

Week 8: Assignment 8 - Question 2

Priority Queue using Linked List

In this question, a linked list is partially implemented where each element in the linked list is a structure of the following format:

```
struct node{
    int id;
    int priority;
    struct node *next;
};
```

The field **priority** is a positive integer, which denotes the priority of an element inside the list. The higher the value of integer in this field, the higher priority.

You have to complete the C code for performing the following operations in the linked list:

1. Create and return a node **e** with given **id** and value **val**
`struct node * create_node(int id, int val);`
2. Add an node **e** to the beginning of the **list**. Return the new list.
`struct node * append(struct node * list, struct node * e);`
3. Search for a node **e** with **id** inside the **list**. Return a pointer to **e** if found, else return NULL.
`struct node * search(struct node * list, int id);`
4. Change the value of an element with given id (if found), in the **list** to the new value **val**.
`void change_priority(struct node * list, int id, int val);`
5. Extract the element in the **list** with maximum priority. Return pointer to new list.
`struct node* extract_max(struct node * list);`

After `extract_max`, the element having the max priority is removed from the list. Extract max also prints the id of the removed element in the following format "Max: id".

Note: You can assume that the priority of each element in the list is unique.

Note: The code for manipulating the input as well as output is given to you. You only have to write code for the incomplete functions.

Input

A set of lines, each lines containing a character representing the operation and its inputs.

The operation can be one of the following:

- A <id> <val>
Add an node with id and val to the list, at the start of the list.
- C <id> <val>
Change the priority field of the element with id to val.
If an element with this id is not found, do nothing.
- S <id>
If an element with the id is in the list print the id and the priority and a newline.
Else, print the id and -1 and a newline.
- M
Extract the element in the **list** with maximum priority. Print the id of the element as "Max: id"
- E
End of input, exit from the program

Output

The output of search queries and extract max operations.

Sample input

```
-----
A 1 10
A 2 20
S 2
A 3 30
S 3
M
S 3
C 2 30
S 2
E
```

Sample Output

```
-----
2 20
3 30
Max: 3
3 -1
2 30
```

Explanation

- The list is initially empty
- Add an element 1 with value 10
`list : (1,10) -> NULL`
- Add an element 2 with value 20
`list : (2,20) -> (1,10) -> NULL`
- Search for element with id 2, print
`2 20`
- Add an element 3 with value 30
`list : (3,30) -> (2,20) -> (1,10) -> NULL`
- Extract Max prints
`Max: 3`
`list : (2,20) -> (1,10) -> NULL`
- Search for element with id 3, print
`3 -1`
- Change priority of 2 to 30
`list : (2,30) -> (1,10) -> NULL`
- Search for element with id 2, print
`2 30`
`End of input`