

HOMEWORK # 3

제출방법 : 프로그램 코드를 요구하는 프로그래밍을 직접해야 하는 경우를 제외하고 모든 숙제(과제)는 **손으로 답안을 작성(워드프로세서, 편집기 사용하지 않음)**해야 한다. 제출방법은 제출장소에 마감시간 이전에 직접 제출하거나, 온라인에 제출해야 하는 경우는 손으로 작성한 리포트를 스캔한 파일을 지정한 폴더에 제출한다.

1. Let B be an array of size $n \geq 6$ containing integers from 1 to $n-5$, inclusive, with exactly five repeated. Describe a good algorithm for finding the five integers in B that are repeated. (Text Book Exercise No: C-3.4)

A. 처음 나오는 $n-5$ 개 만큼의 숫자가 다음 나오는 5개의 연속된(중복된) 두에 영향을 준다. n 이 값이 무엇인지 알지 못하므로, 우선 Input에 입력된 번호들을 받아 하나의 배열로 저장한다. 이 값은 n 이 될 것이다.

1 다음, 정수형(자료형이) 변수 i 를 통해 $i=1$ 부터 $i=n-5$ 까지의 수를, 배열의 $n-5$ 번째 element 부터 $n-1$ 번째 element 까지 순회, if $i =$ 배열의 element 일시 해당 값을 반환해준다.

예를 들어, $n=9$ 일 경우에 그림과 같은 배열 B 가 생성된다.

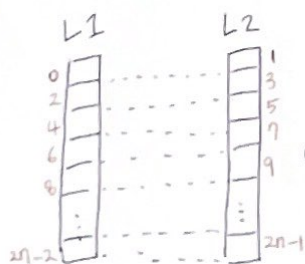
	0	1	2	3	4	5	6	7	8	null
$B :$	1	2	3	4	?	1	?	?	?	?

따라서 배열의 0번째 ~ 9-6 번째($n-6$)까지는 중복될 자연수(아정수) 등이 나타나고, 이후의 9-5 번째($n-5$) 부터 9-1 번째($n-1$) 까지는 5개의 일련의 값을 알아내야 한다. 따라서 $i=1$ 부터 $i=4$ 까지의 for문을 배열의 4, 5, 6, 7, 8 번째 elements의 값을 확인해 반복해준다. 예시로, $n=6$ 이면 5개의 중복된 값은 1 1 1 1 1 이 된다.

2. Describe a function for performing a card shuffle of an array of $2n$ elements, by converting it into two lists. A card shuffle is a permutation where a list L is cut into two lists, L_1 and L_2 , where L_1 is the first half of L and L_2 is the second half of L , and then these two lists are merged into one by taking the first element in L_1 , then the first element in L_2 , followed by the second element in L_1 , the second element in L_2 , and so on. (Text Book Exercise No: C-6.5)

A. 총 $2n$ 개의 elements 가 있고, L_1 에 n 개, L_2 에 n 개가 있게 된다.

function shuffle \hookrightarrow 는 다음과 같은 기능을 구현해야 한다.



새로 생기는 list L 은 0번째 ~ $2n-1$ 번째 원소까지 다음과 같이 정렬되어야 한다. 따라서, 주어진 shuffle func.은,

```
(List) function shuffle (List L1, List L2) {
    for (int i=0; i<n; i++) {
        L[2i] = L1[i];
        L[2i+1] = L2[i];
    }
}
```

44

와 같이

정리할 수 있다.

$L[0] = L1[0]$ $L[1] = L2[0]$
 $L[2] = L1[1]$ $L[3] = L2[1]$
 $L[4] = L1[2]$ $L[5] = L2[2]$
 \vdots
 $L[2n-2] = L1[n-1]$ $L[2n-1] = L2[n-1]$

3. 스택구조를 설명하고 스택을 위한 ADT를 설계하시오.

