

2010학년도 중등교사신규임용후보자선정경쟁시험

정보 · 컴퓨터

1차 시험	2 교시 (전공)	40 문항 80 점	시험 시간 120 분
-------	-----------	------------	-------------

- 문제지 전체 면수가 맞는지 확인하십시오.
- 문항의 배점이 1.5점과 2.5점인 문항에는 배점이 표시되어 있습니다. 나머지 문항은 2점입니다.
- 각 문항의 정답을 컴퓨터용 흑색 사인펜을 사용하여 OMR 답안지에 표시하십시오.

1. 교과교육은 교육과정에 기초하여 운영되고 있으며, 학교교육의 본질을 대변하는 중요한 영역이다. 컴퓨터 교과교육에 관련된 설명으로 옳은 것을 <보기>에서 고른 것은?

<보기>

- ㄱ. 컴퓨터 교과교육의 발전을 위해서는 컴퓨터 교과교육의 본질에 대한 명확한 인식이 필요하다.
- ㄴ. 컴퓨터 교과교육학은 컴퓨터 교과 내용과 컴퓨터 교육 방법의 단순한 물리적 결합을 의미한다.
- ㄷ. 컴퓨터 교과교육학은 교육 내용으로서의 지식 그리고 가르치는 방법으로서의 교육학으로 분리하는 이론 중심 교육학의 한 분야이다.
- ㄹ. 컴퓨터 교과교육을 발전시키는 과제 중 하나는 독자적인 학문 체계로서 컴퓨터 교과교육학의 정체성을 확립하는 것이다.

- ① ㄱ, ㄴ ② ㄱ, ㄷ ③ ㄴ, ㄷ
- ④ ㄴ, ㄹ ⑤ ㄷ, ㄹ

2. 효과적인 수업의 열쇠는 교사의 수업 전문성에 있으며, 수업 전문성의 핵심 중 하나는 교과교육학 지식(PCK : Pedagogical Content Knowledge)이다. 컴퓨터 교과교육과 PCK에 관련된 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

<보기>

- ㄱ. 교과내용 지식은 PCK의 발달에 영향을 미치지 않는다.
- ㄴ. PCK는 특정 교과내용과 수업방법을 학습자의 특성에 알맞게 조직하고, 적용하는 지식이다.
- ㄷ. 술만(Shulman)의 주장에 따르면, 교사의 지식에 대한 분류는 교과내용 지식(Content Knowledge), 일반교육학 지식(General Pedagogical Knowledge), 교과교육학 지식 등을 포함한다.
- ㄹ. 디코더의 동작 원리에 대해 오개념(誤概念)이 형성된 학습자가 개념을 올바르게 형성할 수 있도록 교사가 새로운 도식을 제작하여 수업에 활용하는 것은 PCK의 한 예이다.

- ① ㄱ, ㄴ ② ㄴ, ㄷ ③ ㄱ, ㄴ, ㄷ
- ④ ㄴ, ㄷ, ㄹ ⑤ ㄱ, ㄴ, ㄷ, ㄹ

3. <보기>에 제시된 교사의 수업 상황 설명 중 '구성주의 학습 이론'에 해당되는 것을 모두 고른 것은?

<보기>

- ㄱ. 김 교사 : 문제의 이해와 표현에 대한 수업을 설계할 때 봄소풍으로 갈 놀이 공원에서 주어진 시간 안에 자신이 좋아하는 놀이 기구를 많이 탈 수 있는 방법을 각자 생각해 보고 문제를 해결해 나가는 수업으로 설계한다.
- ㄴ. 박 교사 : 컴퓨터의 역사에서 중요한 사건을 연표로 만드는 수업을 할 때 중요한 사건들을 외울 수 있도록 반복하여 말하는 연습을 하게 한다.
- ㄷ. 송 교사 : C 언어 프로그래밍 수업을 할 때 교사가 먼저 시범을 보이고 학생에게 프로그래밍을 하게 한다. 교사는 순회하면서 도움을 주다가 점차 도움을 줄여 학습자 스스로 프로그래밍을 할 수 있도록 한다.
- ㄹ. 이 교사 : 이진수와 십진수 변환 활동지를 학생들이 풀고 있을 때, 교사가 순회 지도를 한다. 학생이 풀 문제가 틀렸으면 오답이라고 즉시 말해주고 다시 풀도록 한다.
- ㅁ. 장 교사 : 우리 반의 홈페이지를 제작하려고 한다. 설계와 디자인을 맡은 모둠, 웹 프로그래밍으로 개발을 맡은 모둠으로 나누어 활동한 후 함께 모여 개발 과정을 토의하고 상호 점검하는 과정이 포함되는 수업을 설계한다.

- ① ㄱ, ㄷ, ㅁ ② ㄱ, ㄷ, ㅁ ③ ㄴ, ㄷ, ㄹ
- ④ ㄱ, ㄴ, ㄷ, ㅁ ⑤ ㄱ, ㄴ, ㄷ, ㄹ, ㅁ

4. 제7차 컴퓨터 교육과정의 편성 및 운영에 관한 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

<보기>

- ㄱ. 중학교 '컴퓨터' 과목은 5~6학년의 실과 교육을 바탕으로 남녀의 구분 없이 7~9학년에서 선택하여 이수할 수 있는 과목이다.
- ㄴ. 중학교의 교과 재량 활동의 연간 수업 시간 수는 102시간 이상이며, 한문, 컴퓨터, 환경, 생활 외국어, 기타의 선택 과목 학습 시간에 우선 배정한다.
- ㄷ. 고등학교 1학년에서는 선택 중심 교육과정에 편성된 '정보 사회와 컴퓨터' 과목을 교과 재량 활동 시간에 이수할 수 있다.
- ㄹ. 고등학교 선택 중심 교육과정의 전문 교과에는 정보, 농업, 공업, 상업에 관한 교과가 포함된다.

- ① ㄱ, ㄴ ② ㄴ, ㄷ ③ ㄷ, ㄹ
- ④ ㄱ, ㄴ, ㄷ ⑤ ㄱ, ㄴ, ㄷ, ㄹ

5. 컴퓨터 교과 내용의 선정과 조직에 관련된 설명으로 옳은 것을 <보기>에서 고른 것은?

<보 기>

ㄱ. 교사들이 컴퓨터 교과 내용을 체계적으로 선정하고 조직함으로써, 컴퓨터 교육 과정에 대한 교사의 전문성을 향상시킬 수 있다.

ㄴ. 교육 내용 선정 원리 중 타당성의 원리는 사실, 개념, 원리, 이론들을 가리키는 학문의 구조와 합리적 탐구 방법을 포함해야 한다는 원리이다.

ㄷ. 교육 내용의 조직 원리는 수평적 조직 원리와 수직적 조직 원리로 나뉘는데, 수평적 조직 원리에는 연속성과 계열성이 있고, 수직적 조직 원리에는 범위와 통합성이 있다.

