

[5장 연습문제]

01. 다음 중 스택에 대한 옳은 내용으로만 나열된 것은?

- ㄱ. FIFO 방식으로 처리된다.
- ㄴ. 순서 리스트의 뒤에서 노드가 삽입되며 앞에서 노드가 제거된다.
- ㄷ. 선형 리스트의 양쪽 끝에서 삽입과 삭제가 모두 가능한 자료구조이다.
- ㄹ. 인터럽트 처리, 서브루틴 작업 등에 응용된다.

- ① ㄱ, ㄴ
- ② ㄴ, ㄷ
- ③ ㄹ
- ④ ㄱ, ㄴ, ㄷ, ㄹ

02. 스택 메모리에 대한 정보의 입출력 방식은?

- ① FIFO
- ② FILO
- ③ LILO
- ④ LIFO

03. 스택의 응용 분야와 거리가 먼 것은?

- ① 운영체제의 작업 스케줄링
- ② 함수 호출의 순서 제어
- ③ 인터럽트 처리
- ④ 수식 계산

04. 서브프로그램이 호출될 때 사용되는 자료구조로 옳은 것은?

- ① 연결 리스트
- ② 큐
- ③ 스택
- ④ 힙

05. 다음은 스택에 자료를 삽입하는 알고리즘이다. 괄호에 적합한 내용은?

```

procedure Insert(data, n, top, Stack)
  if top ≥ n then call Stack - Full;
  top = top + 1;
  Stack(top) = (      );
end Insert

```

- ① top
- ② data
- ③ top-1
- ④ data-1

06. 다음 문장에서 괄호에 들어갈 단어는?

A () is an ordered list in which all insertions and deletions are made at one end, called the top.

- ① stack
- ② queue
- ③ list
- ④ tree

07. 데이터의 삽입과 삭제가 top이라고 부르는 한쪽 끝에서만 이루어지는 후입선출 형태의 자료구조는?

- ① 스택
- ② 큐
- ③ 데크
- ④ 원형 큐

08. 스택에 대한 설명으로 옳은 것은?

- ① 인터럽트 처리, 서브루틴 호출 작업 등에 응용된다.
- ② FIFO 방식으로 처리된다.
- ③ 순서 리스트의 뒤(Rear)에 노드가 삽입되며, 앞(Front)에서 노드가 제거된다.
- ④ 선형 리스트의 양쪽 끝에서 삽입과 삭제가 모두 가능한 자료구조이다.

09. 스택 알고리즘에서 T가 스택 포인터이고, m이 스택 길이일 때, 서브루틴 "AA"가 처리해야 하는 것은?

```
T ← T + 1
if T > m then goto AA
else STACK(T) ← item
```

- ① 오버플로 처리 ② 언더플로 처리
- ③ 삭제 처리 ④ 삽입 처리

10. Which of the following is a linear list in that elements are accessed, created and deleted in a last-in-first-out order?

- ① Queue ② Graph
- ③ Stack ④ Tree

11. 스택의 자료 삭제 알고리즘이다. 괄호 안에 들어갈 내용으로 가장 적합한 것은? (단, Top은 스택 포인터이고, S는 스택 이름이다.)

```
if Top = 0
then( )
else {
    Remove S(Top)
    Top = Top - 1
}
```

- ① Over flow ② Top = Top+1
- ③ Under flow ④ Top = Top-2

12. 스택 컴퓨터의 특성에 대한 설명으로 옳지 않은 것은?

- ① 피연산자를 나타내지 않기 때문에 인스트럭션의 길이가 짧아 기억 공간의 이용이 효율적이다.
- ② 스택에 기억된 데이터만 이용해 연산하므로 인스트럭션 수행 시간이 짧다.

13. 순서가 A, B, C, D로 정해진 입력 자료를 스택에 입력하였다가 출력한 결과로 가능한 것은?

14. 순서가 A, B, C, D인 자료를 스택에 입력 후 출력한 결과로 옳지 않은 것은?

15. 스택에서 A, B, C, D 순서로 입력한 자료를 Push→Push→Pop→Push→Pop→Push→Pop→Pop으로 연산할 때 출력은?

16. 스택 S에서 B, A, D, C 순서로 입력 후 A, B, C, D 순서로 출력하기 위한 push와 pop의 횟수는?

