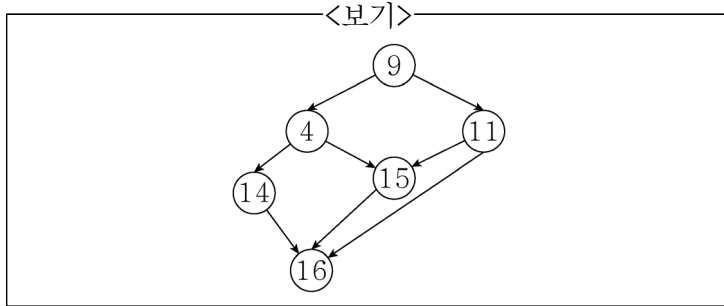


1. <보기>의 그래프에 대해 위상 정렬한 결과로 가장 옳지 않은 것은?



- ① 9 - 11 - 15 - 4 - 14 - 16
 ② 9 - 4 - 14 - 11 - 15 - 16
 ③ 9 - 11 - 4 - 14 - 15 - 16
 ④ 9 - 11 - 4 - 15 - 14 - 16

2. <보기>의 C언어 문장을 사용하여 이중 연결 리스트(doubly linked list)에서 p가 가리키는 노드 다음에 new_node가 가리키는 새로운 노드를 삽입하려고 한다. 이를 수행하기 위한 문장 순서로 가장 옳지 않은 것은? (단, 노드에서 이전/다음 노드를 가리키는 포인터는 각각 prev/next이고, 새 노드 삽입 전에 p가 가리키는 노드 다음에는 다른 노드가 반드시 존재한다.)

<보기>

(가) new_node->prev = p;
 (나) new_node->next = p->next;
 (다) p->next->prev = new_node;
 (라) p->next = new_node;

- ① (가) - (나) - (다) - (라)
 ② (가) - (나) - (라) - (다)
 ③ (다) - (나) - (가) - (라)
 ④ (다) - (나) - (라) - (가)

3. <보기>의 C언어 재귀 함수는 크기가 n인 영문과 숫자로 구성된 문자 배열 str[]이 회문(palindrome)이면 1을 반환하고, 그렇지 않으면 0을 반환한다. ㉠과 ㉡에 들어갈 내용을 옳게 짝지은 것은? (단, 회문(palindrome)이란 문자열을 거꾸로 해도 원래의 문자열과 동일한 문자열을 말한다. 예시: 1234321, abcba, a)

<보기>

```
int is_palin(char str[], int n) {
    if ( n <= 1 ) return 1;
    if ( str[0] == str[n-1] ) return is_palin ( ㉠, ㉡ );
    else return 0;
}
```

- ① ㉠ str ㉡ n-1 ② ㉠ str ㉡ n-2
 ③ ㉠ str+1 ㉡ n-1 ④ ㉠ str+1 ㉡ n-2

4. <보기>는 중위 표기식(infix expression)으로 표현된 수식의 앞부분이다. 이 수식을 스택을 활용하여 후위 표기식(postfix expression)으로 변환할 때, 문자 F까지 처리한 후의 스택의 상태에 해당하는 것은?

<보기>

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

①
+
(
/
-

②
+
/
*
-
(

③
+
(
/
*
-
(

④
+
(
/
-
(

5. 시간 복잡도 함수 $f(n)$ 의 점근 표기법에 대한 설명으로 가장 옳지 않은 것은?

- ① $f(n) = \Theta(n^2)$ 이면 $f(n) = O(n^2)$ 이다.
 ② $f(n) = O(n^2)$ 이면 $f(n) = \Theta(n^2)$ 이다.
 ③ $f(n) = O(n^2)$ 이면 $f(n) = O(n^3)$ 이다.
 ④ $f(n) = \Omega(n^2)$ 이면 $f(n) = \Omega(n)$ 이다.

6. <보기 1>은 오름차순으로 숫자를 정렬하기 위한 알고리즘을 C언어로 구현한 함수이다. <보기 2>의 길이가 7인 정수 배열을 sort() 함수에 입력하여 오름차순 정렬 연산을 수행하고자 한다. i=4일 때의 정렬을 완료한 직후, 배열의 중간 정렬 결과는?