ㄹ. 학생들이 장차 살아갈 사회에서 필요로 하는 지식, 기능, 가치를 컴퓨터 교과의 교육 내용으로 선정한다면, 이것은 교육 내용 선정 원리 중 '사회적 유용성의 원리'가 적용된 것이다.

- ① ㄱ, ㄴ ② ㄱ, ㄷ ③ ㄴ, ㄷ
 ④ ㄴ, ㄹ ⑤ ㄷ, ㄹ

6. <보기>의 컴퓨터 교수·학습 상황에 적용할 수 있는 수업 방법으로 가장 적절하게 짝지어진 것은?

<보 기>

㉠ 다양한 네트워크 서비스를 이해하는 수업으로 FTP, 전자우편, 블로그, 메신저, P2P 서비스를 주제별로 모듈 학습을 한 후 자신의 모듈로 돌아가 발표한다.

㉡ 컴퓨터에 사용되는 다양한 문자 코드 방식을 설명할 때 지난 시간에 학습했던 컴퓨터의 문자 표현 방식과 문자 코드를 연결 지을 수 있는 자료를 제시한다.

㉢ 개인 정보 보호에 대한 태도를 기르기 위한 수업으로 개인 정보 침해 사례를 각 모듈에 나누어 준다. 자료를 읽어 보고 자신의 생각, 자신의 침해 사례 및 해결 방법을 모듈원들과 이야기한다.

㉣ 교육용 프로그래밍 언어로 조건문과 반복문을 활용한 프로그램을 작성하는 수업을 할 때 조건문과 반복문을 나누어 실습한 후 종합하여 실습한다.

- | | | | |
|------------|---------------------|----------|------|
| ㉠ | ㉡ | ㉢ | ㉣ |
| ① 유의미 수용학습 | 과제분담학습 I (Jigsaw I) | 집단토론 | 분습법 |
| ② 과제분담학습 I | 유의미 수용학습 | 집단토론 | 분습법 |
| ③ 유의미 수용학습 | 분습법 | 과제분담학습 I | 집단토론 |
| ④ 분습법 | 과제분담학습 I | 유의미 수용학습 | 집단토론 |
| ⑤ 과제분담학습 I | 집단토론 | 유의미 수용학습 | 분습법 |

7. 모리슨(G. Morrison)은 학습 내용 제시 방법을 학습 활동 관련 내용 제시(Learning-related sequencing), 영역 관련 내용 제시(World-related sequencing), 개념 학습 관련 내용 제시(Concept-related sequencing)로 분류하였다. <보기> 활동지를 보고 ㉠~㉣에 해당하는 내용 제시 방법으로 바르게 짝지은 것은?

<보 기>

※ 학습 주제 : 컴퓨터 구성 요소에 대해 알아보자.

㉠ 다음 밑줄 친 곳에 알맞은 말을 쓰시오.

- 컴퓨터 구성 요소는 크게 ___ 와/과 ___ 로/으로 구분되고, 소프트웨어는 크게 ___ 와/과 ___ 로/으로 구분된다.
- 응용 소프트웨어 중 일반 업무용 소프트웨어에는 _____ 등이 있다.

㉡ 컴퓨터 하드웨어의 역사인데, 밑줄 친 곳에 알맞은 말을 쓰시오.

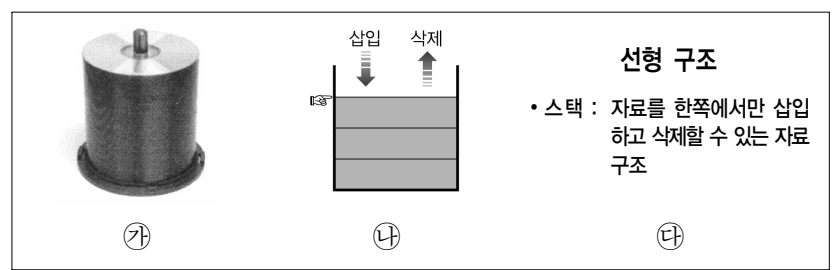
- 1833년 : 배비지(C. Babbage)의 해석 기관
- 1949년 : _____ (최초의 프로그램 내장 방식인 컴퓨터)
- 1964년 : IBM 360 (IC를 최초로 사용한 컴퓨터)

㉢ 자신이 사용하는 소프트웨어의 종류를 적어 보고, 제시된 전문가용 소프트웨어의 하는 일을 알아본 후 아래 표에 맞게 쓰시오.

하는 일	자신이 사용하는 소프트웨어	전문가가 사용하는 소프트웨어
보고서를 작성한다.		
그림을 편집한다.		
주소록을 관리한다.		
(이하 생략)		

- | | | | |
|---|-------|-------|-------|
| | ㉠ | ㉡ | ㉢ |
| ① | 학습 활동 | 영역 | 개념 학습 |
| ② | 영역 | 학습 활동 | 개념 학습 |
| ③ | 개념 학습 | 학습 활동 | 영역 |
| ④ | 개념 학습 | 영역 | 학습 활동 |
| ⑤ | 학습 활동 | 개념 학습 | 영역 |

8. 다음은 선형 구조의 하나인 스택(stack) 학습을 위해 제작한 수업 자료이다. 이와 같은 시각 자료 제작과 관련된 일반적인 원리와 활용 방법에 대한 설명으로 적절하지 않은 것은?



- ① 하나의 시각 자료는 한 개의 개념만을 표현하는 것이 바람직하다.
- ② 정적인 시각 자료에 화살표와 같이 방향성을 나타내는 아이콘을 삽입하면 자료에 방향성을 줄 수 있다.
- ③ ㉠과 같이 추상성이 높을수록 개념과 성질 등을 더 경제적으로 표현할 수 있다.
- ④ 선형 구조에 대한 선행 지식이 있는 중학생이라면, ㉡보다 ㉠과 같이 사실에 근접한 시각 자료를 사용하는 것이 교수·학습에 더 효과적이다.
- ⑤ 시각 자료에 내용을 제시할 때에는 삼등분 원칙, 즉 화면의 가로와 세로를 각각 삼등분하여 삼등분 선이 만나는 곳에 중요한 내용을 배치하는 것이 바람직하다.

9. 다음은 협동 학습 방법 가운데 STAD(Student Teams Achievement Division) 모형을 적용한 수업으로 네 단계를 간략히 나타낸 것이다. STAD에 적절한 교수·학습 활동을 모두 고른 것은?

