

1. <보기>의 C언어 함수(func)에서 func(4)+func(9)를 수행한 결과 값은?

<보기>

```
int func(int n) {
    int res = 0;
    if(n <= 1) res = n;
    else {
        int fn, fnm2 = 0, fnm1 = 1;
        for(int i = 2; i <= n; i++) {
            fn = fnm1 + fnm2;
            fnm2 = fnm1;
            fnm1 = fn;
        }
        res = fn;
    }
    return res;
}
```

- ① 23
- ② 37
- ③ 58
- ④ 94

2. <보기>는 알고리즘의 시간 복잡도(time complexity)를 표시하는 방법 중 하나인 빅오(big-oh) 표기법에 대한 정의이다. 빈칸 (가)와 (나)에 각각 들어갈 수식은?

<보기>

두 개의 함수 $g(n)$ 과 $f(n)$ 에 대해서 $O(g(n))$ 은 다음과 같이 정의한다.

$O(g(n)) = \{f(n) : \text{모든 (가)에 대해서 (나)인 양의 상수 } c \text{와 } n_0 \text{가 존재한다.}\}$

- | | (가) | (나) |
|---|--------------|----------------------------------|
| ① | $n \geq n_0$ | $0 \leq f(n) \leq c \times g(n)$ |
| ② | $n_0 \geq n$ | $0 \leq f(n) \leq c \times g(n)$ |
| ③ | $n \geq n_0$ | $0 \leq g(n) \leq c \times f(n)$ |
| ④ | $n_0 \geq n$ | $0 \leq g(n) \leq c \times f(n)$ |

3. 1부터 50까지의 정수가 저장된 이진 탐색 트리(binary search tree)에서 루트노드부터 29가 저장된 노드를 탐색하려고 한다. 탐색하는 과정 중에 방문한 노드들의 순서로 옳지 않은 것은?

- ① 40 → 5 → 39 → 25 → 29
- ② 25 → 30 → 26 → 27 → 29
- ③ 10 → 30 → 15 → 17 → 29
- ④ 20 → 35 → 17 → 33 → 29

4. 큐(queue)를 적용할 수 있는 상황이 아닌 것은?

- ① 도착한 순서대로 프린트 처리
- ② 비동기적 데이터 전송(파일 입·출력, 파이프, 소켓)
- ③ 프로세스 스케줄링
- ④ 텍스트 에디터에서의 작업 취소(undo) 과정

5. <보기>는 가중치가 부여된 무방향 그래프(weighted undirected graph)를 인접 행렬(adjacency matrix)로 표현한 것이다. 이 그래프에 대한 최소 비용 신장 트리(minimum spanning tree)에서 간선(edge)의 가중치를 모두 더한 값은?

<보기>

	A	B	C	D	E
A	0	8	3	∞	∞
B	8	0	5	4	∞
C	3	5	0	2	6
D	∞	4	2	0	7
E	∞	∞	6	7	0

- ① 14
- ② 15
- ③ 16
- ④ 17

6. C언어로 작성된 <보기>의 함수는 입력된 문자열을 검사하여 0 또는 1을 반환한다. 이 함수의 인자로 넣었을 때 1을 반환하는 문자열은? (단, push()는 문자 하나를 스택에 삽입하는 함수이고, pop()은 문자 하나를 스택에서 추출하는 함수이다.)

<보기>

```
int isTrue(char str[])
{
    int length = strlen(str);
    for (int i = 0; i < length; i++)
        push(str[i]);
    for (int i = 0; i < length; i++)
        if (pop() != str[i])
            return 0;

    return 1;
}
```

- ① AAAAAAaaaaaaa
- ② aaaAAAaaaAAA
- ③ AAAaaaaaaaAAA
- ④ AaAaAaAaAaAa

7. <보기>는 단순 연결 리스트(singly linked list)에서 원소들의 순서를 역순으로 변환하는 C언어 프로그램이다. (가), (나), (다)에 들어갈 내용을 옳게 짝지은 것은?

