Abhishikat

```
class Dictionary
  {. . . }

//Algo insert

void Dictionary :: insert (int key)
 {   index = int (key % max);
     ptr[index] = (node_type)* malloc (sizeof (node_type));
     ptr [index] -> data = key;
     if (root [index] == NULL)
        { root [index] = ptr [index];
          root [index] -> next = NULL;
             temp [index] = ptr [index];
        }
     else { temp [index] = root [index];
            while (temp [index] -> next != NULL)
                     temp [index] = tem [index] -> next;
            temp [index] -> next = ptr [index];
          }
 }
// Search
 void Dictionary :: search (int key)
  { int flag = 0;
     idx = int (key % max);
     ~~temp~~ (idx) = root (index)
while (temp[index] != NULL)
  { if (temp[index] ->data == key)
      { cout << "Found";
        flag = 1;
           break;
      }
     else { temp[index] = temp[index] -> next;
          }
  }
if (flag == 0)
    cout << "Not found";

// Algo delete

   index = int(key%max);
  temp[index] = root[index];
  while (temp[index]->data != key && temp[index] != NULL)
  {   ptr[index] = temp[index];
      temp[index] = temp[index] -> key;
  }
  ptr[index]->next = temp[index]->next;
  cout << "\n" << temp[index]->data << " Deleted";
   temp[index]->data = -1;
    temp[index] = NULL;
   free(temp[index]);
}
```