

## < 소스 코드 >

```
1  #include <iostream>
2  using namespace std;
3
4  class LinkedStack; // 연결스택 전방선언
5  class LinkedQueue; // 연결큐 전방선언
6
7  class ChainNode {
8      friend class LinkedStack; // 연결스택 클래스 프렌드 지정
9      friend class LinkedQueue; // 연결큐 클래스 프렌드 지정
10 private:
11     int data; // 노드의 데이터 필드
12     ChainNode *link; // 노드의 링크 필드
13 public:
14     ChainNode(int element = 0, ChainNode *next = 0) { // 노드 생성자 함수
15         data = element; link = next;
16     }
17 };
18 class LinkedStack {
19 private:
20     ChainNode *top;
21 public:
22     LinkedStack() { top = 0; }; // 연결스택 생성자 함수
23     void Push(const int&); // 연결스택 삽입 함수
24     int *Pop(int&); // 연결스택 삭제 함수
25     void Print(); // 연결스택 출력 함수
26 };
27 // 스택 삽입
28 void LinkedStack::Push(const int& e) {
29     top = new ChainNode(e, top); // 새 노드를 스택의 맨 위에 저장
30 }
31 // 스택 삭제
32 int* LinkedStack::Pop(int &x)
33 //스택에서 톱 노드를 삭제하고 값을 x에 저장한 후에 x의 주소를 반환
34 {
35     if (top == 0) return 0; //빈 스택이면 null을 반환
36     ChainNode* delNode = top;
37     x = top->data; //톱 노드의 data 필드를 x에 저장
38     top = top->link; //톱 위치를 다음 노드로 이동
39     delete delNode; //노드 삭제
40     return &x;
41 }
42 // 스택 출력
43 void LinkedStack::Print() {
44     ChainNode *p = top;
45     if (top == 0) { // 빈 스택일 경우
46         cout << "Stack is empty!!" << endl;
47         return; // 함수 종료
48     }
49     cout << "Stack : ";
50     for (p; p->link; p = p->link) { // p가 톱에서 밑으로 이동하면서
51         cout << p->data << " "; // p의 데이터값 출력
52     }
53     cout << p->data << endl; // 마지막 수를 가르킬 때 출력
54 }
```

```

55
56 class LinkedQueue {
57     private:
58         ChainNode *front, *rear;
59     public:
60         LinkedQueue() { front = 0; }; // 연결큐 생성자 함수
61         void Push(const int&); // 연결큐 삽입 함수
62         int* Pop(int&); // 연결큐 삭제 함수
63         void Print(); // 연결큐 출력 함수
64     };
65     // 큐 삽입
66     void LinkedQueue::Push(const int& e)
67     {
68         if (front == 0) front = rear = new ChainNode(e, 0); //공백 큐
69         else rear = rear->link = new ChainNode(e, 0);
70         //노드를 삽입하고 rear를 수정함
71     }
72     // 큐 삭제
73     int* LinkedQueue::Pop(int& retvalue)
74     //큐에 있는 첫번째 노드를 제거하고 그 데이터의 포인터를 반환
75     {
76         if (front == 0) return 0; // 공백 큐이므로 null을 반환
77         ChainNode* delNode = front; // 새로운 노드를 생성해서 맨 앞의 노드를 가르키게 함
78         retvalue = front->data; // 삭제되는 값을 retvalue에 저장
79         front = front->link; // front는 다음 큐로 이동
80         delete delNode; // 첫 번째 노드 삭제
81         return &retvalue;
82     }
83     // 큐 출력
84     void LinkedQueue::Print() {
85         ChainNode *p;
86         if (rear == 0) { // 빈 큐일 경우
87             cout << "Queue is empty!!" << endl;
88             return; // 함수 종료
89         }
90         cout << "Queue : ";
91         for (p = front; p->link; p = p->link) { // p가 맨 앞부터 맨 뒤로 이동
92             cout << p->data << " "; // 마지막 전 data까지 출력
93         }
94         cout << p->data << endl; // 마지막 data를 가리킬 때 출력
95     }
96     int main(void) {
97         LinkedStack myStack; // 연결스택 객체 생성
98         LinkedQueue myQueue; // 연결큐 객체 생성
99         int num, value;
100
101         cout << "-----메뉴-----" << endl;
102         cout << "1. 스택에 삽입          2. 큐에 삽입" << endl;
103         cout << "3. 스택에서 삭제          4. 큐에서 삭제" << endl;
104         cout << "5. 스택 내용 보기          6. 큐 내용 보기" << endl;
105

```

```

106 while (1) {           //반복하여 실행
107     cin >> num;        // 메뉴 선택 입력 받기
108     switch (num) {     // 조건문
109     case 1:
110         cin >> value;
111         myStack.Push(value);    // 연결스택 삽입함수 호출
112         break;
113     case 2:
114         cin >> value;
115         myQueue.Push(value);    // 연결큐 삽입함수 호출
116         break;
117     case 3:
118         myStack.Pop(value);     // 연결스택 삭제함수 호출
119         break;
120     case 4:
121         myQueue.Pop(value);     // 연결큐 삭제함수 호출
122         break;
123     case 5:
124         cout << "-> ";
125         myStack.Print();        // 연결스택 출력함수 호출
126         break;
127     case 6:
128         cout << "-> ";
129         myQueue.Print();        // 연결큐 출력함수 호출
130         break;
131     default:
132         return 0;
133     }
134 }
135 return 0;
136 }

```

## < 실행 화면 >

```
C:\Users\LEE KUNUK\Desktop\2-2의 건축이\고급프로그래밍\2-2 C\Debug\2-2 C.exe
-----메뉴-----
1. 스택에 삽입                2. 큐에 삽입
3. 스택에서 삭제            4. 큐에서 삭제
5. 스택 내용 보기            6. 큐 내용 보기

1 11
1 22
1 33
1 44
1 55
5
-> Stack : 55 44 33 22 11
3
5
-> Stack : 44 33 22 11
3
5
-> Stack : 33 22 11
2 10
2 20
2 30
2 40
2 50
6
-> Queue : 10 20 30 40 50
4
6
-> Queue : 20 30 40 50
4
6
-> Queue : 30 40 50
-1
계속하려면 아무 키나 누르십시오 . . .
```