```c
#include <stdio.h>
#include <stdlib.h>

typedef struct Node{
    int value;
    struct Node * next;
}Node;

Node *head;
Node *tail;

// Add a new node at the tail
void enqueue(Node *newNode){
    tail->next = newNode;
    tail = tail->next;
}

void printList(Node *head){
    while(head != NULL){
        printf("%d", head->value);
        head = head->next;
        if(head != NULL)
            printf(" "); // No trailing spaces
    }
}

// Remove the head from the list and return it
Node* dequeue(){
    Node *temp = head;
    head = head->next;
    return temp;
}

int main(){
    int n, x, t;
    scanf("%d", &n);
    Node arr[n];
    for(int i = 0; i < n; i++){
        scanf("%d", &x);
        arr[i].value = x;
        arr[i].next = NULL;
    }
    scanf("%d", &x);
    head = &arr[x];
    for(int i = 0; i < n-1; i++){
        int x1, x2;
        scanf("%d%d", &x1, &x2);
        arr[x1].next = &arr[x2];
    }

    // Find the tail of the linked list
    Node *temp = head;
    while(temp->next != NULL)
        temp = temp->next;
    tail = temp;

    printList(head);
    printf("\n");
    scanf("%d", &t);
    while(t--){
        int x, y;
        scanf("%d %d", &x, &y);
        if(x == 2)
            printf("%d\n", dequeue()->value);
        else{
            Node *newNode = (Node*)malloc(sizeof(Node));
            newNode->value = y;
            enqueue(newNode);
        }
```

```
        printList(head);
        if(t > 0)
            printf("\n"); // No trailing newlines
    }
}
```