단계	교수·학습 활동
교사의 수업 안내	<ul style="list-style-type: none"> ○ 'C 프로그래밍 언어에서 포인터(pointer)의 개념을 설명할 수 있다'는 학습 목표를 제시한다. ○ 포인터에 대한 개념을 설명하고, 구조도를 제시한다. ○ 소집단 활동에 대한 안내를 제공한다.
소집단 학습	<ul style="list-style-type: none"> ㉠ 포인터 학습을 위한 소집단의 구성원은 4~6명으로 구성하도록 한다. ㉡ 소집단의 구성원은 성별, 성적, 성격 등을 고려하여 최대한 비슷한 학습자기리 구성하도록 한다. ○ 소집단의 구성원은 각자의 역할을 분담한다.
평가	<ul style="list-style-type: none"> ㉢ 개인의 향상 점수를 정하는 기준은 선행 학습이었던 C 프로그래밍 언어의 '배열'에 대한 학습 결과인 자신의 퀴즈 점수의 평균으로 한다. ○ 포인터의 개념에 대한 내용으로 개인별 퀴즈를 실시한다. ㉣ 향상 점수의 기준은 한 번 정했으면 변경할 수 없다. ㉤ 소집단 점수는 구성원들의 향상 점수의 평균으로 산정하도록 한다.
소집단별 보상	<ul style="list-style-type: none"> ○ 최대한 빨리 점수를 발표한다. ○ 가능한 한 많은 소집단에게 시상하도록 한다.

- ① ㉠, ㉡ ② ㉢, ㉤ ③ ㉠, ㉡, ㉣
 ④ ㉠, ㉡, ㉣ ⑤ ㉠, ㉡, ㉣, ㉤

10. '인간 중심 교육과정에서는 교육 활동을 통해 자아실현이나 인간 완성을 추구해야 한다'는 교육적 신념을 가진 이 교사는 참여나 협동과 같은 기초적인 학습 태도가 부족한 학생들을 위하여 학습목표를 확대하여 알고리즘에 대한 수업을 재구성하였다. 다음은 '순서도'에 관한 수업의 진행 과정을 단계에 따라 상세히 기술한 교수·학습 지도안의 일부이다.

교 과	컴퓨터	단원명	순서도	대상	지도교사	차시	교과서
				7학년	신정보	4~5/10	0 쪽
교 수 학 습 방 법 및 모 형	협동학습 -STAD	학 습 목 표	1) 조건문의 의미와 표현 방법을 설명할 수 있다. 2) 여러 가지 도구를 활용하여 조건문을 능숙하게 도식화할 수 있다. 3) 가족에게 감사하는 마음을 느낄 수 있다. 4) 소집단 활동에 적극적으로 참여할 수 있다.				

다음은 이 수업에서 학습자가 작성한 내용(활동지의 필기체)이 포함된 활동지이다.

활 동 지

(1) 가족과 자신이 가장 행복했던 순간은 언제인가요? 행복 곡선을 그래프로 그려봅시다.

(2) 만약 그 행복한 순간이 없었다면, 지금은 어떻게 지내고 있을까요?

소중한 추억을 만들 수 있는 가족 없이 외롭게 지내고 있을 것이다.

(3) 자신만의 표현 방식으로 문항 (2)번의 결과를 그림으로 나타내어 볼까요?

(4) 자신의 표현 방법과 일반적인 표현 방법을 비교해 볼까요?

의미	나의 표현	일반적 표현	용어
비교	∧	◇	조건문
이동	→	↓, ←	분기

(5) 여러 도구를 활용하여 문항 (3)번의 그림을 순서도로 변환시켜 볼까요?

(6) 행복한 추억이 있다는 것에 대해 무엇을 느꼈나요?

가족은 무엇과도 바꿀 수 없는 커다란 선물입니다^^

(7) 감사하는 마음을 누구에게 어떻게 전할지 의논해 볼까요?

사랑하는 아빠, 엄마, 동생!
먼저 부모님 받을 씻겨 드려야겠어요.
그리고 일기도 쓰고 싶어요.

(8) 문항 (7)에 답한 내용을 순서도로 나타내어 볼까요?

이 수업에 관련된 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

—<보 기>—

ㄱ. 블룸(B. Bloom)의 교육목표 분류학에 따르면, 학습목표 '1) 조건문의 의미와 표현 방법을 설명할 수 있다'는 인지적 영역의 '이해(Comprehension)' 능력을 기르는 것에 해당하며, 활동지의 문항 (4)에 관련이 있다.

ㄴ. 크래스웰(D. Krathwohl)에 따르면, '4) 소집단 활동에 적극적으로 참여할 수 있다'는 정의적 영역의 학습목표는 '반응(Responding)'의 범주에 속하고, 문항 (7)에 관련이 있다.

ㄷ. 심슨(E. Simpson)에 따르면, '2) 여러 가지 도구를 활용하여 조건문을 능숙하게 도식화할 수 있다'는 심체적 영역의 학습 목표는 '창조(Origination)'의 범주에 속하고, 문항 (5)에 관련이 있다.

ㄹ. 전인 교육의 관점에서는 인지적·정의적·심체적 학습은 따로따로 이루어지는 것이 아니라 유기적 관련 하에서 상호 작용한다고 믿는다.

- ① ㄱ, ㄷ ② ㄴ, ㄹ ③ ㄱ, ㄴ, ㄹ
 ④ ㄴ, ㄷ, ㄹ ⑤ ㄱ, ㄴ, ㄷ, ㄹ

11. 다음은 가네(R. Gagné)의 9가지 교수 사태(instructional events) 이론을 적용한 컴퓨터 수업 활동 계획서이다. 교수 사태의 순서에 따라 수업 단계 ㉠~㉨를 적절하게 배열한 것은? [2.5점]

○ 학습 목표 : 숫자 카드 활동을 통해 선택 정렬의 원리를 이해할 수 있다.

단계	수업 활동 계획	교수·학습 자료
㉠	숫자 카드로 선택 정렬 과정을 보여 주면서 학습을 안내한다.	숫자 카드 프레젠테이션
㉡	학생에게 정렬되지 않은 여덟 장의 숫자 카드를 나누어 주면서 학습 내용을 제시한다.	숫자 카드
㉢	학급 도서들이 책장에 정리되지 않은 모습의 사진을 보여 주고, 원하는 책을 빨리 찾지 못한 학생들의 경험담을 이야기하게 하여 수업의 흥미를 유발한다.	사진
㉣	학생들이 스스로 숫자 카드를 이용하여 오름차순과 내림차순으로 선택 정렬을 하도록 한다.	숫자 카드
㉤	학습 목표를 제시한다.	프레젠테이션
㉥	지난 시간에 학습했던 내용을 질문한다.	프레젠테이션
피드백 제공	교실을 순회하면서 학생들이 숫자 카드로 선택 정렬하는 과정을 살펴보고 지도한다.	숫자 카드
수행 평가	순서가 흩어진 열 장의 카드를 보여 주면서 오름차순으로 정렬하는 문제를 풀게 한다.	프레젠테이션 숫자 카드
과지와 전이	학급도서 열 권을 책이름을 기준으로 오름차순 선택 정렬하게 한다.	학급의 도서 책장

