Object Oriented Programming (Week 9)

2023

KWANGWOON UNIVERSITY
DEPT. OF COMPUTER ENGINEERING

Contents

ASSIGNMENT 3-1. 1

Assignment 3-1. 1

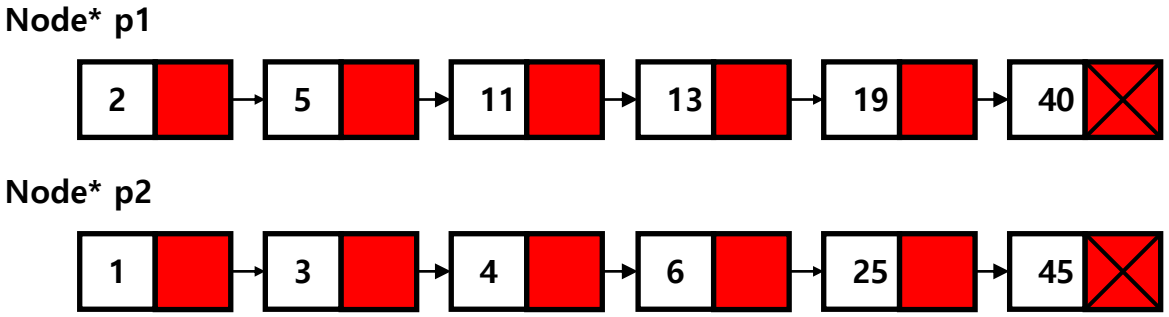
- Write Merge_List(Node *p1, Node *p2, Node* p3) function that has three arguments: pointers to the **first nodes of two sorted lists (p1 and p2)**. Then, it **merges the two lists into the third list (p3), which also is sorted**. Note that each node of the linked lists is for a string and the nodes in a list are sorted in an **ascending order** of alphabet. You should show the result on a screen as shown in the example below: (Not case-sensitive)

```
Ex)
Input>>
Input list 1: Well done is better than well said
Input list 2: Blaze with the fire that is never extinguished

Output>>
Result: better Blaze done extinguished fire is is never said than that the Well well with
```

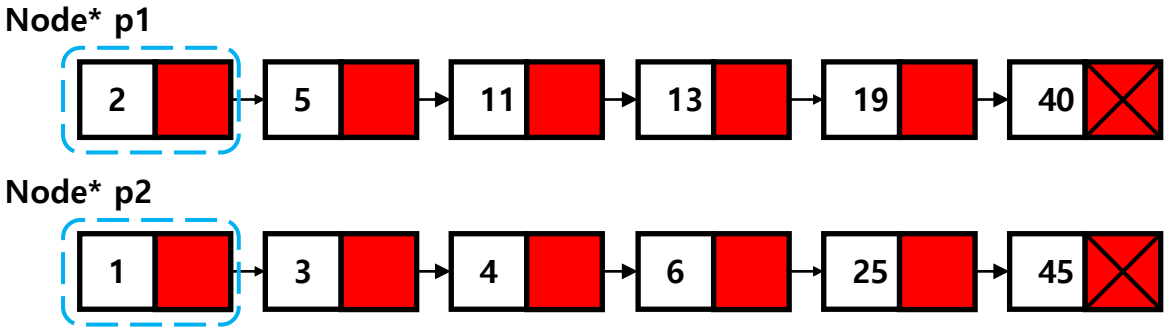
Assignment 3-1. 1

- Merging two sorted list
  - Initial State



Assignment 3-1. 1

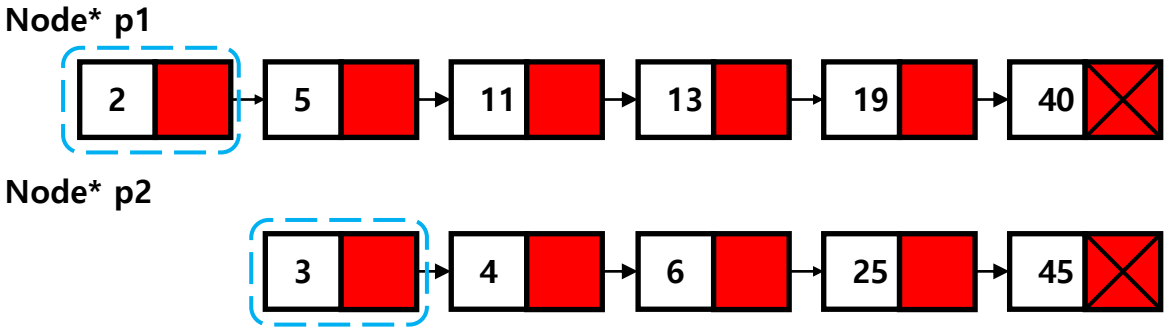
- Merging two sorted list
  - Initial State



