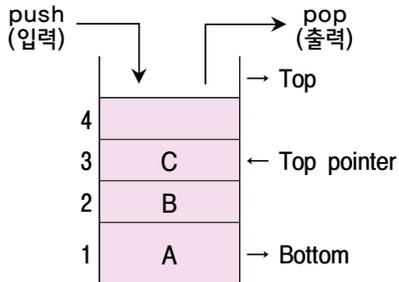


1. 스택(stack)



- 스택은 항아리와 같은 구조이다.
- 먼저 들어간 자료가 늦게 출력된다.
- 스택에서는 입력을 **push**
- 출력을 **pop**이라 한다.

// 스택 구조 및 특성

- ① 입출력이 리스트의 한쪽 끝에서만 이루어지는 제한구조(바닥이 막힌 구조)
- ② LIFO(Last In First Out) 방식으로 작동한다.
- ③ Top : 입출력이 이루어지는 리스트의 끝
- ④ Bottom : Top의 반대쪽
- ⑤ Top Pointer : 입출력되는 자료의 위치를 가리킨다.
- ⑥ 함수 호출할 때 복귀할 주소 보관, 수식 계산, 트리 구현 등에 사용된다.

// 스택 알고리즘(스택 크기가 4인 경우)

초기상태(공백상태)	top = 0	
push (입력)	<pre> If top >= 4 Then Stack_full(overflow 발생) Else top++ 자료입력 EndIf </pre>	top 포인터 증가 후, 자료입력
pop (출력)	<pre> If top <= 0 Then Stack_empty(underflow 발생) Else 자료삭제 top-- EndIf </pre>	자료삭제 후, top 포인터 감소

◆ 배열을 이용한 순차스택 구현

```
#define STACK_SIZE 5 //스택 크기는 5
char stack[STACK_SIZE]; //스택을 배열로 선언
int top = -1; //C에서는 배열의 인덱스가 0부터 시작되므로 초기 상태는 -1

int empty(){ return ( top==-1 ? 1 : 0 ); } //스택이 비어 있는지 검사
int full(){ return ( top==STACK_SIZE-1 ? 1 : 0 ); } //스택이 가득 찼는지 검사

void push(char ch){
    if( full() ) printf("\nStack Overflow!"); exit(1);
    else stack[++top] = ch; //top 포인터 증가 후 자료 입력
}

char pop(){
    if( empty() ) printf("\nStack Underflow!"); exit(1);
    else return(stack[top--]); //자료 삭제 후 top 포인터 감소
}

void scan_stack(){
    int i;
    if( empty() ){ printf("\nStackEmpty...\n"); return; }
    printf("\nTop pointer : %d",top);
    printf("\nStack Data :");
    for(i=0;i<=top;i++) printf("%2c",stack[i]);
    printf("\n");
}

void main(){
    char k, data;
    while(1){
        printf("\nPush : 1, Pop : 2 => "); k = getche();
        switch(k){
            case '1': printf("\nPush data ==> ");
                data = getche();
                push(data); break;
            case '2': printf("\nPop Data==>%c",pop()); break;
            default : printf("Error...\n"); exit(0);
        }
        scan_stack();
    }
}
```

기출문제 분석

1. 다음 일련의 스택 연산에 의해 스택에서 삭제되는 자료를 순서대로 바르게 나열한 것은? (단, push(x)는 스택에 자료 x를 삽입하는 연산이고, pop()은 스택에서 자료 한 개를 삭제하는 연산이다) [2018년 국가 7급]

push(9), pop(), push(5), push(7), push(2), pop(), push(9), pop(),
push(4), pop(), pop(), push(6), push(3), pop(), push(2), pop(), pop()

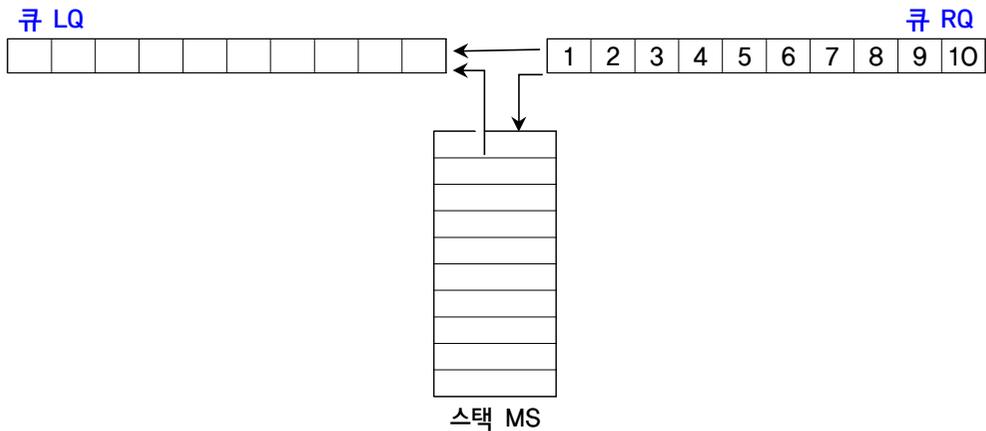
- ① 9, 5, 7, 2, 9, 4, 6, 3
- ② 9, 2, 9, 4, 7, 3, 2, 6
- ③ 2, 3, 6, 4, 9, 2, 7, 5
- ④ 9, 5, 9, 4, 7, 6, 2, 2

