

CHAPTER 13

큐 구현 및 응용 실습

실습 1

문제 1)

정수를 저장하는 큐를 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 만들어보자.

push : 정수를 큐에 넣는 연산

pop : 큐에서 정수를 빼고 출력한다. 큐에 정수가 없는 경우에는 -1 출력

size : 큐에 들어있는 정수의 개수 출력.

empty : 큐가 비어있으면 1, 아니면 0 출력

front : 큐의 가장 앞에 있는 정수 출력. 큐에 정수가 없는 경우에는 -1 출력

back : 큐의 가장 뒤에 있는 정수 출력. 큐에 정수가 없는 경우에는 -1 출력

```
15
push 1
push 2
front
1
back
2
size
2
empty
0
pop
1
pop
2
pop
-1
size
0
empty
1
pop
-1
push 3
empty
0
front
3
```

실습 2

문제 2)

1번부터 N번까지 N명의 사람이 원을 이루면서 앉아있고, 양의 정수 $K(\leq N)$ 가 주어진다. 이제 순서대로 K번째 사람을 제거한다. 한 사람이 제거되면 남은 사람들로 이루어진 원을 따라 이 과정을 계속해 나간다. 이 과정은 N명의 사람이 모두 제거될 때까지 계속된다. 원에서 사람들이 제거되는 순서를 (N, K)-요세푸스 순열이라고 한다.

N과 K가 주어지면 (N, K)-요세푸스 순열을 구하는 프로그램을 작성하시오.

 Microsoft Visual Studio 디버그 콘솔

```
8 3
<3, 6, 1, 5, 2, 8, 4, 7>
```

실습 3

문제 3)

N장의 카드가 있다. 각각의 카드는 차례로 1부터 N까지의 번호가 붙어 있으며, 1번 카드가 제일 위에, N번 카드가 제일 아래인 상태로 순서대로 카드가 놓여 있다. 이제 다음과 같은 동작을 카드가 한 장 남을 때까지 반복하게 된다. 우선, 제일 위에 있는 카드를 바닥에 버린다. 그 다음, 제일 위에 있는 카드를 제일 아래에 있는 카드 밑으로 옮긴다.

N이 주어졌을 때, 제일 마지막에 남게 되는 카드를 구하는 프로그램을 작성하시오.



실습 4

문제 4)

참가인원을 입력 받고 인원 수만큼의 약실을 만든 다음, 랜덤한 위치에 총알 한 개를 넣는다. 총알이 발사되기 전까지 쏘거나 멈출 수 있다. 출력 예시는 다음과 같고, 이외의 상황은 없다고 가정한다.

```
참가 인원: 10
Shot(1) or Stop(0)
1
Shot(1) or Stop(0)
1
Shot(1) or Stop(0)
1
Shot(1) or Stop(0)
1
Shot(1) or Stop(0)
1
WASTED
Game Over!!!
```

```
참가 인원: 5
Shot(1) or Stop(0)
1
Shot(1) or Stop(0)
0
YOU ALIVE!!
```

실습 5

문제 5)

다음 그림과 같은 결과를 출력할 수 있는 덱을 구현해라

```
Add front(1)/Add Rear(2)/Delete Front(3)/Delete Rear(4)/Quit(0) : 1
Input number : 2
2
Add front(1)/Add Rear(2)/Delete Front(3)/Delete Rear(4)/Quit(0) : 1
Input number : 3
3 | 2
Add front(1)/Add Rear(2)/Delete Front(3)/Delete Rear(4)/Quit(0) : 2
Input number : 5
3 | 2 | 5
Add front(1)/Add Rear(2)/Delete Front(3)/Delete Rear(4)/Quit(0) : 3
2 | 5
Add front(1)/Add Rear(2)/Delete Front(3)/Delete Rear(4)/Quit(0) : 4
2
Add front(1)/Add Rear(2)/Delete Front(3)/Delete Rear(4)/Quit(0) : 5
Error!

Add front(1)/Add Rear(2)/Delete Front(3)/Delete Rear(4)/Quit(0) : 0
```

실습 6

문제 6)

덱을 이용하여 숫자로 이루어진 문자열이 회문인지 확인해라
이때 문자열의 최대 크기는 100이다.