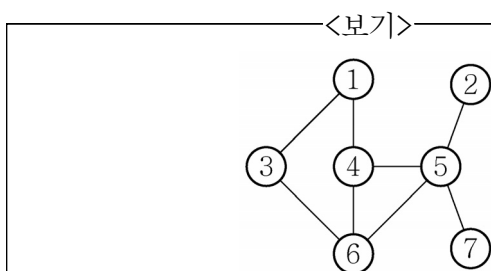
<보기>

```
typedef struct node {
    int key;
    struct node *link;
} ListNode;

ListNode *reverse(ListNode *L)
{
    ListNode *p, *q, *r;
    p = L;
    q = NULL;
    r = NULL;
    while (p != NULL) {
        (가);
        (나);
        p = p->link;
        (다);
    }
    return q;
}
```

- | | (가) | (나) | (다) |
|---|-------|-------|-------------|
| ① | q = p | r = q | q->link = r |
| ② | r = q | q = p | q->link = r |
| ③ | r = q | q = p | r = q->link |
| ④ | q = p | r = q | r = q->link |

8. 시작점이 1일 때, <보기>의 그래프에 대한 너비 우선 탐색(breadth first search)의 방문 순서를 바르게 나열한 것은? (단, 같은 우선순위의 정점들은 숫자가 작은 정점을 먼저 방문한다.)



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

9. 이진 트리(binary tree)를 배열로 저장할 때 <보기 1>과 같은 특성을 갖는다고 가정하자. 이진 트리를 <보기 2>와 같은 형태의 배열로 저장했을 때, 이 트리에 대한 설명으로 가장 옳은 것은?

- <보기 1>
- 배열 인덱스 i 에 저장된 노드의 부모노드 인덱스는 $\lfloor i/2 \rfloor$
 - 배열 인덱스 i 에 저장된 노드의 왼쪽 자식노드 인덱스는 $2i$
 - 배열 인덱스 i 에 저장된 노드의 오른쪽 자식노드 인덱스는 $2i+1$

<보기 2>

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
	20	7	28	1	15		30			11	17				36

- ① 이 트리는 포화 이진 트리(full binary tree)이다.
 ② 이 트리는 완전 이진 트리(complete binary tree)이다.
 ③ 단말 노드(leaf node) 개수는 3개이다.
 ④ 이 트리는 이진 탐색 트리(binary search tree)이다.

10. 공백 AVL 트리에 키(key)를 차례대로 저장하여 AVL 트리를 구축한다고 할 때, 생성되는 AVL 트리의 모양이 다른 것은?

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

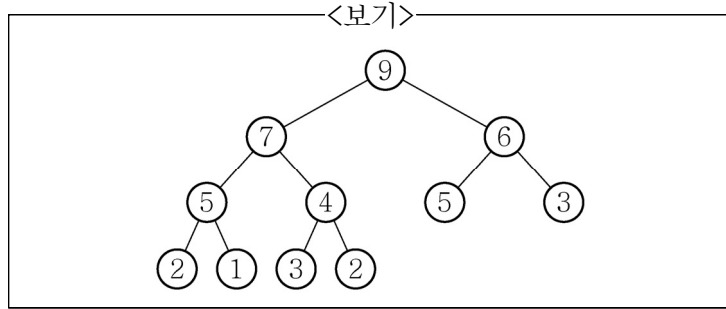
11. <보기>의 중위 표기식(infix expression)을 전위 표기식(prefix expression)으로 변환한 것으로 옳은 것은?