♣ 스택 연산

- push(9),
pop() → 9 삭제
push(5)
push(7)
push(2),
pop() → 2 삭제
push(9)
pop(), → 9 삭제
push(4)
pop() → 4 삭제
pop() → 7 삭제
push(6)
push(3)
pop() → 3 삭제
push(2)
pop() → 2 삭제
pop() → 6 삭제

• 점수 주려고 드문드문 출제되는 문제 유형이다.

2. 스택 수(stack number)란 다음 장치의 왼쪽 큐 LQ에 들어 갈 수 있는 숫자열을 말한다. 오른쪽 큐 RQ로부터 데이터가 하나씩 중간에 있는 스택 MS를 거치거나 바로 LQ로 입력될 수 있다. 즉, 데이터가 LQ에 입력될 때 RQ에서 직접 입력되거나 MS에서 한 데이터를 삭제해서 LQ에 추가할 수 있다. 다음 장치에서 생성할 수 있는 스택 수는? [2012년 국가 7급]



- ① 1 8 5 6 7 4 9 3 10 2 ② 3 4 6 7 8 5 9 2 1 10
 ③ 3 4 8 5 6 7 2 9 1 10 ④ 9 4 7 5 6 3 8 1 2 10

☞ 스택 수(stack number)

push(1), push(2), LQ 입력(3), LQ 입력(4), push(5), LQ 입력(6), LQ 입력(7), LQ 입력(8), LQ 입력(pop(5)), LQ 입력(9), LQ 입력(pop(2)), LQ 입력(pop(1)), LQ 입력(10)

정답 : ②

3. 서브프로그램(subprogram)이 호출될 때 사용되는 자료구조로 옳은 것은? [2011년 국가 7급]

- ① 연결리스트(linked list) ② 큐(queue)
 ③ 스택(stack) ④ 힙(heap)

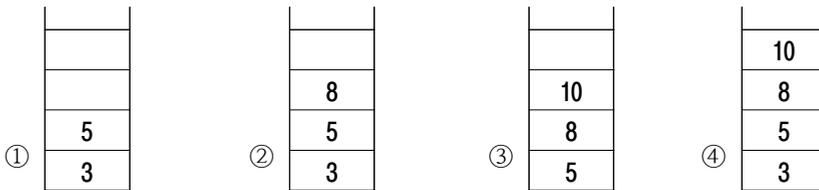
☞ 서브프로그램 호출

• 서브프로그램(subprogram)의 복귀주소는 스택에 저장한다.

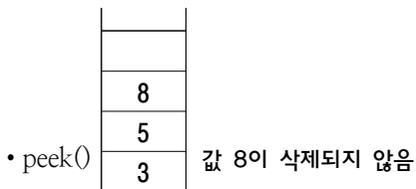
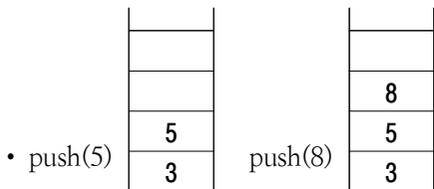
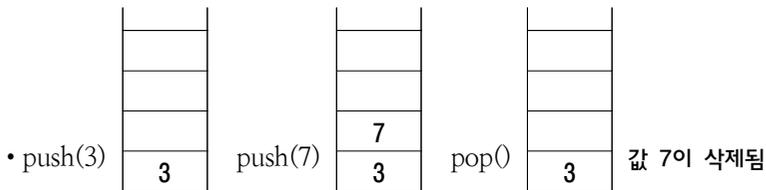
정답 : ③

4. 크기가 4인 빈 스택에 다음 연산을 차례로 수행한 후의 스택 상태를 바르게 표현한 것은?
 [2022년 국가 7급]

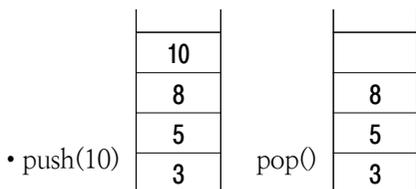
 push(3) → push(7) → pop() → push(5) → push(8) → peek() → push(10) → pop()



☞ 스택



문제에서 peek()의 기능을 제시하지 않았지만 영어단어 의미 그대로 적용해야 함 (옛보다)



