



3. ความสูงต้นไม้ (High)

งานของคุณ

ให้เขียนโปรแกรมที่มีประสิทธิภาพ เพื่อหาความสูงของ Binary Tree ที่กำหนดให้

1

/ \

10 39

/

5

กำหนดให้ความสูงของ Tree เท่ากับ 3 ถ้า root เป็น NULL (empty Tree) ความสูงจะเท่ากับ 0

ข้อมูลนำเข้า

แต่ละบรรทัดจะเป็น integer n ตามลำดับของ Level Order ซึ่งมีความเป็นไปได้สามแบบดังนี้

- $n \geq 0$, เป็น index ของ node
- $n = -1$, เป็น NULL
- $n = -2$ เป็นการสิ้นสุด input



ตัวอย่าง

input	Graph
1	หมายเหตุ: ทั้งสองกราฟเป็นกราฟเดียวกัน
2	
3	
-1	
-1	
4	
6	
-1	
5	
-1	
-1	
7	
-1	
-2	

ข้อมูลส่งออก

มีบรรทัดเดียว ความสูงของ Tree

ตัวอย่างที่ 1

input	Output
1	5
2	
3	
-1	
-1	
4	
6	
-1	
5	
-1	
-1	
7	
-1	
-2	

ตัวอย่างที่ 2

input	Output
1	2
2	
3	
-2	



ตัวอย่างที่ 3

input	Output
2	3
-1	
1	
3	
-1	
-2	

ตัวอย่างที่ 4

input	Output
10	3
20	
30	
40	
60	
-1	
-1	
-2	

ข้อกำหนด $1 \leq \text{จำนวนบรรทัด} \leq 10^5$



ข้อกำหนด

หัวข้อ	เงื่อนไข
ข้อมูลนำเข้า	Standard Input (คีย์บอร์ด)
ข้อมูลส่งออก	Standard Output (จอภาพ)
ระยะเวลาสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	1 วินาที
หน่วยความจำสูงสุดที่ใช้ในการประมวลผล ต่อชุดทดสอบหนึ่งชุด	32 MB
จำนวนชุดทดสอบ (โปรแกรมประมวลผลครั้งละชุดทดสอบ)	10
คะแนนสำหรับชุดทดสอบแต่ละชุด	10
คะแนนสูงสุดของโจทย์	100
เงื่อนไขการการรับโปรแกรม	โปรแกรมจะต้องประมวลผลข้อมูลตามตัวอย่างที่ให้มาได้

ข้อมูลคำสั่งเพิ่มเติม

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C ให้ระบุชื่อแฟ้มข้อมูล high.c และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
```

```
TASK: high
```

```
LANG: C
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: WU
```

```
*/
```

สำหรับผู้เข้าแข่งขันที่เขียนโปรแกรมด้วยภาษา C++ ให้ระบุชื่อแฟ้มข้อมูล high.cpp และระบุส่วนหัวของโปรแกรกดังนี้

```
/*
```

```
TASK: high
```

```
LANG: C++
```

```
AUTHOR: YourName YourLastName
```

```
CENTER: WU
```

```
*/
```