

2과목	자 료 구 조	(36~60)
출제위원 : 방송대 정광식		
출제범위 : 교재전체 (해당 멀티미디어강의 포함)		

36. $I = P(D)$ 의 해석으로 옳은 것은? (4점)

- ① 정보(Information)는 자료(Data)를 처리(Process)해서 얻어진 결과(Result)다.
- ② 정보(Information)는 결과(Result)를 처리(Process)해서 얻어진 자료(Data)다.
- ③ 자료(Data)는 결과(Result)를 처리(Process)해서 얻어진 정보(Information)다.
- ④ 자료(Data)는 정보(Information)를 처리(Process)해서 얻어진 결과(Result)다.

37. 자료구조의 유형 중 미리 정의된 자료구조인 것은 무엇인가? (4점)

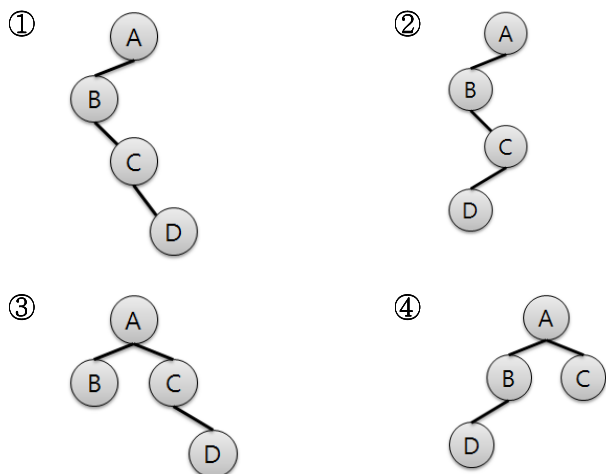
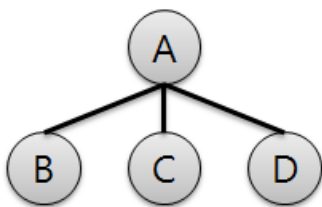
- ① 배열
- ② 스택
- ③ 큐
- ④ 트리

38. 다음 수식을 전위 표기식으로 바르게 나타낸 것은? (2점)

$A+B+C*D$

- ① ++AB*CD
- ② ABCD+++
- ③ ++*ABCD
- ④ AB+B+C*

39. 다음 일반 트리를 이진트리로 변환한 것으로 옳은 것은? (2점)



40. 알고리즘의 조건에 포함되지 않는 것은? (4점)

- ① 출력
- ② 입력
- ③ 절대성
- ④ 유한성

41. 인덱스와 원소값(<index, value>)의 쌍으로 구성된 집합이며, 정의된 각 인덱스는 그 인덱스와 관련된 값으로 정의되는 자료 구조는 무엇인가? (4점)

- ① 리스트
- ② 스택
- ③ 배열
- ④ 큐

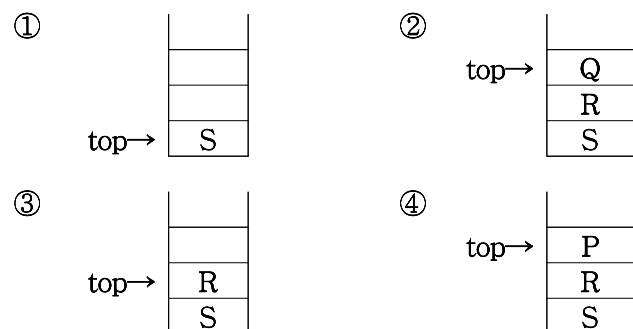
42. 다음과 같은 행렬이 행우선 방식으로 저장된다면, [3,4]의 다음에 저장되는 행렬의 원소는 무엇인가? (2점)

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
[0]	0	20	0	0	9	0	0	11	0
[1]	0	0	0	0	0	0	0	0	0
[2]	78	0	0	0	0	0	0	0	0
[3]	0	0	0	0	67	0	0	0	0
[4]	0	31	0	0	0	0	0	0	0
[5]	0	0	0	91	0	0	44	0	0
[6]	0	0	0	0	0	0	0	0	0
[7]	0	0	0	0	19	0	0	27	0

- ① [3,5]
- ② [3,3]
- ③ [4,4]
- ④ [4,3]

43. 다음은 스택에 대한 연산이다. 교재에 기술된 스택의 추상자료형을 따른다면, ⑧ 연산을 수행한 후의 스택의 모습은 무엇인가? (2점)

- ① CreateS(3);
- ② Push(stack, 'S');
- ③ Push(stack, 'T');
- ④ Pop(stack);
- ⑤ Push(stack, 'R');
- ⑥ Push(stack, 'P');
- ⑦ Pop(stack);
- ⑧ Pop(stack);