- ① ㉤→㉥→㉦→㉡→㉠→㉣
- ② ㉤→㉥→㉦→㉡→㉣→㉠
- ③ ㉤→㉥→㉦→㉠→㉣→㉡
- ④ ㉥→㉤→㉦→㉡→㉠→㉣
- ⑤ ㉥→㉦→㉤→㉡→㉣→㉠

12. 저작권법과 저작권 보호 기술에 관한 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

<보 기>

<p>㉠. 저작권의 보호를 받는 저작물에는 어문 저작물, 음악 저작물, 연극 저작물, 미술 저작물, 사진 저작물, 컴퓨터 프로그램 저작물 등이 있다.</p> <p>㉡. 교사가 중학교에서 교육을 목적으로 교수·학습의 자료로 다른 사람의 저작물을 복사하여 사용하는 경우에는 저작 재산권이 침해되므로 사용할 수 없다.</p> <p>㉢. 사실의 전달에 불과한 시사 보도도 저작물의 요건을 충족하기 때문에 저작권법의 보호를 받는다.</p> <p>㉣. 그림, 음악, 동영상과 같은 디지털 콘텐츠에 육안으로 구별할 수 없는 디지털 워터마크(watermark)를 삽입함으로써 불법 사용을 방지할 수 있다.</p>
--

- ① ㉠, ㉢ ② ㉠, ㉢ ③ ㉠, ㉡, ㉢
- ④ ㉠, ㉢, ㉣ ⑤ ㉠, ㉢, ㉣

13. 컴퓨터 교육의 평가에 대한 교사들의 대화이다. <보기>에서 옳은 것을 모두 고른 것은? [1.5점]

<보 기>

<p>㉠. 권 교사 : HTML을 이해하고 있는지 평가하기 위해 선택형 문항으로 문제지를 만들었다. 학생 90명을 대상으로 동일한 시험지를 3주 간격을 두고 반복 평가한 결과 상관관계가 낮았다. 이 경우 평가 도구의 신뢰도가 낮은 것이 한 요인일 수 있다.</p> <p>㉡. 안 교사 : 알고리즘의 표현 방법 중 하나인 순서도를 학습하였다. 순서도를 이해하고 있는지 평가하기 위해 C 언어의 문법에 대한 형성 평가를 실시하였다. 이 경우 평가 도구의 타당도가 높다고 볼 수 있다.</p> <p>㉢. 한 교사 : 동영상 제작 프로그램을 사용하여 동영상을 제작할 수 있는지 평가하기 위해 실습 평가를 실시했는데, 평가자 간 채점 결과의 편차가 크고 다양하게 나타났다. 이 경우 평가의 객관도가 낮다고 볼 수 있다.</p>

- ① ㉠ ② ㉠, ㉡ ③ ㉠, ㉢
- ④ ㉠, ㉢ ⑤ ㉠, ㉡, ㉢

14. 다중 쓰레드 프로세스(multithreaded process)에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은?

<보 기>

<p>㉠. 사용자 쓰레드는 커널의 직접 지원 없이 사용자 공간에서 관리되며, 커널 쓰레드는 운영체제에 의해 직접 지원되고 관리된다.</p> <p>㉡. 같은 프로세스 내의 쓰레드들은 공통의 주소 공간을 통해 정보를 교환할 수 있고, 각 쓰레드의 자원인 스택, 힙 메모리를 서로 공유한다.</p> <p>㉢. 사용자 쓰레드와 커널 쓰레드의 일대일(one-to-one) 모델은 다대일(many-to-one) 모델보다 더 많은 병렬성을 제공한다.</p> <p>㉣. 사용자 쓰레드와 커널 쓰레드의 다대일(many-to-one) 모델은 한 쓰레드가 봉쇄형 시스템 호출을 해도 전체 프로세스가 봉쇄되지 않으며, 다중 쓰레드가 다중 처리기에 의해 수행될 때 병렬로 작동된다.</p>
--

- ① ㉠, ㉢ ② ㉠, ㉢ ③ ㉢, ㉣
- ④ ㉠, ㉡, ㉣ ⑤ ㉠, ㉢, ㉣

15. 하나의 데이터베이스를 다수의 병행 프로세스들이 공유할 때, 운영체제의 프로세스 동기화 문제가 발생하는데 이를 '읽기-쓰기 문제(readers-writers problem)'라 한다. 다음은 세마포 제어 함수(`wait()`, `signal()`), 세마포(`mutex`, `wrt`), 변수(`readcount`)를 사용한 C 언어 유형의 읽기 프로세스 알고리즘이다. 이 알고리즘의 ㉠, ㉡, ㉢, ㉣, ㉤에 들어갈 세마포 제어 함수로 옳은 것은?

```
typedef struct {
    int value;
    struct process *list;
} Semaphore;

Semaphore mutex = {1}; /* readcount 변경 시 상호배제용으로 사용한다. */
Semaphore wrt = {1}; /* 임계구역으로 진입하는 첫 번째 읽기 프로세스와 임계구역을 빠져나오는 마지막 읽기 프로세스에 의해 사용된다. */
int readcount = 0; /* 읽기에 참여하고 있는 프로세스 개수 */

void wait(Semaphore *s) {
    s->value--;
    if (s->value < 0) {
        이 프로세스를 대기 큐(waiting queue)에 넣는다;
        자기를 호출한 프로세스를 중지시킨다;
    }
}

void signal(Semaphore *s) {
    s->value++;
    if (s->value <= 0) {
        대기 큐(waiting queue)로부터 하나의 프로세스 p를 꺼낸다;
        봉쇄된 프로세스 p의 실행을 재개시킨다;
    }
}

void readers() {
    while (1) {
        ... /* 다른 계산이 실행된다. */
        wait(&mutex);
        readcount++;
        if (readcount == 1) ㉠ ;
        ㉡ ;
        ... /* 읽기 연산이 실행된다. */
        ㉢ ;
        readcount--;
        if (readcount == 0) ㉣ ;
        ㉤ ;
    }
}
```

	㉠	㉡	㉢	㉣	㉤
①	wait(&wrt)	signal(&mutex)	wait(&mutex)	signal(&wrt)	signal(&mutex)
②	wait(&wrt)	wait(&mutex)	signal(&mutex)	signal(&wrt)	signal(&mutex)
③	wait(&wrt)	signal(&mutex)	wait(&mutex)	signal(&mutex)	signal(&wrt)
④	signal(&mutex)	wait(&wrt)	signal(&wrt)	wait(&mutex)	wait(&wrt)
⑤	signal(&mutex)	signal(&wrt)	wait(&mutex)	signal(&mutex)	wait(&wrt)