정답 : ②

5. 비어있는 스택에 <보기 1>에 주어진 숫자들을 순서대로 <보기 2>의 규칙을 따라 처리할 때, 숫자들을 모두 처리한 후 스택에 저장된 숫자들을 모두 pop을 이용하여 출력한 결과로 가장 옳은 것은? (단, <보기 2>의 top은 가장 최근에 스택에 push된 값을 가리킨다) [2022년 서울 7급]

-----<보기 1>-----

1, 5, 0, 0, -2, 6, 1, -2, 0, 10

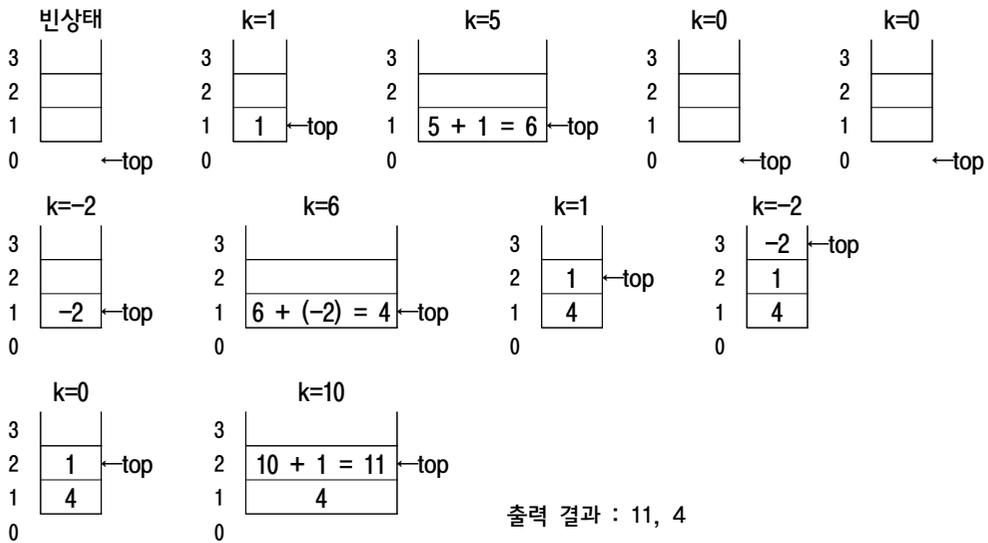
-----<보기 2>-----

- 처리하려는 숫자를 K라고 하자.
- K가 0이 아닌 경우 아래의 조건 가운데 해당하는 항목을 수행한다.
 - 스택이 비어있거나 top이 가리키는 숫자가 K보다 큰 경우 : K를 push한다.
 - top이 가리키는 숫자가 K보다 작거나 같은 경우 : pop 한 뒤 추출된 숫자에 K를 더한 값을 push한다.
- K가 0인 경우 아래 조건 가운데 해당하는 항목을 수행한다.
 - 스택이 비어있는 경우 : 다음 숫자를 처리한다.
 - 스택이 비어있지 않은 경우 : pop한다.

- ① 10 ② 10 -2 1 4
 ③ 10 6 3 1 ④ 11 4

☞ 스택

// 주어진 조건에서 스택은 다음처럼 작동된다.



6. 다음과 같이 배열을 이용하여 스택 자료구조를 정의하였다. ㉠, ㉡에 들어갈 내용을 바르게 연결한 것은? [2021년 국가 7급]

```

-----
#define MAX 10
int stack[MAX];
int top = -1;
int push(int t) {
    if (top >= MAX - 1) {
        printf("\n Stack overflow.");
        return -1;
    }
    stack[ ㉠ ] = t;
    return t;
}
int pop(void) {
    if (top < 0) {
        printf("\n Stack underflow.");
        return -1;
    }
    return stack[ ㉡ ];
}
-----

```

- | | |
|---------|-------|
| ㉠ | ㉡ |
| ① ++top | --top |
| ② ++top | top-- |
| ③ top++ | --top |
| ④ top++ | top-- |

☞ 스택에서 입출력

push	<ul style="list-style-type: none"> • top 포인터 증가 후, 자료입력 • stack[++top] = t;
pop	<ul style="list-style-type: none"> • 자료삭제 후, top 포인터 감소 • return stack[top--];

7. 스택을 일차원 배열 `stack[0:99]`를 이용하여 구현하고자 한다. `top` 변수의 초기값은 `-1`로 설정하며, `top`이 `-1`이면 스택은 비어 있다는 것을 나타낸다. C 언어를 이용하여 스택에 데이터를 추가하는 함수 `push()`와 삭제하는 함수 `pop()`을 다음과 같이 작성하였을 때, ㉠과 ㉡에 들어갈 내용은? [2009년 국가 7급]

```
void push(int *top, int data){ //스택에 데이터를 추가
    if(*top >= 99){ stack_full(); return; }
    ㉠ ;
}

int pop(int *top){ //스택에서 데이터를 삭제 후 반환
    if(*top == -1){ return stack_empty(); }
    ㉡ ;
}
```