<보기 1>

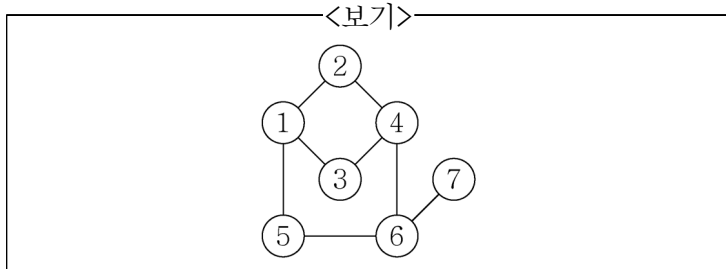
```
void sort(int list[], int n) // n은 항목의 개수임
{
    int i, j, key;
    for(i=1; i<n; i++) {
        key = list[i];
        for(j=i-1; j>=0 && list[j]>key; j--)
            list[j+1] = list[j];
        list[j+1] = key;
    }
}
```

<보기 2>

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | 3 | 8 | 4 | 9 | 1 | 6 |
|---|---|---|---|---|---|---|

- ① 1, 3, 4, 5, 8, 9, 6
 ② 3, 1, 4, 5, 6, 8, 9
 ③ 3, 4, 5, 8, 9, 1, 6
 ④ 3, 5, 8, 4, 9, 1, 6

7. <보기>의 그래프를 1번 노드에서 시작하여 너비 우선 탐색(breadth first search)을 수행하려 한다. 1번 노드를 첫 번째로 방문한다고 할 때, 다섯 번째로 방문하게 되는 노드는? (만약, 방문할 수 있는 노드가 둘 이상 있다면, 번호가 작은 쪽을 먼저 방문한다.)



- ① 4 ② 5
③ 6 ④ 7

8. 리스트 $L = (l_1, l_2, \dots, l_n)$ 에 대해서, $head(L)$ 은 L 의 첫 번째 원소 l_1 을, $tail(L)$ 은 (l_2, l_3, \dots, l_n) 을 나타낸다. 즉, $head(L)$ 은 주어진 리스트의 맨 첫 번째 원소를 의미하며 $tail(L)$ 은 그 원소를 제외한 나머지 원소로 이루어진 리스트를 의미한다. $reverse(L)$ 은 $(l_n, l_{n-1}, \dots, l_1)$ 을 나타내며, 주어진 리스트의 원소 순서를 반대로 뒤집어서 만든 리스트를 의미한다. $reverse(L)$ 과 항상 같은 결과를 나타내는 것은? (단, +는 두 원소 또는 리스트를 순서대로 이어서 만든 리스트를 나타낸다.)

- ① $l_{n-1} + reverse(tail(L))$
② $head(L) + reverse(tail(L))$
③ $reverse(tail(L)) + head(L)$
④ $reverse(tail(L)) + l_{n-1}$

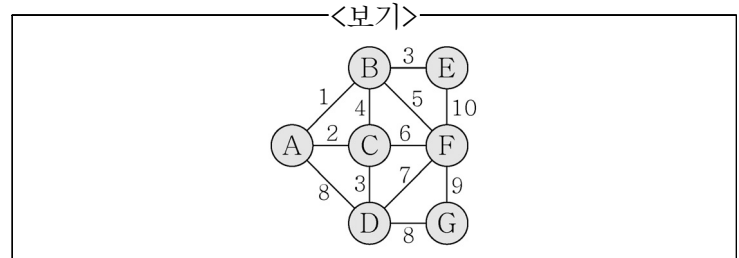
9. <보기>에 주어진 정수 데이터에 대하여 해싱(hashing) 수행 시, 오버플로우(overflow) 문제를 처리하기 위해 체이닝(chaining) 방식을 이용한다. 여기서, 해싱을 위한 해시테이블(hash table)은 5개의 버킷(0~4번)을 가지며, 해시함수는 나머지 연산자(mod 또는 %)를 포함하는 제산함수이다. 모든 데이터 항목에 대한 해싱이 완료된 이후, 해시테이블 내에서 가장 긴 리스트를 가지는 버킷 주소(번호)는?

<보기>

| | | | | | | | | | | |
|---|---|---|---|----|----|----|----|----|----|----|
| 0 | 1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 67 | 28 |
|---|---|---|---|----|----|----|----|----|----|----|

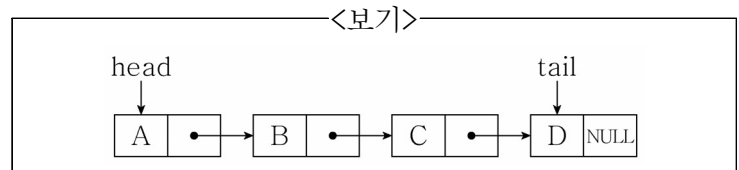
- ① 1 ② 2
③ 3 ④ 4

10. 다익스트라(Dijkstra)의 최단 경로 알고리즘을 이용하여 <보기>의 그래프의 정점 A에서 각 정점으로의 최단 경로를 구할 때, 최단 경로 발견 순서가 정점 E 바로 다음인 정점은? (단, 각 간선 위의 숫자는 양 정점들 사이의 거리를 의미한다.)