16. 요구 페이징(demand paging)에서 논리 주소는 메인 메모리 공간에 있는 페이지 테이블에 의해 번역되는데, 이러한 페이지 테이블을 거쳐 메인 메모리에 있는 자료를 접근하려면 결국 두 번의 메인 메모리 접근이 필요하게 된다. 이 시간을 개선하기 위하여 TLB(Translation Look-aside Buffer)라는 메모리를 추가하여 이용한다. 이 경우 원하는 페이지 테이블 항목이 TLB 내에 있다면 단지 한 번의 메인 메모리 접근이 필요하다. 접근의 90%가 TLB 내에 있고, 10%는 TLB 실패(miss)를 일으킨다. TLB 실패 중 20%가 페이지 부재(page fault)를 일으키며, 교체될 페이지의 20%가 변경된 페이지(dirty page)라고 할 때, 아래 <조건>과 같은 시스템의 실질(effective) 메모리 접근 시간으로 옳은 것은? [2.5점]

- <조건>—
- TLB의 접근 시간은 0μs이다.
 - 메인 메모리 접근 시간은 1μs이다.
 - 디스크 시스템은 접근하고 전송하는 데 평균 10μs가 소요된다.
 - 페이지 부재 처리 시간에서 인터럽트 처리와 프로세스 재시작 시간은 무시하며, 페이지 테이블은 항상 메인 메모리에 있다.

- ① 1.252μs ② 1.3μs ③ 1.34μs
 ④ 1.9μs ⑤ 2.0μs

17. 다음 <조건>에서 어떤 파일의 논리 블록 4(네 번째 논리 블록)에 접근하고자 한다. 이때 파일 시스템에서 파일의 메모리 할당 기법인 연속(continuous) 할당, 연결(linked) 할당, 색인(indexed) 할당 기법을 적용할 때, 각 메모리 할당 기법에 대한 최소 디스크 접근 횟수의 합은? [1.5점]

- <조건>—
- 파일 포인터는 현재 논리 블록 10에 위치하고 있다.
 - 파일 시스템의 논리 블록 크기와 물리 블록 크기는 1024 바이트이다.
 - 각 파일에 대한 디렉토리 정보는 항상 메인 메모리에 존재한다.
 - 디렉토리 정보에 등록된 모든 파일은 외부 단편화가 발생하지 않는다.
 - 색인 할당 기법의 경우 각 파일은 한 개의 색인 블록을 사용하고, 고정 크기 블록을 지원한다.
 - 연결 할당 기법의 경우 여러 블록을 하나로 묶는 클러스터 단위 할당을 지원하지 않으며, 물리 블록 간의 연결 정보는 시작 블록에서 마지막 블록으로의 단방향 연결리스트를 사용한다.

- ① 6회 ② 7회 ③ 8회
 ④ 11회 ⑤ 12회

18. 다음과 같이 정의된 덱(deque) 연산을 이용하여 스택과 큐의 삽입, 삭제 연산을 설계하고자 한다. <보기>의 ㉠, ㉡, ㉢에 들어갈 내용으로 옳은 것은? [1.5점]

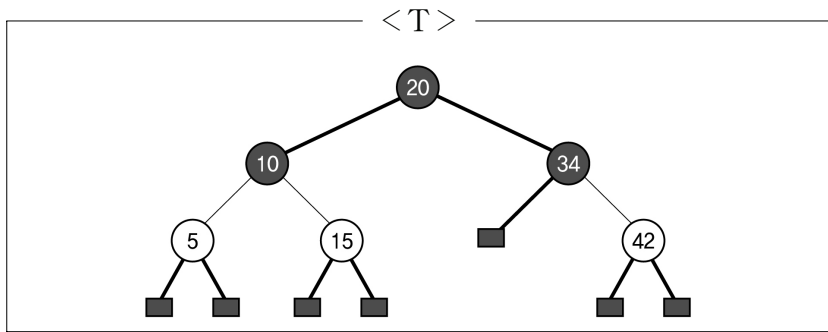
ADT Deque
 데이터 : 0개 이상의 원소를 가진 유한 순서 리스트
 연산 :
 insertFirst() ::= 덱 앞(front)에 새로운 원소 삽입
 insertLast() ::= 덱 뒤(rear)에 새로운 원소 삽입
 deleteFirst() ::= 덱 앞(front)의 원소 삭제
 deleteLast() ::= 덱 뒤(rear)의 원소 삭제
End Deque

<보기>

	스택 원소 삽입	스택 원소 삭제	큐 원소 삽입	큐 원소 삭제
덱 연산	㉠	deleteLast()	㉡	㉢

- | | | | |
|---|---------------|---------------|---------------|
| | ㉠ | ㉡ | ㉢ |
| ① | insertFirst() | insertFirst() | deleteFirst() |
| ② | insertFirst() | insertLast() | deleteFirst() |
| ③ | insertFirst() | insertLast() | deleteLast() |
| ④ | insertLast() | insertFirst() | deleteFirst() |
| ⑤ | insertLast() | insertLast() | deleteFirst() |

19. 주어진 레드-블랙(Red-Black) 트리 T의 연산과, 레드-블랙 트리와 B-트리의 관계에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은? (단, 트리 T는 확장 이진탐색트리이고, 검은색 노드는 블랙 노드를, 흰색 노드는 레드노드를 나타내며, 사각형 모양은 실패 노드를 나타낸다.)



<보기>

- ㉠. 트리 T에 키(key) 값 12를 삽입하면, 삽입한 노드의 부모 노드와 조부모노드는 색깔의 반전이 필요하다.
- ㉡. 트리 T에 키 값 50을 삽입하면 회전이 필요하고, 회전 후 키 값 34가 저장된 노드는 키 값 42의 자식노드인 레드노드로 변환된다.
- ㉢. 트리 T에서 삭제될 노드가 레드노드이면, 삭제된 후 레드-블랙 트리의 성질은 그대로 유지된다.
- ㉣. 트리 T를 2-3-4 트리(즉, 차수가 4인 B-트리)로 변환하여 표현하면 루트노드의 직계 오른쪽 자식노드는 키 값 34와 42를 갖는 3노드가 된다.

