Report #6 (기한: 11/30)

- 다음 BaseArray 클래스를 고려하라.

```
class BaseArray {
private:
 int capacity; // 동적 할당된 메모리 용량
 int* mem;
protected:
 BaseArray(int capacity) {
  this->capacity = capacity; mem = new int[capacity];
 }
 ~BaseArray() { delete[] mem; }
 void put(int index, int val) { mem[index] = val; }
 int get(int index) { return mem[index]; }
 int getCapacity() { return capacity; }
};
```

Report #6 (2)

- BaseArray를 상속받아서 원형 큐로 동작하는 MyQueue 클래스를 다음과 같이 작성하라. (원형 큐에 관해서는 교재 5.3절 참고)
  - MyQueue는 front, rear의 변수를 포함한다.
  - 큐의 항목은 BaseArray에 포함된 mem에 저장된다.
  - 다음의 함수를 포함한다:
    - enqueue(item): 큐가 꽉 차지 않았으면(is_full()로 검사), item을 BaseArray의 mem의 해당 요소에 저장(put() 이용) (큐가 꽉 차있으면 적절한 오류 메시지 출력)
    - dequeue(): 큐가 비어 있지 않으면(is_empty()로 검사), mem의 해당 요소로부터 항목을 삭제하여 반환(get() 이용) (큐가 비어 있으면 적절한 오류 메시지 출력)
    - is_full(): 큐가 꽉 차있으면 true, 그렇지 않으면 false 반환
    - is_empty(): 큐가 비어 있으면 true, 그렇지 않으면 false 반환

테스트

- 테스트 1:

```
int main() {
  MyQueue q(5);

  q.enqueue(1);
  q.enqueue(2);
  q.enqueue(3);
  cout<< q.dequeue() << endl;
  q.enqueue(4);
  q. enqueue(5);
  while (!q.is_empty()) {
            cout << q.dequeue() << ' '; // 큐에서 제거하여 출력
  }
  return 0;
}
```

테스트 (2)

- 테스트 2:

```
int main() {
  MyQueue q(5);

  q.enqueue(1);
  q.enqueue(2);
  q.enqueue(3);
  q.enqueue(4);
  q. enqueue(5);
  while (!q.is_empty()) {
            cout << q.dequeue() << ' '; // 큐에서 제거하여 출력
  }
  return 0;
}
```

보고서 작성 지침

- 다음의 순서로 문제를 해결하라.
  - 분석, 설계하라. (문제에서 객체를 식별하고, 이를 클래스로 설계하라)
  - 코딩: C++ 프로그램 작성(주석 반드시 포함할 것)
  - 테스트: 프로그램의 올바름을 검증할 수 있도록 다양한 데이터에 대해서 테스트하고, 입력 값에 대해서 결과가 올바른지를 검증하라.
  - 의견: 문제 해결과정에서 느낀 점, 의견 등을 기술한다.

- 보고서는 분석/설계, 코딩, 테스트, 의견으로 순서로 작성되어야 한다.

- 유의사항: Don't Copy!!! (Copy하면 10점 감점)