King Abdulaziz University -Faculty of Engineering
Dept. of Electrical and Computer
Engineering

Lab 4

Name	**ID**
Nawaf Sami Alharbi	1936576

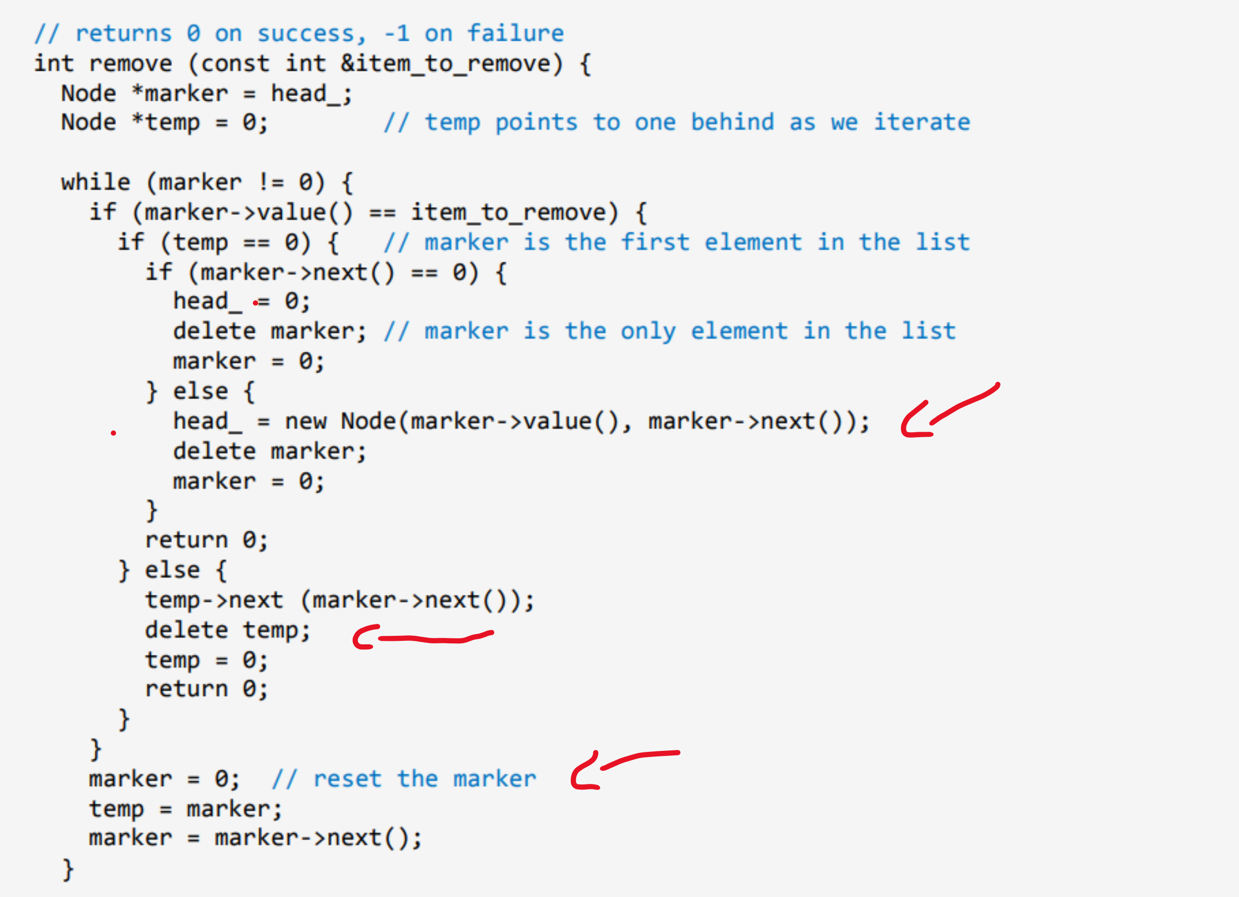
Activity 1:

The problem & its solution:

previously, the program was not deleting anything. Moreover, it was inserting a new node when it is not necessary.

