과목명	자료구조
담당교수	우진운 교수님
학과	소프트웨어학과
학번	32153180
이름	이상민
제출일자	2018.10.29

소스 코드

```
#include <iostream>
using namespace std;

class LinkedStack;              // 연결스택 전방선언
class LinkedQueue;              // 연결큐 전방선언
class ChainNode
{
    friend LinkedStack;         // LinkedStack friend 지정
    friend LinkedQueue;         // LinkedQueue freind 지정
private:
    int data;                   // data field
    ChainNode *link;            // link field
public:
    ChainNode(int element = 0, ChainNode *next = 0)     // ChainNode 생성자
    {
        data = element;
        link = next;
    }
};


class LinkedStack
{
private:
    ChainNode *top;
public:
    LinkedStack()               // LinkedStack 생성자
    {
        top = 0;
    }
    void Push(const int &x);    // LinkedStack 삽입함수
    int* Pop(int &x);           // LinkedStack 삭제함수
    void Print();               // LinkedStack 출력함수
};
void LinkedStack::Push(const int &x)
{
    top = new ChainNode(x, top);        // 새 노드를 top에 저장
}
int* LinkedStack::Pop(int &x)
{
    if (top == 0)                       // 스택이 비어있을 경우
        return 0;
    ChainNode *temp = top;
    x = top->data;                      // top 노드의 data field를 x에 저장
    top = top->link;                    // top을 다음 노드로 이동
    delete temp;                        // 노드 삭제
    return &x;                          // x의 주소 반환
}

void LinkedStack::Print()
{
    ChainNode *p = top;

    if (top == 0)                       // 스택이 비어있을 경우
    {
        cout << "비어있는 스택" << endl;
        return;
    }

    cout << "스택(LIFO순) : ";
    for (p; p->link; p = p->link)       // p가 top부터 다음 노드로 이동
    {
        cout << p->data << ' ';         // p의 data field 출력
    }
    cout << p->data << endl;            // 마지막 data field
}
```

```
class LinkedQueue
{
private:
    int data;
    ChainNode *front;
    ChainNode *rear;
public:
    LinkedQueue()                       // LinkedQueue 생성자
    {
        front = rear = 0;
    }
    void Push(const int &x);            // LinkedQueue 삽입함수
    int* Pop(int &x);                   // LinkedQueue 삭제함수
    void Print();                       // LinkedQueue 출력함수
};
void LinkedQueue::Push(const int &x)
{
    if (front == 0)                     // 큐가 비어있을 경우
        front = rear = new ChainNode(x, 0);
    else
        rear = rear->link = new ChainNode(x, 0);        // 노드 삽입하고 rear 수정
}
int* LinkedQueue::Pop(int &x)
{
    if (front == 0)                     // 큐가 비어있을 경우
        return 0;
    ChainNode *temp = front;            // 새로운 노드가 맨 앞 노드를 가리키도록 설정
    x = front->data;                    // 삭제할 값을 x에 저장
    front = front->link;                // front를 다음
    delete temp;                        // 맨 앞 노드 삭제
    return &x;
}
void LinkedQueue::Print()
{
    ChainNode *p = front;

    if (rear == 0)                      // 큐가 비어있을 경우
    {
        cout << "비어있는 큐" << endl;
        return;
    }

    cout << "큐(FIFO순) : ";
    for (p; p->link; p = p->link)       // p가 front부터 다음 노드로 이동
    {
        cout << p->data << ' ';         // p의 data field 출력
    }
    cout << p->data << endl;            // 마지막 data field 출력
}

int main()
{
    LinkedStack stack;                  // LinkedStack 객체 생성
    LinkedQueue queue;                  // LinkedQueue 객체 생성
    int menu, num;

    cout << "---------------메뉴---------------" << endl;
    cout << "1. 스택에 삽입     2. 큐 삽입" << endl;
    cout << "3. 스택에서 삭제   4. 큐에서 삭제" << endl;
    cout << "5. 스택 보기       6. 큐 보기" << endl;
    cout << endl;
```

```
    while (1)
    {
        cout << "메뉴와 숫자 입력 : ";
        cin >> menu;
        switch (menu)
        {
        case 1:
            cin >> num;
            stack.Push(num);          // LinkedStack 삽입함수 호출
            break;
        case 2:
            cin >> num;
            queue.Push(num);          // LinkedQueue 삽입함수 호출
            break;
        case 3:
            cin >> num;
            stack.Pop(num);           // LinkedStack 삭제함수 호출
            break;
        case 4:
            cin >> num;
            queue.Pop(num);           // LinkedQueue 삭제함수 호출
            break;
        case 5:
            stack.Print();            // LinkedStack 출력함수 호출
            break;
        case 6:
            queue.Print();            // LinkedQueue 출력함수 호출
            break;

        }
    }
}
```

실행 파일

```
C:₩Windows₩system32₩cmd.exe
----------메뉴----------
1. 스택에 삽입          2. 큐 삽입
3. 스택에서 삭제        4. 큐에서 삭제
5. 스택 보기            6. 큐 보기

메뉴와 숫자 입력 : 5
비어있는 스택
메뉴와 숫자 입력 : 6
비어있는 큐
메뉴와 숫자 입력 : 1 1
메뉴와 숫자 입력 : 1 2
메뉴와 숫자 입력 : 1 3
메뉴와 숫자 입력 : 5
스택(LIFO순) : 3 2 1
메뉴와 숫자 입력 : 2 1
메뉴와 숫자 입력 : 2 2
메뉴와 숫자 입력 : 2 3
메뉴와 숫자 입력 : 6
큐(FIFO순) : 1 2 3
메뉴와 숫자 입력 : 3 3
메뉴와 숫자 입력 : 5
스택(LIFO순) : 2 1
메뉴와 숫자 입력 : 4 1
메뉴와 숫자 입력 : 6
큐(FIFO순) : 2 3
```