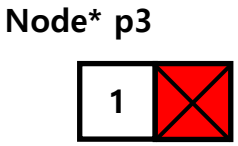
  - Compare the value of the node: **2 > 1**

Assignment 3-1. 1

- Merging two sorted list
  - Initial State



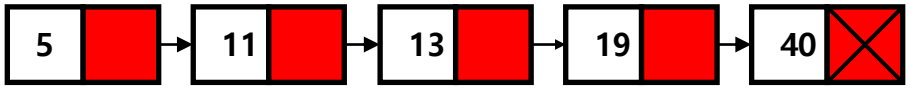
  - Compare the value of the node: **2 < 3**



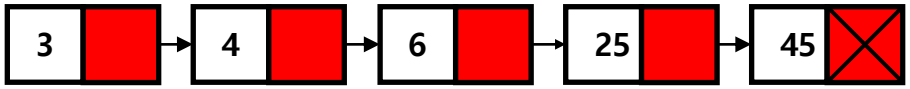
Assignment 3-1. 1

- Merging two sorted list
  - Initial State

Node* p1

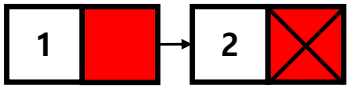


Node* p2



  - Compare the value of the node: **2 < 3**

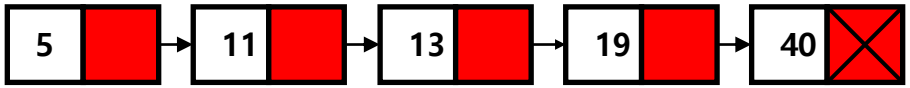
Node* p3



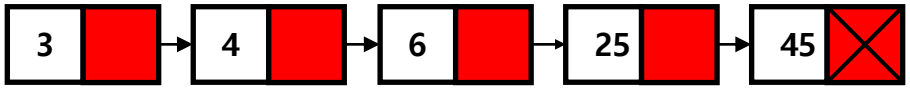
Assignment 3-1. 1

- Merging two sorted list
  - Initial State

Node* p1

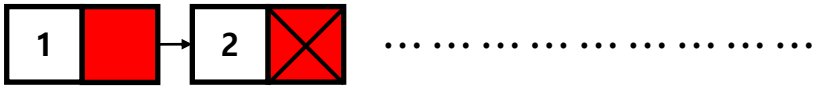


Node* p2



  - Compare the value of the node: ……

Node* p3



……

Assignment 3-1. 1

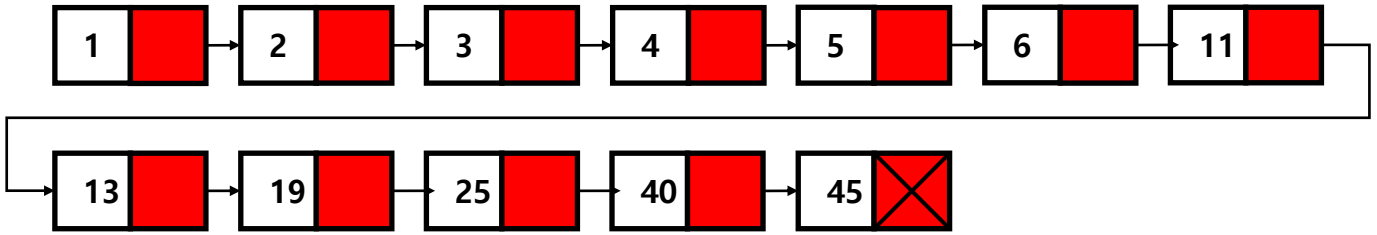
- Merging two sorted list
  - Initial State

Node* p1

Node* p2

  - Completed merged list

Node* p3



ASSIGNMENT 3-1. 2

Assignment 3-1. 2

- As you know, an e-mail address consists of an identification (ID) and a host name separated by at-sign (e.g. name@gmail.com or xyz@kw.ac.kr). **Write a program that reads an e-mail address and break it into multiple parts and then displays them separately.** You must write your own function (my_strtok) to tokenize the e-mail address. The prototype of 'my_strtok' function is as follows:

char* **my_strtok**(char* str)

