

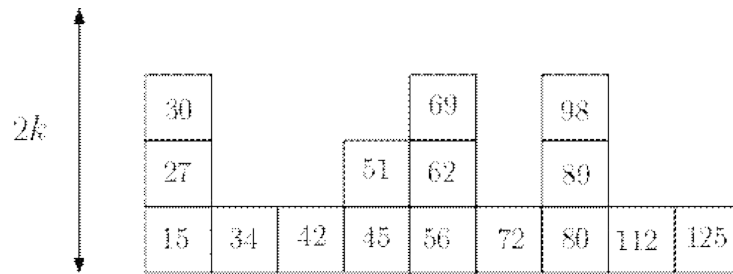


# 대기실

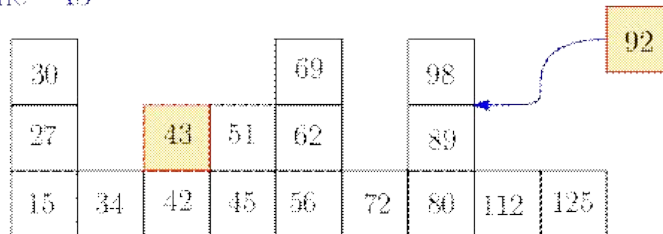
**[문제]** 한 줄에  $2k$  명이 앉을 수 있는 의자가 배치된 대기실이 있다. 이 대기실에 이미 배부된 번호표를 가진 고객이 입장하고 퇴장을 한다. 우리는 항상 고객의 대기 번호 순서대로 자리에 배정한다. 그런데 한 의자에서 최대  $2k$ 명이 앉을 수 있기 때문에 특별히 아래와 같은 배정 알고리즘을 운용한다.

1. 모든 고객은 대기 번호의 오름차순이 항상 유지되도록 배치되어야 한다.
2. 새로 들어온 고객은 특정한 의자 중간이나 제일 끝에 앉을 수 있다.
2. 만일 새로운 고객의 입장으로 의자가  $2k$ 명으로 차게 되면 순서대로 넣은 다음 이 자리를 분할(Split)하여  $k$ 명이 앉는 2개의 의자로 재조정한다.
3. 대기 도중 퇴장하는 고객이 생기면 그 자리는 빠지고 이전 고객의 순서는 그대로 유지되도록 해야 한다.
4. 만일 퇴장한 고객으로 의자가 완전히 비게 되면 그 위치의 의자는 삭제하고 뒤에 있는 의자를 모두 앞으로 당겨서 빈 곳이 없이 순서가 유지되도록 한다.

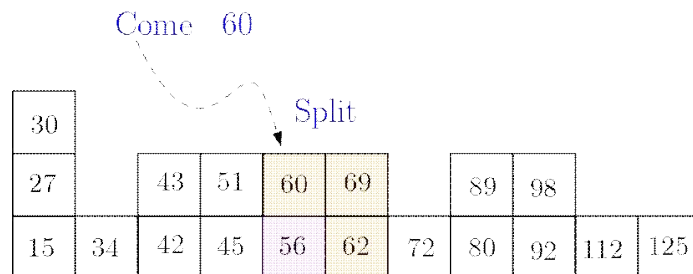
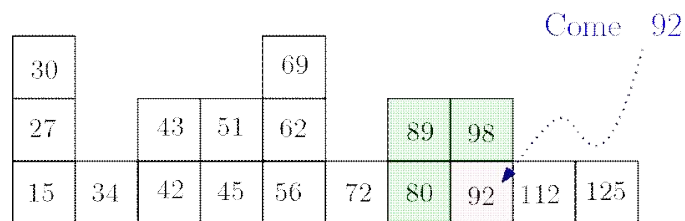
다음 예를 통하여 이 과정을 설명해보자. 이 예에서  $2k = 4$ 이다. 즉 한 의자에 최대 4명까지 앉을 수 있다. 아래 첫 그림은 한 상태를 보여주고 있다. 각 박스는 앉아있는 고객을 나타내고 그 안의 숫자는 대기번호를 의미한다. 대기실의 의자는 더 앞쪽으로 같은 의자에서는 왼쪽(아랫쪽)에 더 빠른 번호의 고객을 배치한다. 그 상태에서 '43'번 고객이 입장하면 3번째 의자의 끝에 앉게된다. 그 이후 '92'번이 입장하면 이 사람은 7번째 의자에 앉아야 한다. 그런데 이 사람이 앉게되면 의자에 딱 차게 되므로 이런 상황이 되면 split이 발생한다. 즉 [80, 89, 92, 98]이 분할(split) 하여 새로운 2개의 의자에 배치된다.