- | | | |
|-----------|--------------|-----------|
| ① ㉠, ㉡ | ② ㉢, ㉣ | ③ ㉠, ㉡, ㉢ |
| ④ ㉡, ㉢, ㉣ | ⑤ ㉠, ㉡, ㉢, ㉣ | |

20. calculateExp()는 스택을 이용하여 후위표기법(postfix)으로 표현된 수식 exp를 계산하는 알고리즘이다. 다음 <조건>에서 exp가 “234+*562/+-\$”일 때, 옳은 것을 <보기>에서 고른 것은?

```
calculateExp(s1, exp) {
  int x, y, z;
  while ((token = getChar(exp)) != '$') {
    if (token == operand) push(s1, token);
    else if (token == operator) {
      x = pop(s1);
      y = pop(s1);
      z = (㉠) operator (㉡);
      push(s1, z);
    }
  }
  print(pop(s1));
}
```

<조건>

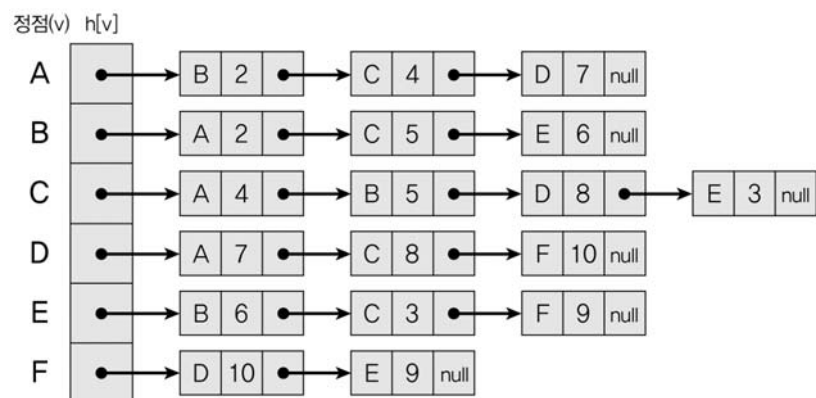
- s1은 스택을 나타내고 초기 상태는 공백이다.
- operator는 연산자, operand는 피연산자를 의미한다.
- 연산자는 '+', '-', '*', '/' 중 하나이고, 피연산자는 0~9 사이의 한 자리 숫자이며, 끝식별자(delimiter)는 '\$'이다.
- exp는 연산자, 피연산자, 끝식별자로 구성된다.
- token은 연산자, 피연산자, 끝식별자 중 하나이다.
- getChar()는 exp로부터 하나의 token을 차례대로 읽는 함수이다.
- push()는 스택에 하나의 원소를 삽입하는 함수이며, pop()은 스택에서 하나의 원소를 삭제하여 반환하는 함수이다.
- print()는 주어진 값을 출력하는 함수이다.

<보기>

- ㉠. ㉠은 x이고, ㉡는 y이다.
- ㉡. calculateExp()의 출력 결과는 6이다.
- ㉢. calculateExp()를 수행하는 동안 push()의 전체 호출 횟수는 11회이다.
- ㉣. while문 수행 중 token이 '*' 인 경우를 수행하고 나면 스택에 들어 있는 원소는 1개이다.
- ㉤. while문 수행 중 token이 '/' 인 경우를 수행하고 나면 스택의 제일 위에 들어 있는 원소 값은 0이다.

- | | | |
|-----------|-----------|-----------|
| ① ㉠, ㉡, ㉢ | ② ㉠, ㉡, ㉣ | ③ ㉡, ㉢, ㉣ |
| ④ ㉡, ㉢, ㉤ | ⑤ ㉢, ㉣, ㉤ | |

21. 다음의 인접리스트는 무방향 그래프 G를 <조건>에 따라 표현한 것이다. Kruskal 알고리즘을 이용하여 그래프 G의 최소비용신장트리(minimum cost spanning tree)를 구하고자 한다. 이에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은?



<조 건>

- 인접리스트의 h[v]는 정점노드 v의 헤더노드이며 정점 v의 인접리스트에 대한 포인터이다.
- 인접리스트의 정점노드는 정점, 가중치, 링크 필드 순서로 구성되어 있다.
- (u, v)는 임의의 정점 u, v를 연결하는 간선을 나타낸다.
- 최소비용신장트리의 간선 집합을 T, 그래프 G의 정점 수를 n, 간선 수를 e로 나타낸다.
- Kruskal 알고리즘에서 간선을 선택할 때 최소 힙(min heap)을 이용한다.

<보 기>

- ㄱ. 알고리즘 종료 후 T는 {(A, B), (A, C), (A, D), (C, E), (E, F)}이다.
- ㄴ. T를 구하는 과정에서 사이클을 형성하여 거부된 간선의 집합은 {(B, C), (B, E), (C, D), (D, F)}이다.
- ㄷ. T에 세 번째로 추가된 간선은 (A, D)이다.
- ㄹ. 알고리즘 종료 후 T의 간선 수가 n-1개보다 적으면 그래프 G는 단절 그래프(disconnected graph)이다.
- ㅁ. Kruskal 알고리즘의 시간복잡도는 $O(e \log_2 e)$ 이다.

- ① ㄱ, ㄹ ② ㄱ, ㄴ, ㄷ ③ ㄱ, ㄴ, ㄹ
 ④ ㄱ, ㄹ, ㅁ ⑤ ㄷ, ㄹ, ㅁ

22. 다음 <조건>에서 십진수 A = -1.375와 IEEE 754 표준 단정도(single-precision) 부동 소수점 형식을 사용하는 <보기>의 B를 더한 결과인 S를 나타내고자 한다. ㉠과 ㉡에 들어갈 내용으로 옳은 것은?

<조 건>

- IEEE 754 표준 단정도 부동 소수점 형식은 다음과 같다.
 $(-1)^{\text{부호}} \times (1 + \text{가수}) \times 2^{(\text{지수} - \text{바이어스})}$
- 바이어스(bias) 값은 127이며, 부호(sign), 지수(exponent), 가수(fraction)가 상위비트부터 하위비트 순으로 배치된다.

<보 기>

	부호	지수부	가수부	
비트위치 →	31	30 ~ 23	22 ~ 15	14 ~ 0
B →	1	10000000	11000000	0000000000000000
S →	1	㉠	㉡	0000000000000000

㉠

㉡

- ① 10000000 00111000
 ② 10000001 00111000
 ③ 10000001 11000000
 ④ 00000100 11000000
 ⑤ 00000100 01110000

23. RISC(Reduced Instruction Set Computer) 명령어 집합의 실행 시간이 다음과 같을 때, 파이프라이닝(pipelining) 유무에 따른 중앙처리장치(CPU) 실행시간을 다음 <조건>에서 구하고자 한다. 3개의 명령어 add, beq, lw에 대해 add→beq→lw 순서로 각 명령어를 한 번씩 실행할 때, <경우> A, B의 중앙처리장치 실행 시간으로 옳은 것은?

(단위 : pico second)

명령어 종류	각 명령어의 실행 단계					명령어별 전체 실행시간
	명령어 인출	레지스터 읽기	ALU 연산	데이터 접근	레지스터 쓰기	
Load 명령어 (lw)	100	50	100	100	50	400
Store 명령어 (sw)	100	50	100	100		350
연산 명령어 (add)	100	50	100		50	300
분기 명령어 (beq)	100	50	100			250

<조 건>

- 파이프라이닝 실행 과정에서 해저드(hazard)는 발생하지 않는다.
- 중앙처리장치 실행시간이란 메모리 접근 등을 고려하지 않고 중앙처리장치가 명령어를 실행하는 시간만을 의미한다.
- 파이프라이닝의 경우 모든 명령어의 각 실행 단계를 한 사이클(cycle)에 실행한다.