- 4 -

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

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

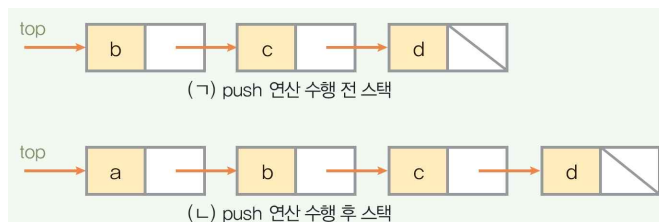
```
int pop(int * top) { // 스택에서 데이터를 삭제 후 반환
    if (*top == -1) { return stack _ empty(); }
    ㉡;
}
```

㉠

㉡

- | | |
|------------------------------|-----------------------------------|
| ① <code>stack[*top++]</code> | <code>return stack[--*top]</code> |
| ② <code>stack[*top++]</code> | <code>return stack[*top--]</code> |
| ③ <code>stack[++*top]</code> | <code>return stack[*top--]</code> |
| ④ <code>stack[++*top]</code> | <code>return stack[--*top]</code> |

18. 연결 리스트를 이용해 스택을 표현한 것이다. 이에 대한 설명으로 옳지 않은 것은? (단, `push`는 스택에 자료를 삽입하는 연산이고, `pop`은 스택에서 자료를 삭제하는 연산이다.)



- ① 스택에 가장 최근에 입력된 자료는 `top`이 지시한다.
- ② 스택에 입력된 자료 중 `d`가 가장 오래된 자료이다.
- ③ (㉡)에서 자료 `c`를 가져오려면 `pop` 연산이 2회 필요하다.
- ④ (㉠)에서 자료의 입력된 순서는 `d, c, b`이다.

19. 연결 리스트 스택의 삭제 알고리즘이다. ㉠~㉣에 들어갈 문장으로 바르게 짝지

어진 것은? (단, 스택이 비었을 때 top은 NULL이고, top은 스택의 최상위 노드를 가리키는 포인터이다. 또한 스택이 빈 상태에서 맨 처음 삽입되는 노드의 next는 NULL이다.)

```
typedef struct node {
    int key;
    struct node *next;
} NODE;           // 스택의 노드 구조
NODE *top = NULL; // 스택의 최상위 노드를 가리키는 포인터
int pop(void) {
    NODE *temp;
    int ret;
    if (㉠) {
        printf("\n Stack underflow.");
        return -1;
    }
    ret = top->key;
    ㉡;
    ㉢;
    free(temp);
    return ret;
}
```

㉠

㉡

㉢

- | | | |
|----------------------|-------------------|--------------------------|
| ① top → next == NULL | temp = top → next | top = top → next |
| ② top → next == NULL | temp = top | top → next = temp → next |
| ③ top == NULL | temp = top | top = top → next |
| ④ top == NULL | temp = top → next | top → next = temp → next |

20. 다음 산술식을 전위 표기법으로 옳게 표현한 것은?

a * (b + c) * d

- | | |
|-------------|-------------|
| ① **a + bcd | ② * + a*bcd |
| ③ abc* + d* | ④ abc + *d* |

21. 다음 중위 표기법의 수식을 후위 표기법으로 옳게 변환한 것은?

A = (B - C) * D + E

① $=A^*-BC+DE$

② $=A++-BCDE$

③ $ABC-D^*E+=$

④ $ABC^*D-E+=$

22. 다음 중위 표기법의 수식을 후위 표기법으로 옳게 변환한 것은?

중위 표기식: $a * (b - c / 5) + (d - 8 * e / 5)$

① $abc5/*-d8-e*5/+$

② $abc5/-*d8e*5/-+$

③ $ab*c5/-+d8-e*5/$

④ $abc5-/*d8e5-*/+$

23. 중위 표기법으로 표현된 다음 수식을 후위 표기법으로 옳게 표현한 것은?

$a + (b * c - d) * (e - f * g) - h$

① $ab*cd+efg^*-h-$

② $abc*d+ef*g^*-h-$

③ $abcd^*-efg^*+^*-h-$

④ $abc*d-efg^*-^*+h-$

24. 다음 수식을 후위 표기법으로 표현할 때 옳은 것은?

$(((A/B) + C) - (D * E))$

① $A/B+C-D^*E$

② $AB/C+DE^*-$

③ $A/B+C-*DE$

④ $AB/C+-DE^*$

25. 다음 수식을 후위 표기법으로 표현할 때 옳은 것은?

$(A * B) + (C * D)$

① $+AB**CD$

② $+*AB*CD$

③ $AB*CD*+$

④ $*AB+*CD$

26. 영문의 괄호 안에 적합한 수식의 표현은?

The reverse Polish notation is in a form suitable for stack manipulation. The expression $(A+B)*(C+D)$ is written in reverse Polish notation as ().