Microsoft Visual Studio 디버

```
Enter the word : 12321  
회문입니다.
```

Microsoft Visual Studio 디버그 콘

```
Enter the word : 12121212  
회문이 아닙니다.
```

실습 7

문제 7)

N개의 원소를 포함하고 있는 양방향 순환 큐를 가지고 있다. 이 큐에서 몇 개의 원소를 뽑아내려고 한다. 이 큐에서 다음과 같은 3가지 연산을 수행할 수 있다.

1. 첫 번째 원소를 뽑아낸다. 이 연산을 수행하면, 원래 큐의 원소가 a_1, \dots, a_k 이었던 것이 a_2, \dots, a_k 와 같이 된다.
2. 왼쪽으로 한 칸 이동시킨다. 이 연산을 수행하면, a_1, \dots, a_k 가 a_2, \dots, a_k, a_1 이 된다.
3. 오른쪽으로 한 칸 이동시킨다. 이 연산을 수행하면, a_1, \dots, a_k 가 a_k, a_1, \dots, a_{k-1} 이 된다.

큐에 처음에 포함되어 있던 수 N이 주어진다. 그리고 지민이가 뽑아내려고 하는 원소의 위치가 주어진다. (이 위치는 가장 처음 큐에서의 위치이다.)

이때, 그 원소를 주어진 순서대로 뽑아내는데 드는 2번, 3번 연산의 최솟값을 출력하는 프로그램을 작성하시오.

```
32 6
27 16 30 11 6 23
59
```


실습 8

문제 8)

오름차순으로 정렬되는 우선 순위 큐를 만들어보자.

숫자 외 입력은 주어지지 않는다. 0을 입력하면 종료된다.

```
5
5
3
3 5
1
1 3 5
2
1 2 3 5
4
1 2 3 4 5
0
```

실습 9

문제 9)

1부터 n까지 순서대로 담긴 큐가 있을 때 하나씩 삭제해주며 걸린 시간을 구하려고 한다. 이때, 세번마다 한번씩 삭제하지 않고 다시 큐에 넣어준다.

```
Size of Queue : 6
Contents of Queue : 0 1 2 3 4 5

1 2 3 4 5
2 3 4 5
3 4 5 2
4 5 2
5 2
2 5
5
Total time : 8
```

실습 10

문제 10)

한 프린터는 다음과 같은 조건에 따라 인쇄한다.

1. 현재 Queue의 가장 앞에 있는 문서의 '중요도'를 확인한다.
2. 나머지 문서들 중 현재 문서보다 중요도가 높은 문서가 하나라도 있다면, 이 문서를 인쇄하지 않고 Queue의 가장 뒤에 재배치 한다. 그렇지 않다면 바로 인쇄를 한다.

자신의 인쇄물이 언제 뽑히는지 출력해보자.

```
number of test cases : 3
Queue size : 1 0
My paper : Priority : 5
1
Queue size : 4 2
My paper : Priority : 1 2 3 4
2
Queue size : 6 0
My paper : Priority : 1 1 9 1 1 1
5
```

실습 11

문제 11)

1부터 n 까지 있는 큐가 있을 때, front를 삭제한 후 현재 front에 있는 정수를 삭제 후 다시 큐에 넣어준다. 이 과정을 카드 한 개가 남을 때까지 반복하고, 버린 순서대로 출력한다.

```
7
1 3 5 7 4 2 6
```

실습 12

문제 12)

피자를 먹기 위해선 줄을 서야한다. N명의 사람은 배가 부르기 위한 피자 수의 수는 각자 다르다. 피자는 한 번에 한 개씩만 받을 수 있기 때문에 피자 여러 조각을 먹기 위해서는 다시 줄을 서야한다. 각 인원에게 대해서 피자를 먹기 위해 걸린 시간을 구하시오.

```
4
1 3 1 4
1 7 3 9
```