<경 우>

- A : 명령어를 파이프라이닝 없이 모두 동일하게 한 사이클에 하나씩 실행하는 경우
- B : 명령어의 각 실행 단계를 모두 동일하게 중첩하여 한 사이클에 실행하는 파이프라이닝 실행을 하는 경우

	①	②	③	④	⑤
경우 A	1200	1200	950	950	950
경우 B	700	600	700	600	500

24. 다음 <조건>을 만족하는 명령어 집합을 설계하고자 한다. 옳은 것을 <보기>에서 모두 고른 것은?

—<조 건>—

- 레지스터 총 개수는 16개이다.
- 명령어 총 개수는 12개이며, 이들 명령어는 3개의 레지스터 (소스 레지스터 RA와 RB, 연산 결과 저장 레지스터 RC)를 사용하는 레지스터 연산을 한다.
- 레지스터와 명령코드(opcode)를 이용한 레지스터 연산 명령어 실행의 의미는 다음과 같다.
 $RC \leftarrow RA \text{ opcode } RB$
- 모든 명령어의 크기는 동일하다.

—<보 기>—

- ㄱ. 명령코드는 최소 4비트가 필요하다.
- ㄴ. 각 레지스터 필드는 모두 동일하며, 최소 5비트가 필요하다.
- ㄷ. 명령어 크기는 최소 2바이트가 필요하다.
- ㄹ. 각 명령어는 opcode, RA, RB, RC 등 최소 4개 필드가 필요하다.
- ㅁ. 명령어 내부 필드 형식을 변경하지 않는다면 레지스터 연산을 하는 새로운 명령어를 추가할 수 없다.

- ① ㄱ, ㄴ, ㄷ ② ㄱ, ㄷ, ㄹ ③ ㄱ, ㄷ, ㅁ
 ④ ㄱ, ㄹ, ㅁ ⑤ ㄴ, ㄷ, ㄹ, ㅁ

25. 다음 부울 함수 F에 대해 카르노 맵(K-map)을 이용하여 입력 변수와 항의 수를 최소로 간소화한 부울 함수 F를 <보기>에서 모두 고른 것은? (단, 입력 변수 A, B, C, D와 \bar{A} , \bar{B} , \bar{C} , \bar{D} 는 각각 동등한 간소화 정도로 본다.)

$F(A, B, C, D) = \Sigma(0, 1, 2, 5, 7, 8, 9, 10, 11, 13, 15)$

—<보 기>—

- ㄱ. $F = BD + \bar{B}\bar{D} + AD + A\bar{B}$
- ㄴ. $F = BD + \bar{B}\bar{D} + AD + \bar{B}\bar{C}$
- ㄷ. $F = BD + \bar{B}\bar{D} + A\bar{B} + \bar{C}D$
- ㄹ. $F = BD + \bar{B}\bar{D} + A\bar{B} + \bar{B}\bar{C}$
- ㅁ. $F = BD + \bar{B}\bar{D} + AD + \bar{C}D$

- ① ㄱ, ㄴ ② ㄹ, ㅁ ③ ㄷ, ㄹ, ㅁ
 ④ ㄱ, ㄴ, ㄷ, ㄹ ⑤ ㄴ, ㄷ, ㄹ, ㅁ

26. 어떤 컴퓨터 시스템이 직접 사상(direct mapping) 방식의 캐시 메모리를 사용하고, 이 캐시 메모리는 블록당 한 워드(32비트)씩 8개 워드를 저장한다. 다음 <조건>에서 아래와 같이 6비트 이진 블록주소가 중앙처리장치(CPU)로부터 요청될 때, 처리 후 캐시 메모리 내부 ㉠, ㉡, ㉢에 들어갈 내용으로 옳은 것은?

6비트 이진 블록주소 요청 처리 순서 :
 100010 → 111000 → 101010 → 110110 → 111000

—<조 건>—

- 캐시 메모리의 초기 상태는 비어 있다.
- 직접 사상 방식은 블록주소를 캐시 메모리의 블록 수로 나머지(modulo) 연산한 값을 인덱스(index)로 사용한다.

인덱스	태그(tag)	블록 데이터
0	㉠	데이터
1		
2	㉡	데이터
3		
4		
5		
6	㉢	데이터
7		

- | | ㉠ | ㉡ | ㉢ |
|---|-----|-----|-----|
| ① | 110 | 100 | 101 |
| ② | 110 | 100 | 111 |
| ③ | 111 | 100 | 110 |
| ④ | 111 | 101 | 110 |
| ⑤ | 010 | 101 | 110 |

27. 데이터베이스 관리 시스템(DBMS)에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은? [1.5점]

—<보 기>—

- ㄱ. 정의(definition) 기능에는 데이터의 논리적 구조와 물리적 구조 사이에 변환이 가능하도록 사상(mapping)에 대한 명세를 포함한다.
- ㄴ. 조작(manipulation) 기능은 DBMS가 데이터베이스 사용자의 요구에 따라 데이터베이스를 접근하게 하고 데이터베이스 구조를 변경할 수 있는 기능을 제공한다.
- ㄷ. 데이터 정의어(DDL)는 정의 기능을 위한 데이터 언어로 데이터베이스 스키마를 컴퓨터가 이해할 수 있게 기술하며, DDL 컴파일러는 기술된 스키마를 번역하여 저장 데이터베이스(stored databases)에 등록한다.
- ㄹ. 표준 SQL 언어에서 DELETE, UPDATE 문은 조작 기능을 지원하는 데이터 조작어(DML)로 저장 데이터베이스의 데이터를 삭제하거나 갱신하는 기능을 제공한다.

