Data Structures
BITL3

Worksheet 7

Fall 2022

27 October 2022

1. **Warm up:** Answer the following True / False questions.

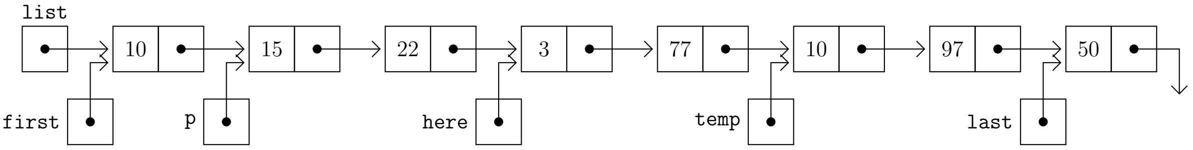
   (a) A doubly linked list of 10 integers take up twice the space of a singly linked list of 10 integers.

   (b) The only difference between a circular linked list and a doubly linked list is the number of pointers.

   (c) In a doubly linked list, whenever a node is deleted, the pointers of all the nodes need to be changed.

2. *(Adapted from Malik, Ex 16.6)* This question is about *pointers* in *linked lists.* Consider the linked list below and the declared variables. The structure `nodeType` has two members, `int info;` for holding an integer, and `nodeType* link;` for holding a pointer to another `nodeType`.

```
nodeType *list, *first, *last, *here, *temp, *follow, *p, *q;
```



   (a) What is the output, if any, of each of the following statements?

   i. `cout << p->info;`

   ii.
```
q = p->link;
cout << q->info << " " << here->info;
```

   iii. `cout << here->link->info;`

   iv.
```
follow = here->link->link;
follow->link = nullptr;
cout << follow->info;
```

   v. `cout << last->link->info;`

   vi.
```
q = here->link;
cout << q->link->link->info;
```

   (b) What is the value of each of the following expressions?

   i. `p->link->link == here`

   ii. `first->link->link->info == 22`

   iii. `temp->link == 0`

   iv. `last->link == nullptr`

   v. `list->link == p`

   vi. `p->link->link->link->info == temp->info`

   (c) What are the effects, if any, of each of the following statements?

   i.
```
follow = temp->link;
follow->link = nullptr;
delete last;
last = follow;
```

   ii. `temp->link = last;`

   iii. `first->info = 58;`

   iv.
```
q = here->link;
here->link = temp;
delete q;
```

   v.
```
q = p->link->link->link;
q->info = 60;
```

   vi. `p->link = temp;`

(d) Write statements that do the following.

i. Set the `info` of the third node to 43.
ii. Make `q` point to the node with `info 77`.
iii. Advance `first` to point to the next node.
iv. Make `follow` point to the node before `here`.
v. Make `p` point to an empty list.
vi. Set the value of the node before `last` to `45`.
vii. Write a `while` loop to make `first` point to the node with `info 97`.

(e) Mark each of the following statements as valid or invalid. If a stetement is invalid, explain why.

i. `p = list->link;`

ii. `first = list;`

iii. `temp->link = nullptr;`

iv. `here->link = temp->info;`

v. `p = *last;`

vi. `first = 90;`

vii. `p->link->info = here->info;`

viii. `here->info = temp->link;`

ix. `*list = *temp;`

x. `temp->link = last->link->link;`

xi. `cout << follow->link->link->info;`