Practice #6

Lists

Yunmin Go

School of CSEE

HANDONG GLOBAL UNIVERSITY

Practice #6 TO-DO List

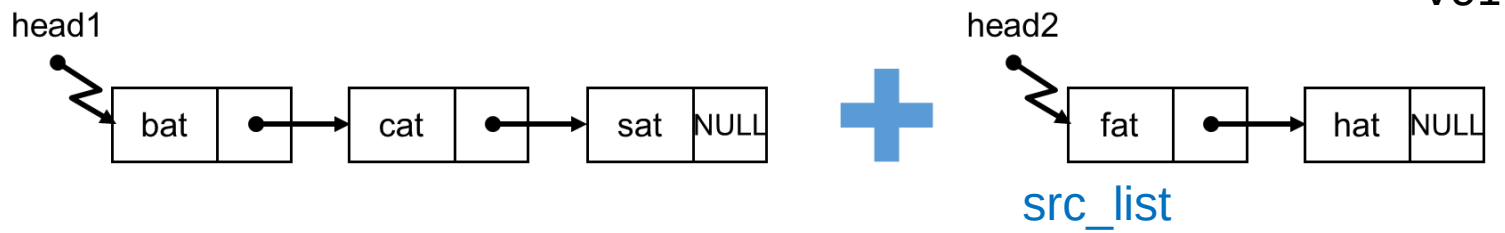
To-Do	Submission	Notes
Singly Linked List	Screenshot and source code (LList.cpp)	p.6-20, Chapter 4
Linked Stack	Screenshot and source code (LStack.cpp)	p.23-24, Chapter 4
Linked Queue	Screenshot and source code (LQueue.cpp)	p.25-26, Chapter 4

- Upload your screenshot and source codes on LMS by 11pm on 4/7 (Wed).
  - All your screenshots should be merged in one pdf file, screenshot.pdf.
  - Your pdf and all source codes should be compressed into zip file.
- File name: practice06_Your Student ID_Name.zip (only zip, not pdf, docx, c, etc)
  - ex) practice06_20400022_고윤민.zip

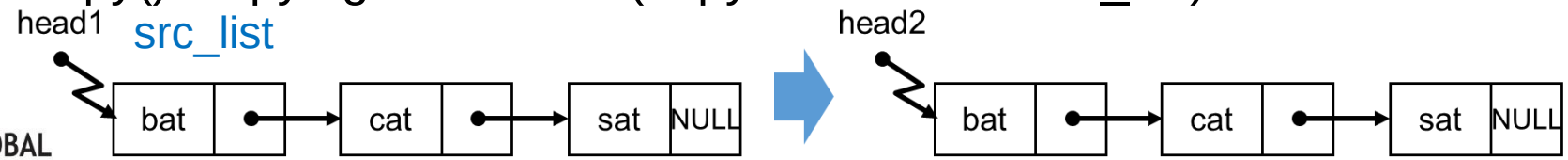
Singly Linked List

- Implement a singly linked list class
  - Skeleton code: LList.cpp (LListclient.cpp: no need to change)
  - Refer to p.6-20, Chapter 4
  - Define LList class and implement following member functions
    - GetBytData(), GetByIndex(), Insert(), InsertAfterNode(), Delete(), Print(), IsEmpty()
      - Above functions have been already introduced in Chapter 4.
    - GetLength(): get the number of elements in list
    - Concat(): concatenation of linked lists

```
int  GetLength();
void Concat(LList* src_list);
void Copy(LList* src_list);
```



  - Copy(): copying linked lists (copy all nodes of src_list)



Singly Linked List

- Expected results

```
PS C:\ds\practice06> .\LListclient.exe
List:
List: bat  cat  sat  vat
GetByData: cat

List: bat  cat  mat  sat  vat
List: vat  sat  mat  cat  bat
GetByIndex: mat

List: vat  sat  mat  bat
List: 333  222  111
List's length: 7
List2's length: 3
List: vat  sat  mat  bat  333  222  111
List: vat  sat  mat  bat  333  222  111
```

Linked Stack

- Implement a linked stack class
  - Skeleton code: LStack.cpp (LStackclient.cpp: no need to change)
  - Refer to p.23-24, Chapter 4
  - Define LStack class and implement following member functions
    - LStack(), ~LStack(), Push(), Pop(), Print(), IsEmpty()
      - Push() and Pop() have been already introduced in Chapter 4.
    - You can also implement other functions if you need

Linked Stack

- Expected results

```
PS C:\ds\practice06> .\LStackclient.exe
Stack: Data_Structures  University  Global  Handong  World  Hello
Pop: Data_Structures
Pop: University
Stack: Global  Handong  World  Hello
```

Linked Queue

- Implement a linked queue class
  - Skeleton code: LQueue.cpp (LQueueclient.cpp: no need to change)
  - Refer to p.25-26, Chapter 4
  - Define LQueue class and implement following member functions
    - LQueue(), ~LQueue(), AddQ(), DeleteQ (), Print(), IsEmpty()
      - Push() and Pop() have been already introduced in Chapter 4.
    - You can also implement other functions if you need

Linked Queue

- Expected results

```
PS C:\ds\practice06> .\LQueueclient.exe
Queue: Hello  World  Handong  Global  University
Delete: 1, Hello
Delete: 2, World
Queue: Handong  Global  University  Data  Structure  Fun!!!!!
```