# Chaining.h

if (m.empty() == true)  // Create a new entry in the hash table

and exit.

| x.first = index; | neuesElement ->next = x.second;<br>x.second = neuesElement;<br>node *neuesElement = nullptr; |

cerr << "\n\tIndex is already in hash table" << endl;
and exit.

| index == it->first | neuesElement ->next = it->second;<br>it->second = neuesElement; |

cout << "\n\tCreating new entry in hash Table..." << endl;
return 0;

| x.first = index; | neuesElement ->next = x.second;<br>x.second = neuesElement;<br>node *x.second = nullptr; |

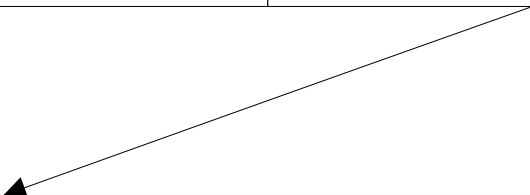it->first = index

it->second = neuesElement

it second = nullptr

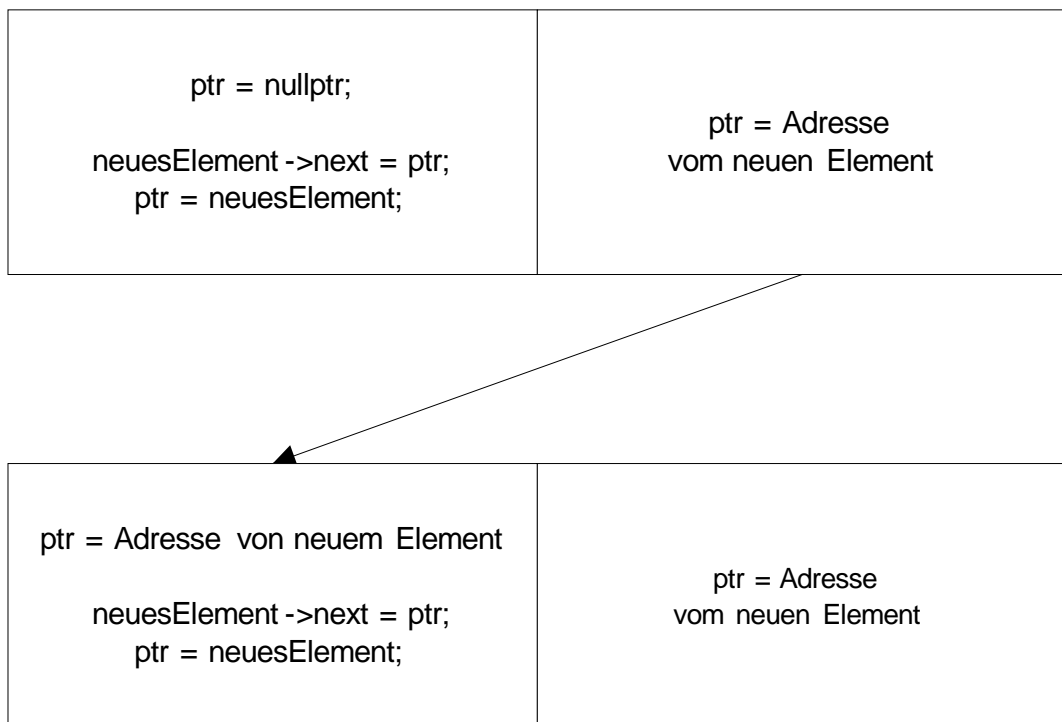| | |
|---|---|
| x.second = nullptr;<br><br>neuesElement ->next = x.second;<br>x.second = neuesElement; | x.second = Adresse<br>vom neuen Element |

| | |
|---|---|
| neuesElement ->next = x.second;<br>x.second = neuesElement; | x.second = Adresse<br>vom neuen Element |

## Linked List

| | |
|---|---|
| ptr = nullptr;<br><br>neuesElement ->next = ptr;<br>ptr = neuesElement; | ptr = Adresse<br>vom neuen Element |

| | |
|---|---|
| ptr = Adresse von neuem Element<br><br>neuesElement ->next = ptr;<br>ptr = neuesElement; | ptr = Adresse<br>vom neuen Element |

Little Helper :)

root ● → 1 ● → 2 ● → tail ● → NULL