- ① ㉠ `stack[*top++]` ㉡ `return stack[--*top]`
- ② ㉠ `stack[*top++]` ㉡ `return stack[*top--]`
- ③ ㉠ `stack[++*top]` ㉡ `return stack[*top--]`
- ④ ㉠ `stack[++*top]` ㉡ `return stack[--*top]`

♣ 일차원 배열 `stack[0:99]`를 이용한 스택 구현 - C 언어 이용

- 주어진 문제에 맞도록 C로 구현하면 다음과 같다.
- 그런데, 주어진 문제에서 출제자는 스택에 데이터를 추가하는 함수인 `push()`에서 `stack[++*top] = data;` 부분을 잘못 기술하였다.

```
void main(){
    int stack[100];
    int top = -1;
    :
    push(&top, 100); //스택에서 top을 1 증가시킨 후 데이터 100을 입력
    :
    pop(&top); //스택에서 top이 가리키는 위치의 데이터를 출력 후, 1 감소
    :
}

void push(int *top, int data){ //스택에 데이터를 추가
    if(*top >= 99){ stack_full(); return; }
    stack[++*top] = data; //㉠ 출제자가 data를 빠트렸다.
}

int pop(int *top){ //스택에서 데이터를 삭제 후 반환
    if(*top == -1){ return stack_empty(); }
    return stack[(*)top--]; //㉡ 이 부분은 (*top)--처럼 괄호가 있어야 한다.
} //실수인지? 잘 모르는 것인지?
```


9. 스택, 큐, 리스트 자료구조에 대한 설명으로 옳지 않은 것은? [2009년 국가 7급]

- ① 연결리스트는 배열구조와 비교하여 삽입, 삭제 시간이 많이 걸리고 메모리 낭비가 많다.
- ② 주로 힙(heap) 구조로 구현되는 우선순위 큐(priority queue)의 데이터 삽입은 임의로 이루어지나, 데이터의 삭제는 가장 큰 값이나 작은 값부터 수행된다.
- ③ 스택은 함수 호출, 후위식 연산 등에 유용하게 사용되며, 연결리스트로 구현할 수 있다.
- ④ 큐는 선입선출 방식의 동작을 수행하므로 잡 스케줄링(job scheduling) 등에 유용하게 사용된다.

☞ 연결리스트와 배열구조

- 연결리스트에서 자료 삽입, 삭제는 포인터 변경만으로 빠르게 수행할 수 있다.
 - 하지만, 배열구조에서 자료 삽입, 삭제는 자료 이동이 필요하므로 시간이 더 필요하다.
-

정답 : ①

10. 스택에 정수 1, 2, ..., n을 순서대로 push하면서 임의로 pop을 하게 되면 여러 가지 수열을 얻을 수 있다. 예를 들어, n=2일 때 push(1), pop(), push(2), pop() 연산을 차례대로 하면 수열 (1, 2)를 얻을 수 있고, push(1), push(2), pop(), pop() 연산을 차례대로 하면 수열 (2, 1)을 얻을 수 있다. 이와 같은 방법에 의해 수열을 얻을 때 다음 중에서 n=4인 경우 얻을 수 없는 수열은? [2021년 군무원 7급]

- ① (1, 2, 4, 3)
- ② (3, 4, 2, 1)
- ③ (3, 2, 1, 4)
- ④ (4, 2, 3, 1)

☞ 스택

- ① (1, 2, 4, 3) : push(1), pop(), push(2), pop(), push(3), push(4), pop(), pop()
 - ② (3, 4, 2, 1) : push(1), push(2), push(3), pop(), push(4), pop(), pop(), pop()
 - ③ (3, 2, 1, 4) : push(1), push(2), push(3), pop(), pop(), pop(), push(4), pop()
 - ④ (4, 2, 3, 1) : 불가
-

정답 : ④

11. 다음 자료구조 중에서 비선형구조로만 묶은 것은? [2011년 국가 7급] [2019년 군무 7급]

ㄱ. 스택(stack)	ㄴ. 트리(tree)
ㄷ. 연결리스트(linked list)	ㄹ. 그래프(graph)

- ① ㄱ, ㄴ ② ㄱ, ㄷ
③ ㄴ, ㄷ ④ ㄴ, ㄹ

☞ 선형구조 / 비선형구조

-
- 선형구조 : 스택(stack), 연결리스트(linked list)
 - 비선형구조 : 트리(tree), 그래프(graph)
-

정답 : ④