1
u2 v2
⋮
un-1 vn-1
```

輸入的第一行包含一個整數 n ，表示樹中的節點數量。

第二行包含 n 個整數，表示節點 $1, 2, \dots, n$ 的初始顏色。

接下來的 $n - 1$ 行描述樹的邊。每行包含兩個整數 u 和 v ，表示節點 u 和節點 v 之間有一條邊。保證給定的邊構成一棵樹，且節點 1 是根節點。

Output

輸出一個整數，表示使所有節點顏色變為 0 所需的最少操作次數。

Sample Input 1

```
3
1 1 1
1 2
1 3
```

Sample Output 1

```
1
```

Sample Input 2

```
4
1 0 1 0
1 2
1 3
2 4
```

Sample Output 2

```
2
```

Note

- $1 \leq n \leq 10^6$
- $1 \leq u, v \leq n, u \neq v$
- $c_i \in \{0, 1\}$
- 所有輸入輸出皆為整數

subtask

- **subtask1**: 40% $n \leq 100$
- **subtask2**: 60% *As statement*

E. 區間操作 *Segment Operations*

time limit 1s

memory limit 256MB

Statement

給定一個長度為 n 的整數序列 a_1, a_2, \dots, a_n ，你需要處理以下兩種操作共 q 次：

1. 區間修改 (**Segment Modify**) :

對於給定的兩個索引 l 和 r ，將區間 $[l, r]$ 中的所有元素 a_i 替換為 $\lfloor \frac{a_i}{2} \rfloor$ 。

2. 單點查詢 (**Point Query**) :

給定一個索引 i ，輸出當前序列中 a_i 的值。

Input

n

$a_1 \ a_2 \ \dots \ a_n$

q

$query_1$

$query_2$

\vdots

$query_q$

第 1 行有一個整數 n ，表示序列長度。

第 2 行有 n 個整數，表示初始序列。

第 3 行有一個整數 q ，表示操作筆數。

接下來 q 行：每行一筆操作，格式如下：

- $1 \ l \ r$ ：表示進行一次區間向下取整操作。
- $2 \ i$ ：表示查詢第 i 個元素的值。

Output

Ans_1

Ans_2

\vdots

對於每個單點查詢操作，輸出對應的元素值，每個值佔一行。

Sample Input

```
5
5 8 3 9 2
3
1 2 4
2 4
2 1
```

Sample Output

4

5

Note

- $1 \leq n, q \leq 10^5$
- $1 \leq a_i \leq 10^9$
- $1 \leq l \leq r \leq n$
- 所有輸入皆為正整數

Subtask

- **subtask1:** 30% $n \times q \leq 10^6$
- **subtask2:** 70% *As statement*

F. 拔樹 *Weed*

time limit 3s

memory limit 256MB

Statement

ysh 有一棵樹，奇高無比，高到白天時會把所有陽光給擋住，對此 **ysh** 感到相當困擾，於是 **ysh** 打算對這棵樹進行一些操作來讓他獲得陽光。

對於一棵樹，有 n 個節點、 $n - 1$ 條邊，而從根節點 1 到任一節點只有一條路徑，長 h_i , $1 \leq i \leq n$

而 **ysh** 會挑任一個邊 $edge_i$ 進行 **拔樹**，並更改其長度(沒錯，鋸木頭的機器不只能鋸樹也可以揠苗助長)。

但在 **拔樹** 的同時，**ysh** 也相當好奇節點 i 目前的高度，也就是 h_i 為何，好讓他精確地控制進度。

於是你必須寫一個程式幫他完成這件事:D

Input

```
n
a1 b1 c1
a2 b2 c2
...
an-1 bn-1 cn-1
k
op1 ...
op2 ...
...
opk
```

輸入會有若干行，第一行輸入 n, m ，接著會有 m 行描述這棵樹一開始的樣子，每一行共有三個數，分別是 a_i, b_i, c_i , $1 \leq i \leq n$ ，代表 a_i 與 b_i 之間有一條邊，長度為 c_i

接著會有一個數字 k ，代表操作筆數。

而後會有 k 行描述操作，每行的第一個數字為 op_i , $1 \leq i \leq k$

而 op_i 僅有可能為 0 或 1，其中

- $op_i = 0$ 時，後面會接著一個數字 x_i , $1 \leq i \leq k$ ，代表 **ysh** 想知道點 x_i 距離地面的長度 h_i ，而你必須回答他:D
- $op_i = 1$ 時，後面會接著三個數字，分別為 l_i, r_i, x_i , $1 \leq i \leq k$ ，代表 **ysh** 使用了神奇鋸木器將節點 l_i 到節點 r_i 的邊之距離改為了 x_i

Output

```
Anso1
Anso2
...
Anso|o|
```

其中 $o \equiv [x | op_x = 1]$

Sample Input

```
6
1 5 3
2 6 2
3 4 4
3 5 3
5 6 5
3
1 1 5 1
0 4
0 2
```

Sample Output

```
8
8
```

Note

- $1 \leq n \leq 2 \times 10^5$
- $k \leq 10^6$
- $0 \leq c_i \leq 10^9, \forall 1 \leq i \leq n - 1$
- $0 \leq x_i \leq 10^9, \forall (1 \leq i \leq n \wedge op_i = 1)$
- $1 \leq x_i \leq n, \forall (1 \leq i \leq n \wedge op_i = 0)$
- $1 \leq a_i, b_i \leq n, \forall 1 \leq i \leq n - 1$

Subtask

- **subtask1:** 10% $n \leq 10$
- **subtask2:** 10% $k \leq 100$
- **subtask3:** 20% $n \leq 1000, k \leq 1000$
- **subtask4:** 30% 圖是一直線
- **subtask5:** 30% **As statement**