This function must operate similar to the built-in 'strtok' function which separates a string by several delimiters. **Unlike the 'strtok' function, 'my_strtok' function has two default delimiters ('@', and '.').** The parameter, str, is the pointer indicating the string that will be separated by 'my_strtok' function. This function returns the address of the first token of 'str' in each function call. Handle all possible exceptions of input in your program. You are not allowed to use any string functions

Ex1) name@gmail.com name gmail com	Ex2) xyz@kw.ac.kr xyz kw ac kr

Assignment 3-1. 2

- Static variable
  - Have a property of preserving their value event after they are out of scope
  - Preserve their previous value in their previous scope
  - Not initialized again in the new scope

```
1  #include <iostream>
2
3  int f()
4  {
5      static int n = 0;
6      n++;
7      return n;
8  }
9
10 int main()
11 {
12     for (int i = 0; i < 10; i++)
13         std::cout << f() << std::endl;
14
15     return 0;
16 }
```

```
$ ./a.out
The vaule of n in f(): 1
The vaule of n in f(): 2
The vaule of n in f(): 3
The vaule of n in f(): 4
The vaule of n in f(): 5
The vaule of n in f(): 6
The vaule of n in f(): 7
The vaule of n in f(): 8
The vaule of n in f(): 9
The vaule of n in f(): 10
```

ASSIGNMENT 3-1. 3

Assignment 3-1. 3

- Define the queue class. The size of queue should be obtained by user. Queue class must have IsEmpty(), IsFull(), Pop() and Push() functions. And Write a main function as a demonstration program to test your class and functions.

```
class Queue
{
  private:
    Node* m_pHead;
    Node* m_pTail;
    int m_Size;
    int m_NumElement;

  public:
    Queue();
    ~ Queue();

    void SetSize(int n);
    bool IsEmpty();
    bool IsFull();
    bool Push(Node* pNode);
    Node* Pop();
    void PrintQueue();
};
```

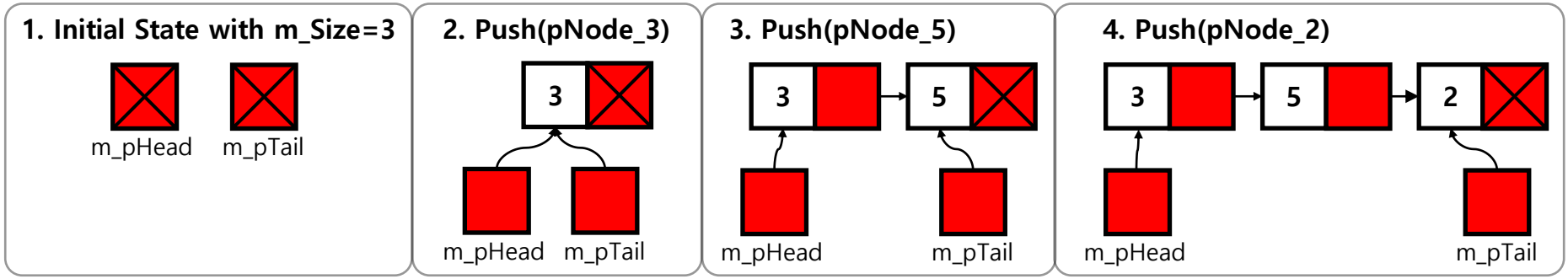
```
class Node
{
  private:
    Node* m_pNext;
    int m_Data;

  public:
    Node();
    ~ Node();

    void SetData(int n);
    void SetNext(Node* pNext);
    int GetData();
    Node* GetNext();
}
```