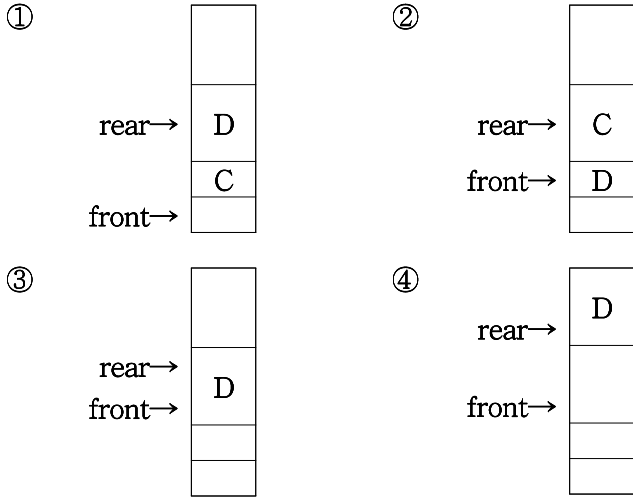


44. 큐의 운용과 유사하게 운영되는 것이 아닌 것은 무엇인가? (3점)

- ① 문서 출력을 위해 프린터기를 이용할 때, 여러 개의 문서를 출력해도 먼저 인쇄버튼을 누른 문서부터 차례로 출력된다.
- ② 은행에서 번호표를 뽑고 창구에 가기를 기다린다.
- ③ 택시 승강장에서 택시를 기다린다.
- ④ 웹브라우저에서 방금 전 방문했던 사이트 기록 저장 후 '이전 페이지로 돌아가기'를 클릭한다.

45. 다음은 큐에 대한 연산이다. 교재에 기술된 큐의 추상자료형을 따른다면, ⑧ 연산을 수행한 후의 스택의 모습은 무엇인가? (2점)

- ① Create_q(4);
- ② Add_q(queue, 'A');
- ③ Add_q(queue, 'B');
- ④ Add_q(queue, 'C');
- ⑤ Delete_q(queue);
- ⑥ Delete_q(queue);
- ⑦ Delete_q(queue);
- ⑧ Add_q(queue, 'D');



46. 단순 연결리스트의 사용되지 않는 마지막 노드의 링크 부분을 활용하기 위해 제안된 리스트는 무엇인가? (4점)

- ① 이중 연결 리스트
- ② 원형 연결 리스트
- ③ 회전 연결 리스트
- ④ 중복 연결 리스트

47. 사용하지 않는 포인터를 이용하여 이진 트리 순회를 편리하고 빠르게 향상시킨 트리는 무엇인가? (3점)

- ① 탐색 이진트리
- ② 스레드 이진트리
- ③ 포화 이진트리
- ④ 완전 이진트리

48. 각 노드가 두 개의 자식노드 보다 더 큰 값(패자)을 가지며 최종 승자는 0번 노드에 저장하는 트리는 무엇인가? (3점)

- ① 선택트리
- ② 패자트리
- ③ 승자트리
- ④ AVL 트리

49. 트리에 속한 잎 노드의 개수를 무엇이라고 하는가? (3점)

- ① 트리의 깊이
- ② 트리의 높이
- ③ 트리의 무게
- ④ 트리의 경로

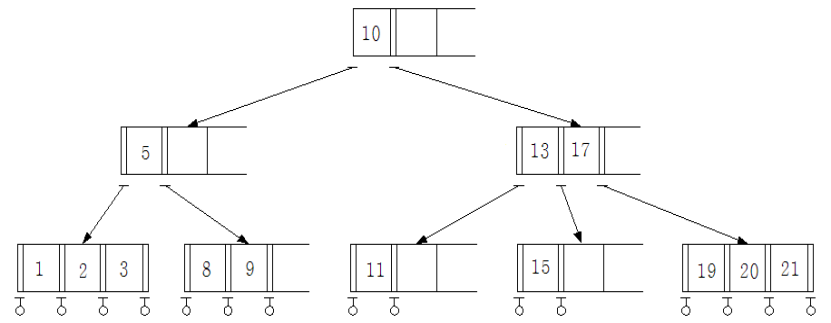
50. 노드 v_i 의 왼쪽 서브트리 높이와 v_i 의 오른쪽 서브트리 높이가 최대 1만큼 차이가 난다는 조건을 만족하는 트리는 무엇인가? (3점)

- ① Splay 트리
- ② BS 트리
- ③ BB 트리
- ④ AVL 트리

51. 그래프에 대한 설명으로 옳은 것은 무엇인가? (3점)

- ① 사이클이 없는 그래프를 트리라고 한다.
- ② 두 정점이 사이클로 연결되었을 때, 두 정점이 인접한다고 정의한다.
- ③ 시작점과 끝점이 같은 경로를 루프라고 한다.
- ④ 한 정점에서 출발하여 자신으로 연결하는 간선을 사이클이라고 한다.

52. 다음과 같은 트리는 무엇인가? (3점)



- ① 이진 트리
- ② 1-2 트리
- ③ 2-3 트리
- ④ 2-3-4 트리

53. 다음 중 B 트리에 대한 설명으로 옳은 것은 무엇인가? (3점)

- ① 트리의 루트는 최소한 3개의 서브트리를 갖는다.
- ② 트리의 모든 잎 노드는 같은 레벨에 있다.
- ③ 루트가 가장 작은 값을 갖고 부모는 자식보다 작은 값을 갖는다.
- ④ 루트와 잎 노드를 제외한 트리의 각 노드는 최소 $\lceil m/3 \rceil$ 개의 서브트리를 갖는다.