A. '후입선출 (Last in First Out)' 형태의 특성을 갖는 자료구조로서, 메모리에 새로 들어오는 데이터를 메모리의 앞단 (Top)에 저장하고, 내보내는 데이터 역시 말단 (Top)에서 가져간다. 스택의 ADT (추상자료형)에는 크게 push, pop이 있다.

push는 주로 입력 (input)을 담당하고, pop은 주로 출력 (output)을 담당한다.

스택구조는 주로 입력문서의 반대 순서(역순)로 출력이 이뤄질 때 많이 사용된다. 예를 들어, 뒤로가기 (Ctrl+Z), (웹 뒤로가기), 문자나 숫자 역순으로 바꾸기, 재귀 알고리즘 (영시 data 저장) 등의 경우가 있다. 스택의 ADT를 설계해보자.

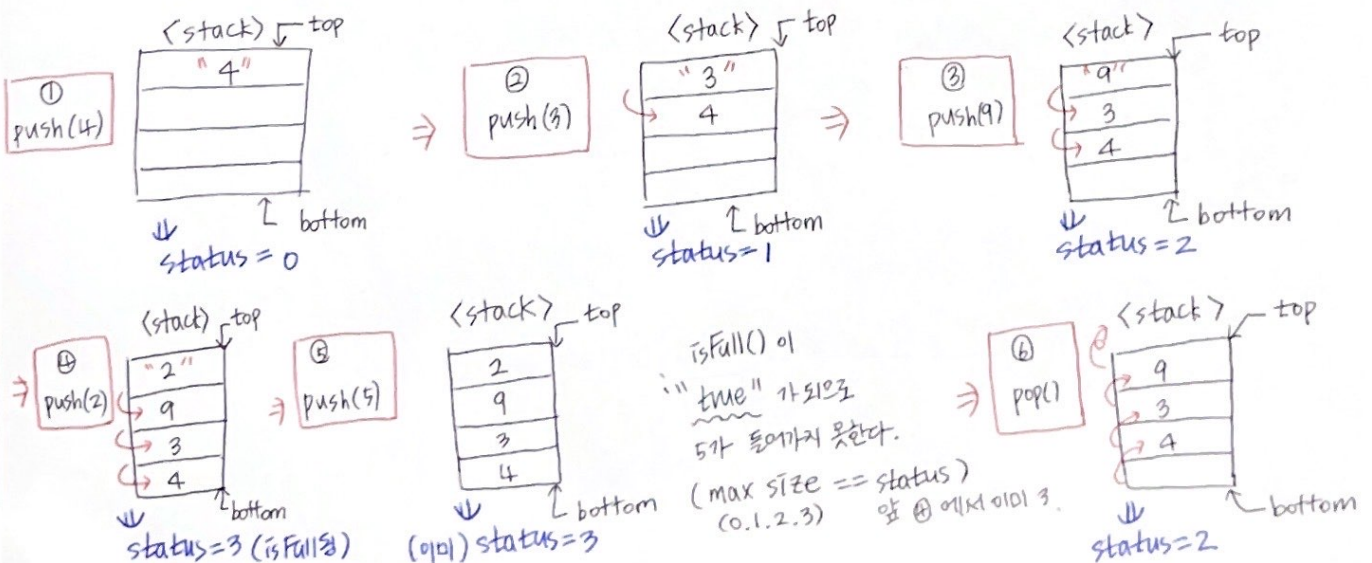
Stack의 현재 위치를 저장할 int 변수가 필요하다. 첫 데이터가 0번째 이므로, 초기에는 $\text{int status} = -1$ 로 초기화해 주고, 데이터가 하나씩 들어올 때 마다 $\text{status}++$ 를 수행하면 된다. 여기서 수행할 연산을 $\text{push}()$ 와 $\text{pop}()$ 이라고 하자.