- ① B ② C
③ D ④ F

11. <보기>와 같은 형태의 단순 연결 리스트(singly linked list)에서 add_last 연산과 delete_last 연산의 수행 시간에 대한 설명으로 가장 옳은 것은? (단, add_last는 tail 포인터가 가리키는 노드 다음에 새로운 노드를 생성하여 연결하는 연산이고, delete_last는 tail 포인터가 가리키는 노드를 삭제하는 연산이다.)



- ① 두 연산 모두 $O(1)$ 시간에 수행된다.
② add_last는 $O(1)$ 시간에 수행되나, delete_last는 $O(1)$ 시간에 수행될 수 없다.
③ add_last는 $O(1)$ 시간에 수행될 수 없으나, delete_last는 $O(1)$ 시간에 수행된다.
④ add_last도 $O(1)$ 시간에 수행될 수 없고, delete_last도 $O(1)$ 시간에 수행될 수 없다.

12. 힙(heap)과 힙 정렬(heap sort)에 대한 <보기>의 설명 중 옳은 것을 모두 고른 것은?

- <보기>
- ㄱ. 최소 힙(min heap)에서 최댓값은 $O(1)$ 시간에 찾을 수 있다.
ㄴ. 최소 힙에서 3번째로 작은 값은 루트의 왼쪽 자식 또는 루트의 오른쪽 자식에 저장된다.
ㄷ. 100개의 원소가 저장된 힙의 높이는 6이다. (단, 루트 노드 하나만 있는 트리의 높이는 0으로 정의한다.)
ㄹ. 힙 정렬의 시간 복잡도는 $O(n \log_2 n)$ 이다.

- ① ㄱ, ㄴ ② ㄷ, ㄹ
③ ㄱ, ㄴ, ㄹ ④ ㄴ, ㄷ, ㄹ

13. C언어로 작성된 함수의 시간복잡도를 빅오표기법으로 표현할 때, 가장 낮은 시간복잡도를 가지는 것은?

(가)

```
void func1(int n)
{
    int i, j, k=0;
    for (i=0; i<n; i++)
        for (j=i; j<n; j++)
            k = i * j;
}
```

(나)

```
void func2(int n)
{
    int i, j, a=0;
    for (i=1; i<=n; i=i*2)
        for (j=1; j<=n; j++)
            a = i + j;
}
```

(다)

```
void func3(int n)
{
    int i, j, k=0;
    for (i=0; i<n; ++i)
        for (j=1; j<1000; ++j)
            ++k;
}
```

(라)

```
int func4(int n)
{
    int i, k=0;
    while (n>2) {
        n--; k++;
        for (i=0; i<n; i++)
            k++;
    }
    return k;
}
```

- ① (가) ② (나)
③ (다) ④ (라)

14. <보기>와 같이 단순 연결 리스트(singly linked list)가 메모리에 저장되어 있다. 이 외의 다른 정보는 메모리에 저장되어 있지 않다고 할 때, 리스트에 저장된 원소를 처음부터 링크를 따라 순서대로 바르게 나열한 것은?

| <보기> | | |
|------|-----|-------------|
| 주소 | 데이터 | 다음 노드 주소 링크 |
| 100 | 31 | NULL |
| 104 | 41 | 116 |
| 108 | 59 | 112 |
| 112 | 26 | 104 |
| 116 | 53 | 100 |

- ① 59 - 26 - 41 - 53 - 31
② 31 - 41 - 59 - 26 - 53
③ 53 - 26 - 59 - 41 - 31
④ 31 - 53 - 41 - 26 - 59

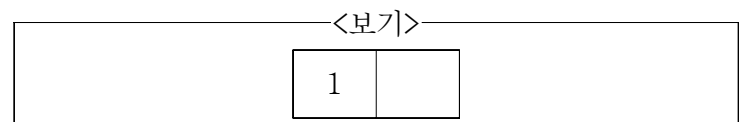
15. 스레드 이진트리(threaded binary tree)에 대한 설명으로 가장 옳지 않은 것은?

