King Abdulaziz University
Faculty of Engineering
Electrical & Computer Eng
Department

LAB #4

Operating Systems– EE463

Lecturer(s): Dr. Abdulghani M. Al-Qasimi
& Eng. Turki Abdul Hafeez
3rd Semester Spring 2023

#	Name	UID
1	Ahmed Alghamdi	1937937

1- The remove() function seems to have a memory leak issue. When an element is removed from the beginning of the list, the head_ pointer is reassigned to a new node, but the old node is never deleted. To fix this, we can add a delete statement to free the memory of the old node. This is code:

```
if (marker->next() == 0) {
 head_ = 0;
 delete marker; // Free the memory of the old node
 marker = 0;
} else {
 head_ = new Node(marker->value(), marker-
>next());
 delete marker; // Free the memory of the old node
 marker = 0;
}
```

2- Another bug is that when removing an element, the next pointer of the previous node is not updated. To fix this, we need to update the next pointer of the previous node to point to the node after the one being removed. This is the code:

```
if (marker->next() == 0) {
  head_ = 0;
  delete marker;
  marker = 0;
} else {
  head_ = new Node(marker->value(), marker->next());
  delete marker;
  temp->next(marker->next()); // Update the next pointer of the previous
node
  marker = 0;
}
temp->next(marker->next()); // Update the next pointer of the previous
node
delete temp;
temp = 0;
```