① $AB+CD*+$

② $AB+CD*+$

③ $+AB+CD*$

④ $*+AB+CD$

27. 다음 중위 표기법 수식을 후위 표기법 수식으로 옳게 변환한 것은?

중위 표현식 : $Y = (1 + ((3 + 4 / 2) * 5)) - 6$

① $Y134256*/++=$

② $Y=1342/+5*+6-$

③ $=Y134/2*5++6-$

④ $Y1342/+5*+6-=-$

28. 중위 표기법으로 나타낸 다음 식을 후위 표기법으로 옳게 표현한 것은?

$(7 + 6 / 2) / 2 + 9 * 4 / 3$

① $6/2+7/29*4/3+$

② $62/7+2/943*/+$

③ $76/2+2/94*3/+$

④ $762/+2/943/+*$

⑤ $762/+2/94*3/+$

29. 다음 전위 표기식의 계산 결과는?

$+ - 5 4 \times 4 7$

① - 19

② 7

③ 28

④ 29

30. 다음 전위 표기법 수식을 후위 표기법 수식으로 변환할 때 변환된 수식의 네 번째 연산자는?

(전위식) $--AB * / CDEF$

- ① * 연산자
- ② + 연산자
- ③ - 연산자
- ④ / 연산자

31. 후위 표기법으로 나타낸 다음 식을 전위 표기법으로 옳게 표현한 것은?

$ABC + D / - AE + BF * / +$

- ① $--A/+BCD/+AE*BF$
- ② $--A/BC+D/+AE*BF$
- ③ $--A/+BCD/+*AEBF$
- ④ $+A-/ +BCD/+AE*BF$
- ⑤ $--+/A+BCD/+*AEBF$

32. 다음 수식을 후위 표기법으로 변환한 후 스택을 이용해 수식을 계산하고자 한다. 수식 계산 과정에서 스택에 일곱 번째로 push되는 값은?

$(A - B) * C - (D + E) / F$

- ① D
- ② E
- ③ D+E
- ④ $(A-B) * C$

33. 중위 표기법 수식과 그에 대응하는 후위 표기법 수식이 일치하지 않는 것은?

- | <u>중위 표기법</u> | <u>후위 표기법</u> |
|-------------------|---------------|
| ① $A * B + C / D$ | $AB * CD / +$ |

- | | |
|-------------------|---------------|
| ② $A+(B+C)/D-E$ | $ABC+D/+E-$ |
| ③ $A * B/(C-D)+E$ | $AB * CD-E/+$ |
| ④ $A/B-C+D * E$ | $AB/C-DE * +$ |

34. 다음 수식을 후위 표기법으로 변환했을 때, 네 번째 연산자는?

$$1 + (2 - 3 * 4 / (5 + 6)) - 7$$

- ① + ② - ③ * ④ /

35. 주변에서 LIFO 방식의 스택의 예를 찾아 설명하시오.

36. 다음은 A, B, C, D로 만들어진 후위 표기법 수식이다. 이 수식을 스택을 이용해 계산할 때, 스택에 여섯 번째로 저장되는 값은?

$$\text{후위 표기식: } A B C + D * -$$

- | | |
|---------------|-------------------|
| ① * | ② (B + C) |
| ③ (B + C) * D | ④ A - (B + C) * D |

37. 다음 후위 표기법 연산식의 결과로 옳은 것은?

$$3\ 4\ *\ 5\ 6\ *\ +$$

38. 스택에서 오버플로가 발생하는 경우와 오버플로를 해결하는 방법을 설명하시오.

39. 다음 수식을 후위 표기법으로 변환하시오.

(가) $A+B-C-D$

(나) $(A+B) * C+D$

(다) $A/B/C/D$

40. 다음 후위 표기법 수식을 스택으로 연산하는 과정을 설명하시오.

$AB-C-D+$

41. 스택을 이용해 네 개 자료 A, B, C, D에 대하여 A, C, B, D의 출력이 나오는 과정을 삽입과 삭제 연산을 사용해 설명하시오.

42. 재귀 함수를 호출한 후에 복귀 주소를 관리할 때 가장 적당한 자료구조가 무엇인지 이유를 설명하시오.

43. [예제 4-2]에서 `reverse()` 연산은 단순 연결 리스트의 노드 순서를 역순으로 바꾸기 위해 포인터 세 개 p, q, r을 추가로 사용하였다. 포인터를 사용하지 않고 스택을 사용해 `reverse()` 연산을 구현하시오.