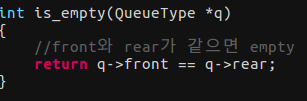
HW3: Queue

2012003567

이상화

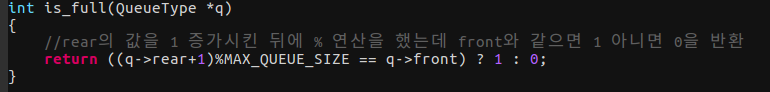
1. 문제 해결의 주요 코드

(1). is\_empty()



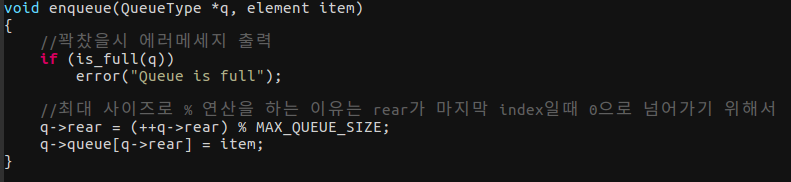
Queue의 초기상태에서 front,rear 둘다 0에서 시작하고 enqueue시 rear는 1증가, dequeue시 front도 1증가 하므로 empty 상태일때는 front와 rear값이 같다.

(2). Is\_full()



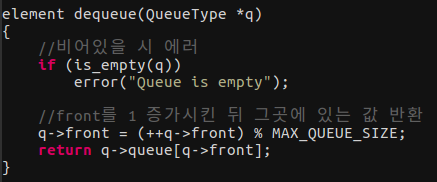
Front는 항상 비어있는 상태로 유지하게 된다. 만약 enqueue를 계속해서 rear값이 front 전까지 왔다면 그 상태는 full인 상태이다. Queue의 size로 %연산을 해주는 이유는 rear가 마지막 index인 상태에서 0으로 넘어가야 하기 때문이다.

(3). Enqueue()



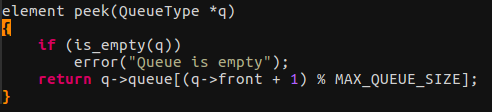
element를 삽입하기 전에 is\_full 함수를 이용해서 꽉찼느지 체크한다. 그 뒤 rear를 1증가 시킨 후 그 위치에 element를 삽입해야한다. 마찬가지로 %연산을 통해 rear가 마지막 index에 있을시에 0으로 옮겨준다.

(4). Dequeue()



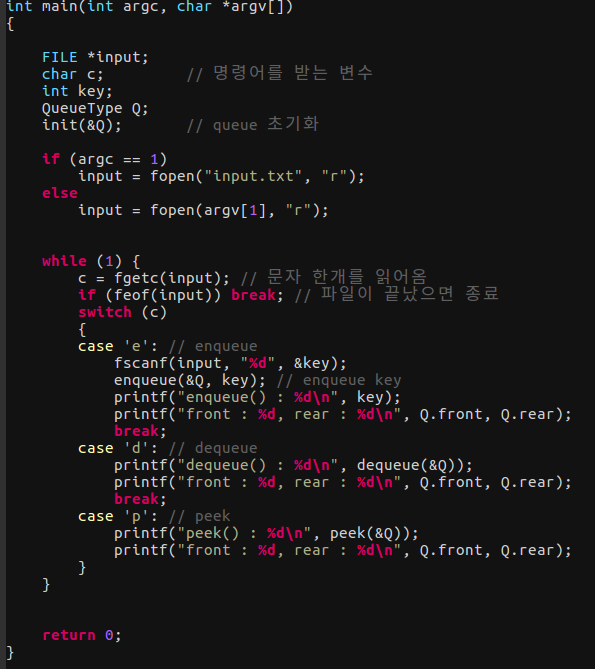
element를 삭제하기 전에 is\_empty 함수를 이용해 비어있는지부터 체크를 한다. front를 1만큼 증가시키고 그 위치에 있는 값을 반환하기만 하면 된다.

(4) peek()



dequeue와 비슷하지만 다른점은 front값을 변화시키지 않고 front에 1을 더한 값을 index로 하는 곳, 즉 가장 먼저 들어온 원소를 반환해주기만 하면 된다.

(5). main()



main 함수는 switch문을 이용해서 text file의 한 라인이 e,d,p중 어느것으로 시작되는지에 따라 달라지도록 만들었다.

2. 실행결과