In the given code, there is a section where we need to make a modification. Instead of creating a new node and assigning it as the next element when the marker is not the only element and the first element, we can simply update the head pointer to point to the next marker and delete the first marker. This change eliminates the need for creating a new node that doesn't serve any purpose. To achieve this, we can replace the highlighted yellow line with the following 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();
  }
```

In the provided code, there is another correction that needs to be made. Currently, we assign the next marker to the temp variable and delete it, but this creates a problem. We actually want to delete the marker itself and keep the temp variable for later use. To address this, we can replace the highlighted green line with the following code:

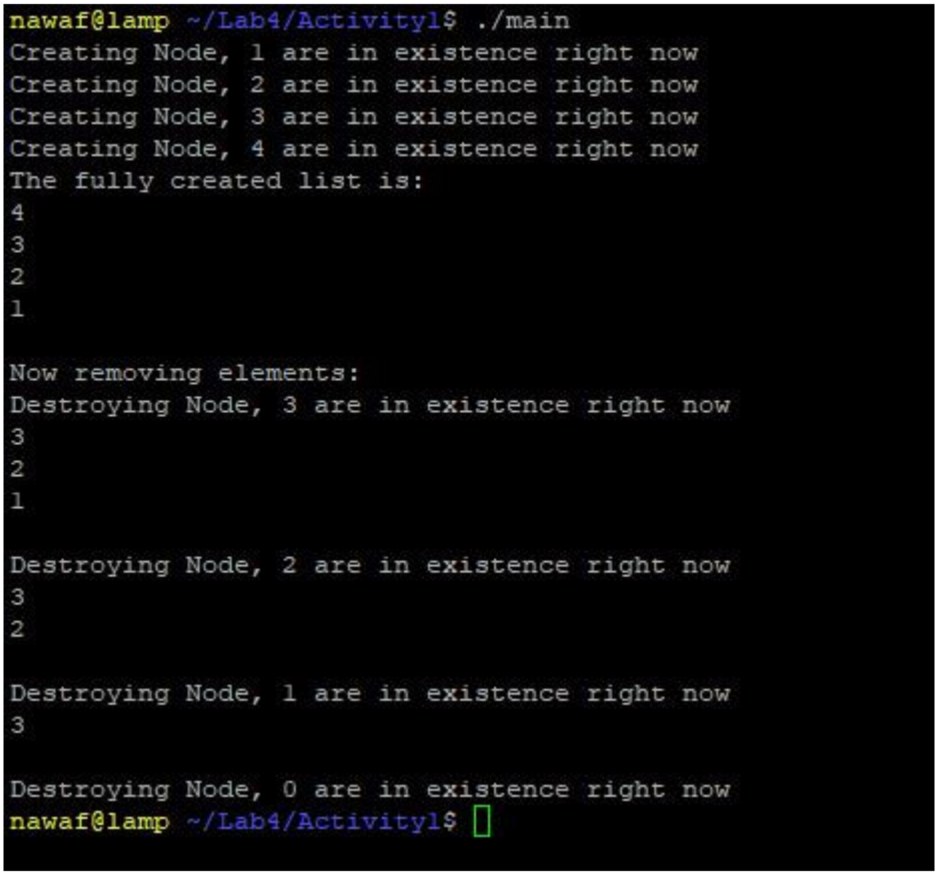
delete marker;

marker = 0;

Additionally, it is mentioned that the red line has already been removed.

With these corrections applied, the code is now fixed. The corrected version of the code will be submitted in the zip file, along with a screenshot of the output.

Output:

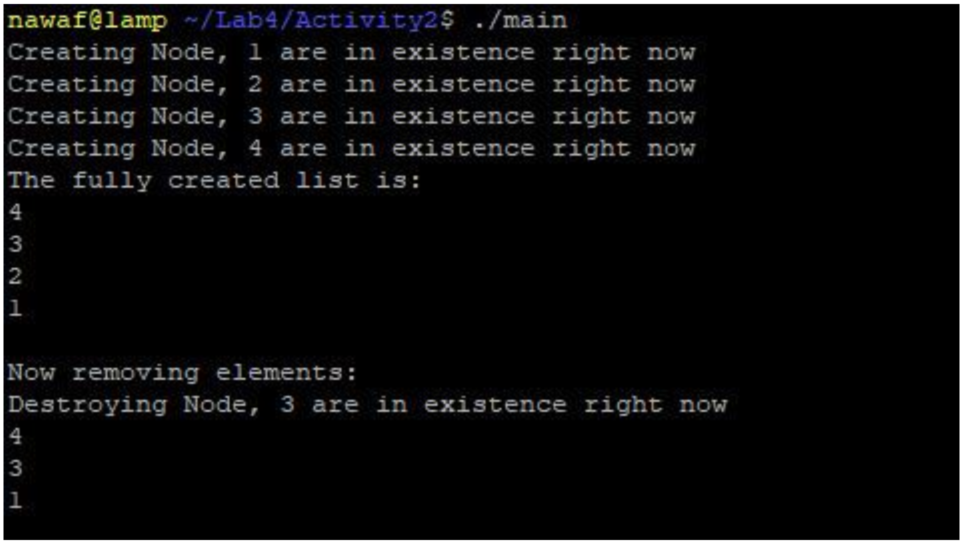


```
nawaf@lamp ~/Lab4/Activity1$ ./main
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

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
nawaf@lamp ~/Lab4/Activity1$ 
```

Activity 2:

After fixing in the first activity, I had no bugs in the second one. The screenshot outputs will be shown below:

1:



```
nawaf@lamp ~/Lab4/Activity2$ ./main
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
```