Assignment 3-1. 3

- Queue Data Structure using Linked List

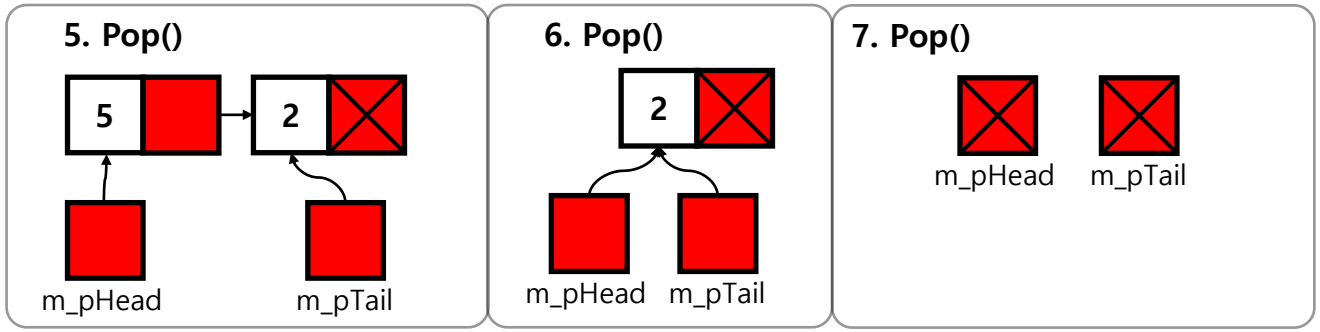


IsEmpty() == true
IsFull() == false

IsEmpty() == false
IsFull() == false

IsEmpty() == false
IsFull() == false

IsEmpty() == false
IsFull() == true



IsEmpty() == false
IsFull() == false
Return pNode_3

IsEmpty() == false
IsFull() == false
Return pNode_5

IsEmpty() == true
IsFull() == false
Return pNode_2

ASSIGNMENT 3-1. 4

Assignment 3-1. 4

- Define the stack class. The size of stack should be obtained by user. Stack class must have IsEmpty(), IsFull(), Pop() and Push() functions. And write a main function as a demonstration program to test your class and functions.

```
class Stack
{
  private:
    Node* m_pHead;
    int m_Size;
    int m_NumElement;

  public:
    Stack();
    ~ Stack();

    void SetSize(int n);
    bool IsEmpty();
    bool IsFull();
    bool Push(Node* pNode);
    Node* Pop();
    void PrintStack();
};
```

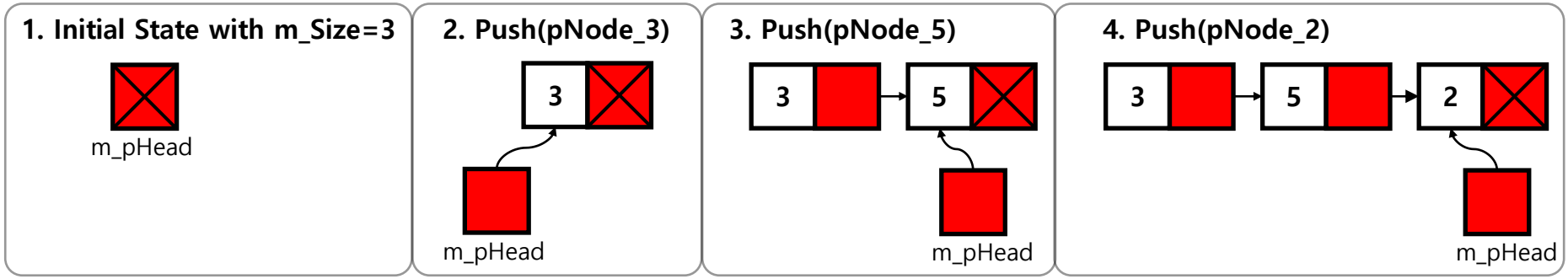
```
class Node
{
  private:
    Node* m_pNext;
    int m_Data;

  public:
    Node();
    ~ Node();

    void SetData(int n);
    void SetNext(Node* pNext);
    int GetData();
    Node* GetNext();
}
```

