

실습문제

1

- 교재 272페이지, 실습문제 5-8
 - ▣ 문제 7번의 MyIntStack를 수정하여 다음과 같이 선언하였다.스택에 저장할 수 있는 정수의 최대 개수는 생성자에서 주어지고 size 멤버에 유지한다. MyIntStack 클래스를 작성하라.
 - ▣ 프로젝트명 : 학번_Chap_5_8
 - ▣ main 파일명: 학번_Chap_5_8.cpp

프로그램 전체 구성 및 class MyIntStack

2

헤드 파일 include

```
class MyIntStack {  
    int *p;           // 스택 메모리로 사용할 포인터  
    int size;         // 스택의 최대 크기  
    int tos;          // 스택의 탑을 가리키는 인덱스  
public:  
    MyIntStack();  
    MyIntStack(int size);  
    MyIntStack(const MyIntStack& s); // 복사 생성자  
    ~MyIntStack();  
  
    bool push(int n); // 정수 n을 스택에 푸시한다.  
                      // 스택이 꽉 차 있으면 false를, 아니면 true 리턴  
    bool pop(int &n); // 스택의 탑에 있는 값을 n에 팝한다.  
                      // 만일 스택이 비어 있으면 false를, 아니면 true 리턴  
};
```

MyIntStack의 각각의 멤버 함수 구현

main() 함수

main() 함수 및 실행결과

3

- MyIntStack 클래스를 활용하는 코드와 실행 결과는 다음과 같다.

```
int main() {
    MyIntStack a(10);
    a.push(10);
    a.push(20);

    MyIntStack b = a;    // 복사 생성: 복사 생성자 호출
    b.push(30);

    int n;
    a.pop(n);            // 스택 a 팝
    cout << "스택 a에서 팝한 값 " << n << endl;
    b.pop(n);            // 스택 b 팝
    cout << "스택 b에서 팝한 값 " << n << endl;
}
```

- 실행결과

```
스택 a에서 팝한 값 20
스택 b에서 팝한 값 30
```

MyIntStack의 생성자 및 소멸자

4

```
MyIntStack::MyIntStack() {  
    size = 10; // 할당할 배열의 크기  
    tos 변수 0으로 초기화           // tos가 0이면 빈 스택을 의미함  
    스택용 int배열[size] 할당하여 p에 저장 // int[] 할당은 예제 4-6 참조  
}  
  
MyIntStack::MyIntStack(int size) {  
    매개변수 size를 멤버변수 size에 저장;  
    tos 변수를 0으로 초기화  
    스택용 int 배열 할당하여 p에 저장  
}  
  
MyIntStack::~~MyIntStack() {  
    if (p) // p가 nullptr이 아니라면, 즉 정상적으로 메모리 할당되었다면  
        스택 메모리 배열 p를 반환 // 예제 4-6 참조  
}
```

MyIntStack의 복사 생성자

5

```
MyIntStack::MyIntStack(const MyIntStack& s) {    // 복사 생성자
    s의 size 멤버를 현재 객체의 size 멤버에 저장
    s의 tos 멤버를 현재 객체의 tos 멤버에 저장
    스택용 int배열[size] 할당하여 p에 저장
    for문을 이용하여    // for문은 예제 4-6 참조
        s의 p[i]를 현재 객체의 p[i]에 저장
}
```

MyIntStack의 push(), pop() 멤버 함수

6

```
bool MyIntStack::push(int n) {  
    스택 가득 찬 상태라면(스택 탑 인덱스가 스택 최대 크기와 동일하면),  
        false를 리턴  
    스택의 탑에(p[?])에 n을 저장  
    스택의 탑을 가리키는 인덱스(?)를 1 증가 시킴  
    true 리턴  
}  
  
bool MyIntStack::pop(int &n) {  
    스택이 비어 있으면(MyIntStack()참조), false를 리턴  
    스택의 탑을 가리키는 인덱스를 1 감소  
    n에 스택의 탑에(p[?])의 값을 저장  
    true 리턴  
}
```