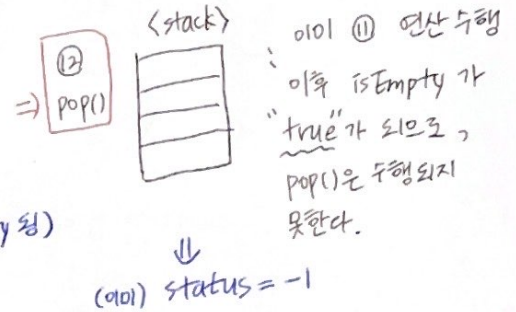
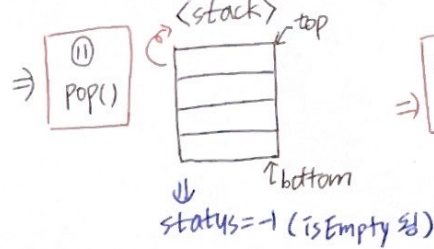
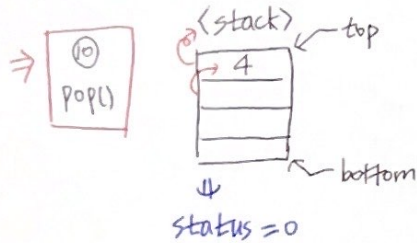
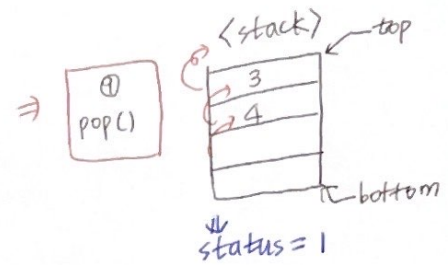
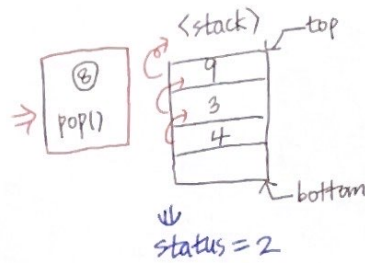
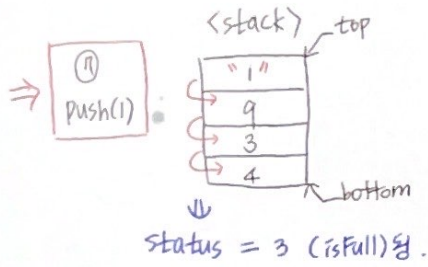
$\text{push}()$ 의 경우, $\text{push}(\text{object})$ 로 매개변수를 받아와, 스택의 가장 top부분 (마지막 다음부분)에 추가해주는 연산이다. 반대로 $\text{pop}()$ 의 경우, 마지막에 들어온 데이터 (가장 top부분)를 리턴해주는 연산이다.

스택의 $\text{Max size} == \text{status}$ 가 되면 해당 stack 구조는 다 찼다 (isFull)는 것이 되므로, 더 이상 element (데이터)가 들어가지 못하게 해준다. (+ isempty)

4. 스택의 기본적인 동작은 압력을 위한 $\text{push}(\text{obj})$ 와 출력을 위한 $\text{pop}()$ 이다. 크기가 4인 스택의 다음 동작에 의한 스택의 모양을 보시오. $\text{push}(4)$, $\text{push}(3)$, $\text{push}(9)$, $\text{push}(2)$, $\text{push}(5)$, $\text{pop}()$, $\text{push}(1)$, $\text{pop}()$, $\text{pop}()$, $\text{pop}()$, $\text{pop}()$.

A. 이 문제를 보고, 내가 3번 문제에서 구현한 stack에 isEmpty 함수를 추가해야 함을 깨달았다. isempty는 $\text{status} == -1$ 이면 true를 반환해주는 boolean 함수이다. 이것을 추가해 문제를 풀어보겠다.





정리해서 말하자면, 데이터를 집어넣는 push(Object)를 수행할 때, isFull()이 true가 되면 해당 연산은 수행되지 못하고, 반대로 데이터를 꺼내오는 pop()을 수행할 때에는 isEmpty()가 true가 되면 수행되지 않는다. else일 경우에만 각각의 연산을 수행하고, stack의 top 위치를 저장하는 status 변수에 push(obj)일 경우 +1, pop()일 경우 -1을 해준다.