Assignment 3-1. 4

- Stack Data Structure using Linked List

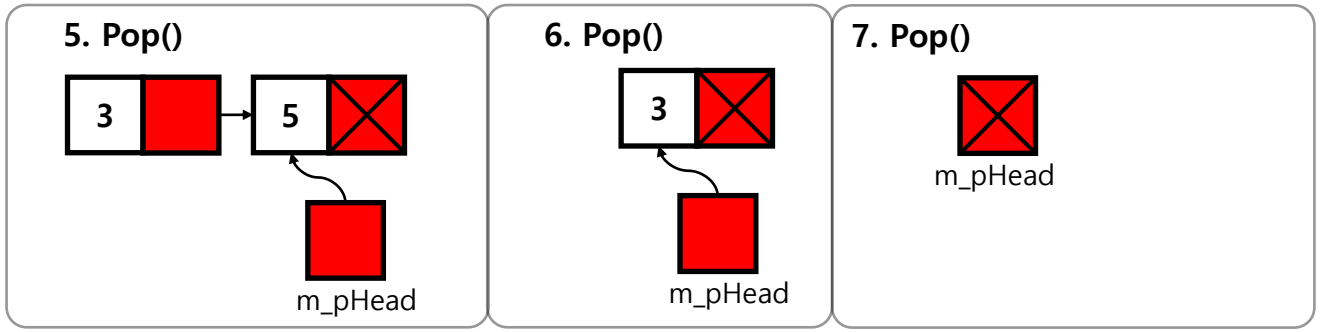


IsEmpty() == true
IsFull() == false

IsEmpty() == false
IsFull() == false

IsEmpty() == false
IsFull() == false

IsEmpty() == false
IsFull() == true



IsEmpty() == false
IsFull() == false
Return pNode_2

IsEmpty() == false
IsFull() == false
Return pNode_5

IsEmpty() == true
IsFull() == false
Return pNode_3

과제 제출 방법

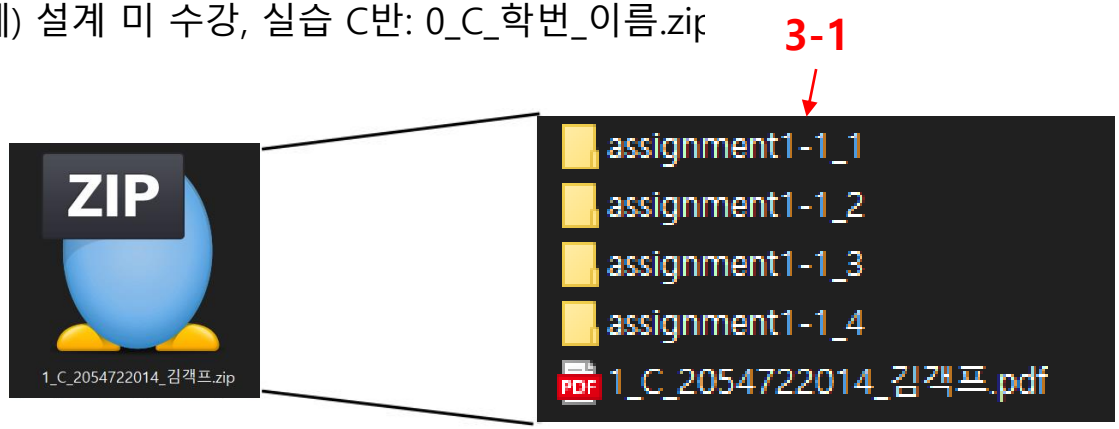
과제 제출 방법

- **FTP Upload (Klas 과제 제출 X)**
  - Address : ftp://223.194.8.1:1321
  - username : IPSL_OBJ
  - password : ipslobj_2023

- **Due date**
  - Soft copy: 마감일 5/5(금) 23:59:59까지 제출 (서버시간 기준)
  - Delay
    - 마감일 이후 +7일까지 제출 가능
    - 단, 1일 초과마다 과제 총점의 10%씩 감점

과제 제출 방법

- Soft copy
  - 과제(보고서, 소스 코드)를 압축한 파일 제출
    - 설계반_실습반_학번_이름.zip
      - 예) 설계1반 수강, 실습 A반: 1_A_학번_이름.zip
      - 예) 설계 수강, 실습 미수강: 2_0_학번_이름.zip
      - 예) 설계 미 수강, 실습 C반: 0_C_학번_이름.zip



  - 과제 수정하여 업로드 시 버전 명시
    - 설계반_실습반_학번_이름_verX.zip

과제 제출 방법

- Soft copy
  - 과제 보고서
    - 영문 또는 한글로 작성
    - **반드시 PDF**로 제출 (PDF 외 파일 형식으로 제출시 0점 처리)
    - 보고서 양식
      - 문제 및 설명(문제 capture 금지) / 결과 화면 / 고찰
      - 보고서 양식은 아래 경로에서 참고
        - https://www.ipsl.kw.ac.kr/post/1%EC%B0%A8-%EA%B3%BC%EC%A0%9C
    - **소스코드 제외**
    - 분량 제한 없음
    - **표절 적발 시 0점 처리**
  - 소스 코드
    - Visual Studio 2022 community 사용 필수
      - https://docs.microsoft.com/ko-kr/visualstudio/install/install-visual-studio?view=vs-2022
    - STL (Standard Template Library) 사용 금지 (vector, map, algorithm 등)
    - Debug 폴더를 제외한 모든 파일 제출
      - .sln 파일 포함(.cpp 만 제출하지 말것)
    - **각 문제마다 프로젝트 파일 생성 필수**
    - **주석 반드시 달기**
    - **소스코드 표절 적발 시 0점 처리**

END OF PRESENTATION

Q&A