File Submit Format: NIM_NAMA.c (Ex: 270220000_Andi.c)

Problem Statement

Create a C program that manages a playlist using an AVL Tree with linked list. The program should support the following operations:

1. Insertion: Implement a function that inserts a new integer value into the AVL tree. If the value already exists in the tree, do nothing.
2. Deletion: Implement a function that deletes an integer value from the AVL tree. If the value does not exist in the tree, do nothing.
3. Predecessor: Implement a function that finds the predecessor of a given integer value in the AVL tree. The predecessor of an integer x is the largest integer in the tree that is smaller than x. If no such integer exists, return -1.
4. Print: Implement a function that prints the AVL tree in a readable format.
5. Initially, the AVL Tree is empty.

Input:

1. The first line contains an integer N ($1 \leq N \leq 100$), representing the number of elements in the AVL tree.
2. The second line contains N space-separated integers, representing the elements of the AVL tree.
3. The third line contains an integer Q ($1 \leq Q \leq 100$), representing the number of queries.
4. Each of the next Q lines contains a query in the format "t x", where t is the type of query (either "I" for Insertion, "D" for Deletion, or "P" for Predecessor), and x is the integer value for the query.

Output:

- For each query of type "P", output the predecessor of the given value x. If no predecessor exists, output -1.

Example Input and Output

Input :	**Output :**
6 10 5 20 3 7 15 5 P 10 I 12 P 12 D 7 P 7	7 10 12 5

Explanation:

- For the first query, we're finding the predecessor of 10, which is 7.
- Then, we're inserting 12 into the AVL tree.
- Next, we're finding the predecessor of 12, which is 10.
- After that, we're deleting 7 from the AVL tree.
- Finally, we're finding the predecessor of 7, which is 5.