

※ 기본적으로 연결 리스트로 구현한다. 단 배열을 기반으로 구현한 경우 정확하게 구현하였다면 점수의 60%를 부여한다.

1. 스택을 구현한다. 단 스택의 시뮬레이션은 100이하의 임의의 난수 5개 정도를 만들어 스택을 채운다. 이때, 원소를 정렬된 순으로 스택에 삽입하는 함수를 추가한다. 스택의 꼭대기에 가장 큰 값이, 바닥에 가장 작은 값이 있게 만들라는 의미이다. 예를 들어 5 3 8 1 2 7의 순으로 입력되면 아래와 같이 스택에 저장된다. (5점)

					8
				8	7
			8	5	5
		8	5	3	3
	5	5	3	2	2
5	3	3	1	1	1

```
[48]
[48] [49]
[33] [48] [49]
[14] [33] [48] [49]
[14] [33] [44] [48] [49]
```

2. 원형 큐를 구현한다. 단 큐의 시뮬레이션은 100이하의 임의의 난수 7개 정도를 만들어 큐를 채운다. 그런 다음 큐의 앞에서부터 k개의 원소를 뒤집는 함수를 추가한다. k는 4로 하여 테스트하자. (8점)

생성된 큐 :	50	33	24	3	59	1	5
앞의 4개가 뒤집힌 결과 :	3	24	33	50	59	1	5

3. 다음은 피보나치 수열이다. k번째까지의 피보나치 수열을 나열하는 O(n)의 시간복잡도를 갖는 프로그램을 작성하라. 10번째까지의 수열을 나열하면 다음과 같다. (5점)

0 1 1 2 3 5 8 13 21 34 55

4. 정렬된 배열로 이진 탐색 트리를 생성하라. 배열은 `int A[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};`로 한다. 배열의 중앙값을 루트로 하여 재귀 기법을 적용하여 트리를 만든다. 트리가 만들어지면 아래의 모습이 될 것이다. 또한 트리의 탐색(노드 출력)은 반드시 실행 예시와 같은 결과가 나오도록 코드를 작성하라. (12점)

※ 실행 결과

5 2 8 1 3 6 9 4 7 10