54. 남은 간선 중에서 무조건 최소 비용인 간선을 선택한 후 사이클을 형성하지 않으면 그 간선을 선택하는 최소 비용 신장 트리 결정 방법은 무엇인가? (3점)

- ① Prim 방법
- ② DFS 방법
- ③ Kruscal 방법
- ④ BFS 방법

55. 두 정점쌍이 간선을 여러 개 가질 수 있는 그래프를 무엇이라고 하는가? (3점)

- ① 중복 그래프
- ② 무방향 그래프
- ③ 다중 그래프
- ④ 단순 그래프

56. 다음 프로그램은 교재에서 정의된 이중 연결 리스트에 새 노드를 삽입하는 연산이다. 프로그램의 흐름상 다음 빈칸 ㉠에 들어갈 알맞은 코드는 무엇인가? (2점)

```
void addDNode(linkedList_h* H, listNode* prevNode, int x){
//NewNode를 prevnode의 오른쪽에 삽입
    listNode* NewNode;
    NewNode = (listNode*)malloc(sizeof(listNode));
    NewNode->Llink = NULL;
    NewNode->data = x;
    NewNode->Rlink = NULL;

    NewNode->Rlink = prevNode->Rlink;
    prevNode->Rlink = NewNode;
    NewNode->Llink = prevNode;
    [      ㉠      ]
}
```

- ① NewNode → Rlink → Llink = NewNode;
- ② prevNode → Llink → Rlink = NewNode;
- ③ prevNode → Rlink → Llink = NewNode;
- ④ NewNode → Rlink → Llink = prevNode;

57. 다음 표에서 (가), (나)의 순서대로 가장 적합한 내용은 무엇인가? (2점)

	자료	연산
추상화	추상 자료형	(가)
구체화	자료형	(나)

- ① 프로그램, 알고리즘
- ② 자료구조, 알고리즘
- ③ 슈도 코드, 프로그램
- ④ 알고리즘, 프로그램

58. 다음은 깊이 우선 탐색(DFS)알고리즘의 의사코드이며 해당 코드는 순환 호출을 이용하는 경우이다. 다음 알고리즘에서 ㉠과 ㉡에 들어갈 알맞은 명령어는 무엇인가? (2점)

```

① void DFS(int v) {
②     int w;
③     extern int VISITED[];
④     [ ㉠ ]
⑤     while(v에 인접한 모든 노드 w)
⑥         [ ㉡ ]
⑦         DFS(w);
⑧ }
```

- ① ㉠ : VISITED[v] = 0;
㉡ : if(!q_empty())
- ② ㉠ : VISITED[v] = 0;
㉡ : if(VISITED[w])
- ③ ㉠ : VISITED[v] = 1;
㉡ : if(!VISITED[w])
- ④ ㉠ : VISITED[v] = 1;
㉡ : if(!VISITED[v])

59. 다음은 배열을 이용해 스택을 구현하고 스택에 데이터를 삽입하는 과정을 나타내는 코드이다. [가]에 들어갈 코드로 가장 알맞은 것은 무엇인가? (2점)

```

<스택의 생성>
#define STACK_SIZE 10
typedef int element;
element stack[STACK_SIZE];
int top = -1;
```

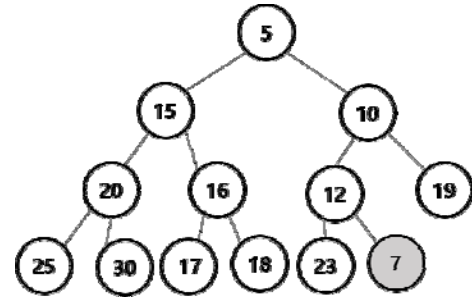
```

<삽입 과정>
void push(element item){
    if( [가] ){
        printf("Stack is Full! \n");
        return;
    }
    else
        stack[++top] = item;
}
```

- ① top == -1
- ② top >= STACK_SIZE - 1
- ③ top >= STACK_SIZE
- ④ top <= STACK_SIZE

60. 다음 프로그램은 최소 힙에서 데이터를 삽입하는 프로그램이다. 아래 그림은 최소 힙에 새 노드를 삽입하기 위해 마지막 노드에 새 노드인 '7'이 위치한 모습이다. 새 노드인 '7'이 삽입한 후 완전한 최소 힙이 되도록 하려면 [가]부분이 몇 번 이루어져야 하는가? (2점)

<최소힙그림>



```

int insertHeap(HeapType *h, int item) {
    int i;
    i = ++(h->heap_size);

    while((i != 1)&&(item < h->heap[i/2])){
        [가]
        h->heap[i] = h->heap[i/2];
        i /= 2;
    }
    h->heap[i] = item;
}
```

- ① 0
- ② 1
- ③ 2
- ④ 3