King Abdul Aziz University
EE-463
LAB #4
8-6-2023

OPERATING SYSTEMS

LAB #4

Omar alahamdi_1936683

The problem:

Before fixing the program, the program did not perform any deletions. Additionally, it would add a new node even when it was not required. As a result, we need to make modifications to the remove function.

The code shown in the following figure demonstrates an important improvement. To begin with, we need to prevent the creation of a new node if the marker is not the sole element in the list and if it is the first element. Instead of creating a meaningless new node and assigning it as the next one, we will update the head pointer to point to the next marker and subsequently delete the first marker. Consequently, we will replace the highlighted line with the code: "head_ = marker->next();".

```
// returns 0 on success, -1 on failure
int remove (const int &item_to_remove) {
  Node *marker = head_;
  Node *temp = 0;         // temp points to one behind as we iterate

  while (marker != 0) {
    if (marker->value() == item_to_remove) {
      if (temp == 0) {   // marker is the first element in the list
        if (marker->next() == 0) {
          head_ = 0;
          delete marker; // marker is the only element in the list
          marker = 0;
        } else {
          head_ = new Node(marker->value(), marker->next());
          delete marker;
          marker = 0;
        }
        return 0;
      } else {
        temp->next (marker->next());
        delete temp;
        temp = 0;
        return 0;
      }
    }
    marker = 0;  // reset the marker
    temp = marker;
    marker = marker->next();
  }
```

Furthermore, it is evident that we are currently assigning the next marker to the "temp next" and then deleting it, which introduces another problem. Once we have removed the current marker, we intend to use "temp" later and delete the marker itself, rather than the "temp" variable. Therefore, we need to replace the highlighted line with the code: "delete marker; marker = 0;".

In conclusion, we have already removed the highlighted line. Consequently, we now have the corrected code, which will be submitted along with the zip file. Additionally, a screenshot of the output is included below.

Ex1:

```
omar_alahmadi@lamp ~/Ex1$ ./maain1
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Destroying Node, 3 are in existence right now
3
2
1

Destroying Node, 2 are in existence right now
3
2

Destroying Node, 1 are in existence right now
3

Destroying Node, 0 are in existence right now
omar_alahmadi@lamp ~/Ex1$
```

Ex2:

```
omar_alahmadi@lamp ~/Ex2$ ./main2
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Destroying Node, 3 are in existence right now
4
3
1
```