Task Management System

Imagine you're developing a task management system for a company that prioritizes tasks based on urgency. A **task** is defined by a pair (**task_id**, **urgency**). Your goal is to efficiently manage these tasks using a heap structure H so that the task with **lower urgency is handled first**.Each task has a unique `task_id` and urgency, ensuring no two tasks have the same urgency level. You are assigned to perform the following:

Tasks:

(a) **AddTask(H, task)**: Insert the given task with unique `task_id` and unique `urgency` (both positive integers $\in [1, 10^6]$) into the Heap H. After insertion, perform the necessary operations to maintain the heap property.

(b) **UpdateUrgency(H, `task_id`, `newValue`)**: Update the urgency of the given task with `task_id` to the `newValue`. Perform the necessary operations to maintain the heap property.

(c) **DeleteUrgency(H)**: Delete the task with the lowest urgency from H and print the corresponding `task_id`. After deletion, perform the necessary operations to maintain the heap property.

(d) **Display(H)**: Display the current state of H as a space-separated list of `task_ids` in level-order traversal.

(e) **PrintKthLevelMin(H, `level`)**: Print the `task_id` with the lowest urgency at the k-th level of H. If the k-th level does not exist, print `-1`.

Assume that all task_ids and urgencies are unique at all times.

Input Format:

Each line contains a character from {'i', 'u', 'd', 'e', 'p', 'q'} followed by zero or more positive integers.

- Character `'i'` is followed by two positive integers: `task_id` and `urgency`. Perform the **AddTask(H, task)** operation.
- Character `'u'` is followed by two positive integers `task_id` and `newValue`. Perform the **UpdateUrgency(H, `task_id`, `newValue`)** operation.
- Character `'d'` performs the **Display(H)** operation.
- Character `'e'` performs the **DeleteUrgency(H)** operation.
- Character `'p'` is followed by a positive integer `level`. Perform the **PrintKthLevelMin(H, `level`)** operation.
- Character `'q'` is to terminate the sequence of operations.

Output Format:

The output (if any) of each command should be printed on a separate line. However, no output is printed for `'i'` and `'q'`.

- For Option `'u'`: Print `-1` if the node is not found else print `task_id`.
- For Option `'e'`: Print `-1` if the heap is empty else print `task_id`.
- For Option `'d'`: Print H as a space-separated list of `task_ids` in level-order traversal. If H is empty, print `-1`.
- For Option `'p'`: Print the `task_id`.

Sample Test Cases

Input 1:

```
i 5 60
i 6 25
d
e
i 7 85
p 1
i 8 125
i 9 10
u 8 5
d
q
```

Output 1:

```
6 5
6
7
8
8 5 9 7
```

Input 2:

```
i 31 70
i 32 85
i 33 50
d
u 31 45
i 34 95
i 35 30
```

p 2
u 35 110
d
e
q

Output 2:

33 32 31
31
32
35
31 32 33 34 35
31