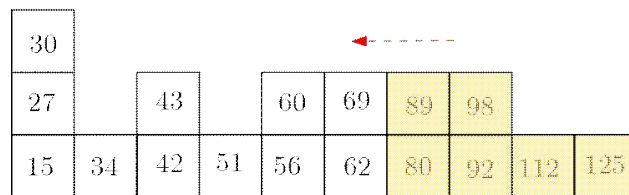
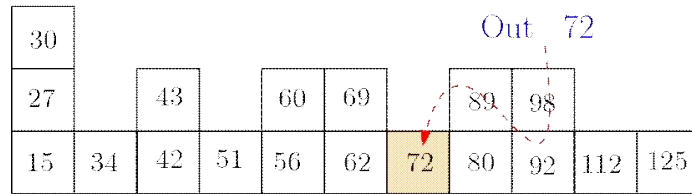
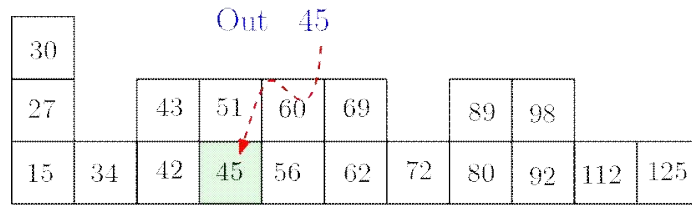
Come 43



이 상황이 다음 그림에 표시되어 있다. 그리고 이 상황에서 “60”이 추가하면 다시 SPLIT을 하여 그 아래와 같이 조정된다.



만일 이 상황에서 ‘45’, ‘72’가 퇴장을 하면 아래와 같이 변화한다. 즉 72가 퇴장하면 의자는 앞쪽으로 한 칸씩 이동하게 된다.



여러분은 N명의 고객이 입장하고 퇴장한 상황에서 최종적으로 정리된 의자의 첫 번째 번호를 순서대로 출력해야 한다. 즉 위와 같은 경우라면 [15, 34, 42, 51, 56, 62, 80, 92, 112, 125]가 되어야 한다.

**[입출력]** 표준 입출력을 사용한다. 첫 줄에 입·출입 사건(transaction)의 수  $N$ 과  $k$ 가 주어진다. 1) 그 범위는  $5 \leq N \leq 10,000$ ,  $2 \leq k \leq 10$  이다. 이어지는 N개의 줄에 들어오고 나간 고객 번호(대기표 번호)  $p_i$ 가 주어진다.  $1 \leq p_i \leq 1,000$ .

입장의 경우에는 '+  $p_i$ '로 표시되고 나간 경우에는 '-  $p_i$ '로 표시된다. 단 퇴장의 경우  $p_i$ 가 대기실에 없는 상황이라면 대기실의 변화는 없다. 경우에 따라서  $p_i = p_j$ 일 수도 있다.

여러분은 **stdin**에 제시된 N건의 입·출입 사건에 모두 종료된 경우 각 대기실 의자의 첫 번째 고객 번호를 순서대로 출력해야 한다. 즉 위 그림의 경우라면 제일 마지막 상태에서 각 의자를 표시한 slot의 제일 밑바닥에 있는, 즉 [15, 34, 42, 51, 56, 62, 80, 92, 112, 125]를 출력해야 한다. 한 줄에 하나씩 순서대로 출력한다. 단 마지막 상황에서 대기실에는 반드시 1명 이상의 고객이 있다고 가정한다.

**[예제]**

stdin	stdout
10 2 //N=10, k=2	32
+ 80	78
+ 34	101
+ 54	
+ 78	
+ 101	
- 54	
+ 32	
+ 501	
- 34	
- 200	

[제한조건] 프로그램 이름은 **waiting.{c,cpp,py,java}**이다. 허용되는 token의 수는 최대 600이다. 모범코드로 선정된 학생(최대 5명)에게는 추가의 +5점이 주어진다.