Lab 6 B

The Mystery of the Missing Clue 🕵️‍♂️

Detective Narain needs your help! He's following a trail of clues, each connected in a sequence (a linked list). But to solve the case, he must find the third-to-last clue in the sequence. Can you crack the case?

Your Task:

- Create a linked list based on the clues (numbers) provided.
- Find the third-to-last clue in the sequence.

If the sequence is too short (fewer than three clues), you must report: "The trail's gone cold!"

Input Format:

- The first line contains an integer **N**, representing the number of elements.
- The second line contains an integer array **clues[]**, representing the clues.

Output Format:

Print the third-to-last clue.**3rd last clue**.

Constraints:

- 0 ≤ **N** ≤ 10e6
- 0 ≤ **clues** ≤ 10e18

Example:

Input 1:

```
5
10 22 35 41 15
```

Copy

Output 1:

```
35
```

Copy

Help Narain solve the case before the trail goes cold! 🔍

Problem 2: Group Bills

You and your friends recently went on a group trip. At the end of the trip you want to add up the total expenditure. During the trip, each person maintained their bills as a sorted linked list. You want to merge everyones linked list while maintaining it in sorted order.

Input Format:

`n` (Total number of people in the group `1<= n <= 1e3`)
`k_1` (The number of bills of the i_th person.)
`b_1 b_2 ... b_k_1` (The value of the k_1 bills paid by person 1. The bills are in sorted order `b_1 <= b_2 <= ... <= b_k_1`)
`k_2` (The number of bills of the i_th person.)
`b_1 b_2 ... b_k_2` (The value of the k_2 bills paid by person 2. The bills are in sorted order `b_1 <= b_2 <= ... <= b_k_2`)
.
.
.
`k_n` (The number of bills of the i_th person.)
`b_1 b_2 ... b_k_n` (The value of the k_n bills paid by person n. The bills are in sorted order `b_1 <= b_2 <= ... <= b_k_n`)

Each k_i satisfies `1 <= k_i <= 1e4`
And `N = k_1 + k_2 + k_3 + ... + k_n <= 1e5`
Each b_i satisfies `1<= b_i <= 1e9`

Output Format:

`o_1 o_2 ... o_N` (A sorted list of all the bills. `o_1 <= o_2 <= o_N`)

Input 1:

```
3
2
4 6
```

Copy

```
3
1 4 7
4
9 10 11 14
```

Output 1:

```
1 4 4 6 7 9 10 11 14
```

Copy

Problem 3: Highest Gap Between Peaks in a Linked List

You are given a sequence of integers stored in a linked list, and you need to find the highest gap between consecutive peak nodes. A node is considered a peak if it is greater than both its immediate neighbor elements. For the first node, assume its previous neighbor is -∞ (negative infinity), and for the last node, assume its next neighbor is -∞ as well.

You need to:

- Traverse the list to identify the peak nodes.
- Calculate the gaps between consecutive peak nodes.
- Print the highest gap between these peaks. If number of peaks is less than two peaks, Print -1.

Input Format:

- A sequence of integers where the last element is -1, which indicates the end of the input. - The sequence will be entered one number at a time and stored in a linked list.

Constraints:

- $1 \leq$ `Number of nodes in a LinkedList` $\leq 10^6$

Output Format:

- Print the highest gap between consecutive peak nodes. - If there are fewer than two peaks, print -1

Input 1:

```
1 3 5 2 4 1 -1
```
Copy

Output 1:

```
2
```
Copy

Explanation:

- Peaks: 5 (index 2), 4 (index 4).
- Gaps between consecutive peaks:
  - Gap between 5 and 4: 4 - 2 = 2
- The highest gap is 2.

Input 2:

```
1 2 3 4 -1
```
Copy

Output 2:

```
-1
```
Copy

Explanation:

- There is only one peak in this sequence, so the result is -1.

Input 3:

```
4 -1
```
Copy

Output 3:

```
-1
```
Copy

Explanation:

- There is only one peak in this sequence, so the result is -1.