- ① 노드가 n개일 때, 순회 연산 시간은 $O(\log_2 n)$ 이다.
② 스택(stack)을 사용하지 않고 중위 순회가 가능하다.
③ 이진트리(binary tree)에 비해서 삽입/삭제 연산 시간이 증가한다.
④ 스레드인지 자식 노드에 대한 포인터인지 구별하기 위한 추가적인 필드가 필요하다.

16. C언어에서 3차원 배열 A[5][6][7]이 선언되었을 때, 메모리 상에서 원소 A[2][3][4]보다 앞에 있는 원소의 개수는? (단, 다차원 배열의 원소는 행우선 방식으로 순서대로 저장된다고 가정한다. 즉, A[0][0][0], A[0][0][1], A[0][0][2], ... 순으로 저장된다.)

- ① 9 ② 56
③ 59 ④ 109

17. <보기>의 2-3 트리에 키값 2, 3, 4, 5, 6, 7, 8을 순서대로 삽입할 경우, 각 삽입 단계별 결과에서 루트 노드가 되는 키값은?



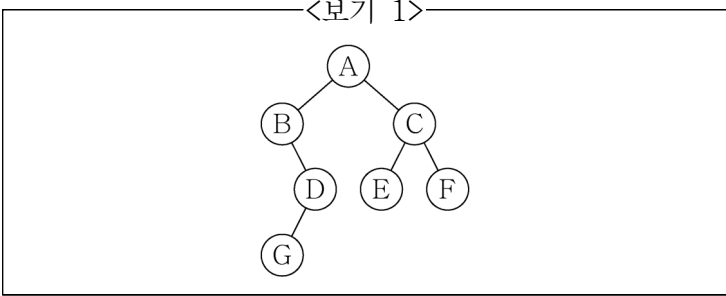
- ① (1,2) - (2) - (2) - (2,4) - (2,4) - (4) - (4)
② (1,2) - (3) - (3) - (4) - (4,6) - (4,7) - (4)
③ (2) - (3) - (2,3) - (2,3) - (3,4) - (3,4) - (4,5)
④ (1,2) - (3) - (3) - (3,4) - (4,5) - (4,7) - (4)

18. 어떤 그래프를 나타내는 인접 행렬(adjacency matrix)이 <보기>와 같이 주어졌을 때, 이로부터 알 수 있는 그래프에 대한 설명으로 가장 옳지 않은 것은?

| <보기> | |
|------|---|
| | $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$ |

- ① 이 그래프의 간선(edge)은 방향이 없거나, 모든 간선(edge)이 양방향이다.
② 노드의 개수는 총 5개이다.
③ 어떤 노드도 자기 자신을 잇는 루프(loop)가 없다.
④ 이 그래프는 연결 그래프이다.

19. <보기 1>의 트리의 루트(root)가 A노드일 때, 루트 노드 A를 인자로 하여 <보기 2>의 함수를 수행한 결과는?

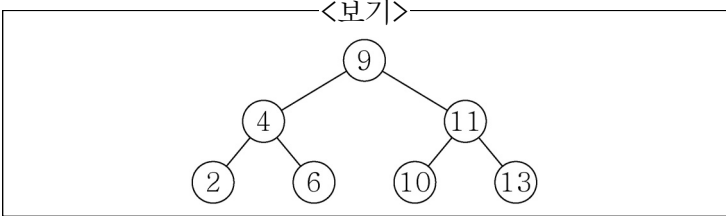


<보기 2>

```
typedef struct Node{
    char data; // 노드에 저장된 문자
    struct Node *left, *right; // 자식 포인터
} Node;
void traverse(Node *node){
    if( !node ) return;
    traverse(node->left);
    printf(" %c", node->data);
    traverse(node->right);
    printf(" %c", node->data);
}
```

- ① A B D G C E F G D B E F C A
- ② A B D G G D B C E E F F C A
- ③ B G D A E C F G D B E F C A
- ④ B G G D D B A E E C F F C A

20. <보기>의 이진 탐색 트리(binary search tree)가 만들어질 수 없는 입력순서는?



- ① 9 - 11 - 6 - 4 - 2 - 13 - 10
- ② 9 - 4 - 11 - 2 - 6 - 13 - 10
- ③ 9 - 11 - 4 - 6 - 2 - 13 - 10
- ④ 9 - 4 - 11 - 2 - 6 - 10 - 13