<보기>

$((x/y) - z) + (u * v * w) - ((x * z) / u)$

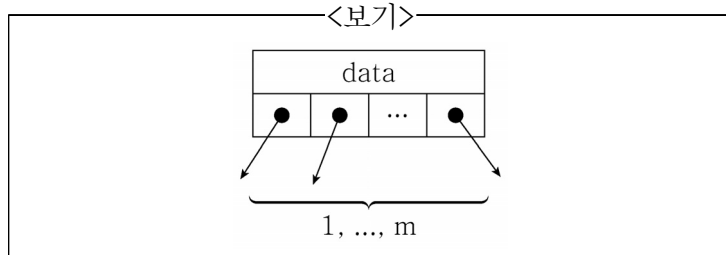
- ① $-+-/xyz**uvw/*xzu$
 ② $-+-/xyz**uvw*/xzu$
 ③ $-/xyz+**uvw/*-xzu$
 ④ $-/xyz+**uvw-/xzu$

12. <보기>의 최대 힙(max heap)에 8을 삽입한 후, 최댓값을 삭제하는 연산을 수행하였다. 이때, 루트 노드의 좌·우 자식노드의 값은?



	좌	우
①	7	5
②	7	6
③	7	8
④	8	7

13. m차 트리(m-ary tree)는 각 노드(node)의 자식 노드가 최대 m개인 트리를 의미한다. <보기>의 그림과 같은 연결 리스트(linked list)로 구현된 m차 트리의 노드의 수가 n개일 때, 이 트리의 널 포인터(null pointer)의 총 개수는?



- ① $n(m+1)+1$
 ② $n(m+1)-1$
 ③ $n(m-1)+1$
 ④ $n(m-1)-1$

14. 최악의 경우 시간 복잡도가 $O(n \log n)$ 인 정렬 알고리즘은?

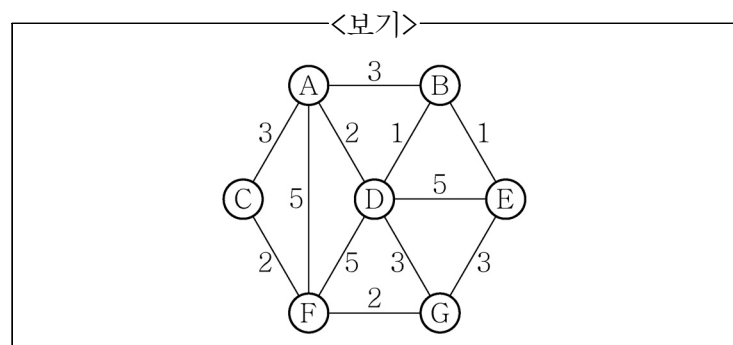
- ① 삽입 정렬(insertion sort)
 ② 퀵 정렬(quick sort)
 ③ 합병 정렬(merge sort)
 ④ 버블 정렬(bubble sort)

15. 덱(deque: double ended queue)에 대해 <보기>와 같이 4가지 연산을 사용할 수 있다고 하자. 덱의 연산을 사용하여 스택과 큐를 구현한다고 할 때, 필요한 덱의 연산을 옳게 짝지은 것은? (여기서 덱을 가리키는 포인터를 d, 스택을 가리키는 포인터를 s, 큐를 가리키는 포인터를 q라고 한다.)

- <보기>
- insert_front(d, item): 덱의 첫 번째 원소로 item 삽입
 - insert_rear(d, item): 덱의 마지막 원소로 item 삽입
 - delete_front(d): 덱의 첫 번째 원소 삭제
 - delete_rear(d): 덱의 마지막 원소 삭제

	<u>스택의 연산</u>		<u>큐의 연산</u>	
	push(s, item)	pop(s)	enqueue(q, item)	dequeue(q)
①	insert_rear (d, item)	delete_rear (d)	insert_rear (d, item)	delete_front (d)
②	insert_rear (d, item)	delete_front (d)	insert_rear (d, item)	delete_rear (d)
③	insert_front (d, item)	delete_front (d)	insert_front (d, item)	delete_front (d)
④	insert_front (d, item)	delete_rear (d)	insert_front (d, item)	delete_rear (d)

16. 다익스트라(Dijkstra)의 최단 경로 알고리즘을 이용하여 <보기>의 그래프의 정점 E에서 각 정점으로의 최단 경로를 구할 때, 정점 E를 선택한 이후 세 번째로 선택되는 정점은? (단, 각 간선 위의 숫자는 양 정점들 사이의 거리를 의미한다.)



- ① A ② C ③ D ④ G