- ① ㄱ, ㄴ ② ㄱ, ㄹ ③ ㄴ, ㄷ
 ④ ㄱ, ㄷ, ㄹ ⑤ ㄴ, ㄷ, ㄹ

28. 다음은 C 언어로 작성된 힙 정렬(heap sort) 프로그램이다. 이에 대한 설명으로 옳은 것을 <보기>에서 모두 고른 것은? (단, ary[0]은 정렬할 원소로 사용되지 않는다.) [2.5점]

```
#include <stdio.h>
void swap(int *x, int *y) {
    int temp = *x; *x = *y; *y = temp;
}
void construct(int ary[], int j, int n) {
    int k, buf = ary[j];
    for (k = 2*j; k <= n; k = k*2) {
        if (k < n && ary[k] > ary[k+1]) k++;
        if ( ) break;
        ary[k/2] = ary[k];
    }
    ary[k/2] = buf;
}
void heapSort(int ary[], int n) {
    int j;
    for (j = n/2; j > 1; j--) construct(ary, j, n);
    for (j = n; j > 1; j--) {
        construct(ary, 1, j);
        swap(&ary[1], &ary[j]);
    }
}
void main() {
    int ary[] = {0, 20, 15, 6, 28, 12, 10, 9};
    heapSort(ary, (sizeof(ary)/sizeof(int))-1);
}
```

<보기>

- ㄱ. ㉠은 buf >= ary[k]이다.
- ㄴ. heapSort()에 있는 첫 번째 for문을 수행하고 나면 ary[]의 원소 값은 순서대로 0, 20, 12, 6, 28, 15, 10, 9가 된다.
- ㄷ. heapSort()에 있는 두 번째 for문에서 j = 6인 경우 반복문을 수행하고 나면 ary[]의 원소 값은 순서대로 0, 20, 12, 10, 28, 15, 9, 6이 된다.
- ㄹ. 프로그램을 실행했을 때, construct()의 밑줄 친 부분 ㉡는 총 8회 수행된다.
- ㅁ. 프로그램이 실행된 후 배열의 원소들은 오름차순으로 정렬된다.

- ① ㄴ, ㄷ ② ㄷ, ㄹ ③ ㄷ, ㅁ
- ④ ㄴ, ㄷ, ㄹ ⑤ ㄱ, ㄴ, ㄹ, ㅁ

29. 다음의 관계형 데이터베이스 스키마에 대해 <보기>에서 주어진 관계대수와 SQL문을 수행했을 때, 그 실행 결과 값이 서로 같지 않은 것을 모두 고른 것은? (단, 선택(selection)은 σ, 프로젝트는 Π, 자연조인은 ⋈_N, 집계함수(aggregate function)는 F, 평균집계함수는 AVG()이다.) [2.5점]

○ 스키마 : 학생(학번, 이름, 학년, 학과명)
 과목(과목번호, 과목명, 학점수)
 수강(학번, 과목번호, 중간고사)

○ 밑줄은 기본키를 의미하며, '수강' 릴레이션의 '학번', '과목번호'는 '학생'과 '과목' 릴레이션을 참조하는 외래키이다.

<보기>

- ㄱ. Π학번(σ과목명='컴퓨터개론'(과목)⋈_N(Π과목번호(σ중간고사>94(수강))))
 select 학번 from 과목 as B, 수강 as C
 where B.과목번호 = C.과목번호 and
 과목명='컴퓨터개론' and 중간고사 > 94;
- ㄴ. Π이름, 학과명(σ학년 ≠ 4(학생))
 select distinct 이름, 학과명 from 학생 where 학년 ≠ 4;
- ㄷ. Π이름, 과목번호, 중간고사((Π학번(σ과목명='컴퓨터교육'(학생))⋈_N수강))
 select distinct 이름, 과목번호, 중간고사 from 학생, 수강
 where 학생.학번 = 수강.학번 and 학과명 = '컴퓨터교육';
- ㄹ. 학과명 F AVG(중간고사)(Π학과명, 중간고사(학생⋈_N수강))
 select 학과명, AVG(중간고사) from 학생, 수강
 where 학생.학번 = 수강.학번 group by 학과명;
- ㅁ. Π이름, 학년(학생⋈_N(Π학번(σ과목번호 = 'C001' ∧ 중간고사 > 89(수강))))
 select distinct 이름, 학년 from 학생 as A, 수강 as C
 where A.학번 = 수강.학번 and C.과목번호 = 'C001'
 and 중간고사 > 89;

- ① ㄱ, ㄷ ② ㄴ, ㄹ ③ ㄷ, ㄹ
- ④ ㄱ, ㄷ, ㅁ ⑤ ㄱ, ㄹ, ㅁ

30. 데이터베이스 항목 A, B, C, D, E의 값을 읽고 갱신하는 5개 트랜잭션 T1, T2, T3, T4, T5의 실행 중에 다음의 로그파일이 생성되었다. DBMS가 로그를 이용하는 즉시갱신(immediate update) 회복 기법을 사용할 때, 이 로그파일의 로그순서번호 16을 마친 시점에서 시스템이 파손되었다고 가정하자. 시스템이 재가동되어 트랜잭션의 복구 연산을 마쳤을 때, 옳은 설명을 <보기>에서 모두 고른 것은? (단, 복구 연산에는 REDO(재실행), UNDO(취소), NO-REDO(재실행 없음), NO-UNDO(취소 없음)가 있고, 로그 레코드는 [트랜잭션번호, 데이터항목, 이전값, 새로운값]이다.)

로그순서번호	로그레코드	로그순서번호	로그레코드
1	[T1, start]	9	checkpoint(a2)
2	[T1, A, 20, 40]	10	[T5, start]
3	[T1, commit]	11	[T5, C, 10, 50]
4	[T3, start]	12	[T2, commit]
5	[T3, B, 90, 190]	13	[T4, start]
6	checkpoint(a1)	14	[T4, D, 70, 130]
7	[T2, start]	15	[T5, commit]
8	[T2, A, 40, 80]	16	[T4, E, 130, 150]

<보기>

- ㄱ. 데이터베이스 항목 A와 C의 값은 80과 50으로 갱신된다.
- ㄴ. 트랜잭션 T1은 NO-UNDO/NO-REDO의 복구 연산을 수행한다.
- ㄷ. 트랜잭션 T3는 항목 B의 값을 190으로 메인 메모리의 데이터베이스 버퍼에 저장한 후 checkpoint(a1)에 의해 이 값을 디스크로 출력한다.
- ㄹ. UNDO 연산은 트랜잭션 T3 → T4 순으로 수행된다.
- ㅁ. REDO 연산은 트랜잭션 T2 → T5 순으로 수행된다.

- ① ㄱ, ㄴ ② ㄴ, ㄹ ③ ㄱ, ㄴ, ㅁ
- ④ ㄱ, ㄷ, ㄹ ⑤ ㄷ, ㄹ, ㅁ

31. 다음의 대학 <업무기술서>를 기반으로 데이터베이스를 구축하고자 한다.

- <업무기술서> —
- OO대학은 여러 과목을 개설하며, 각 과목은 과목코드, 과목명, 시간수에 대한 정보를 가지고 있다.
 - 각 과목은 전공 분야에 따라 여러 교수가 강의할 수 있다.
 - 강의는 정해진 시간과 장소에서 진행된다.
 - 교수는 교수번호와 교수명으로 관리된다.
 - 각 교수는 여러 과목을 강의할 수 있으며, 반드시 한 학과에 소속된다.
 - 각 학과에는 여러 교수가 소속될 수 있다.
 - 학과는 학과코드, 학과명으로 관리된다.
 - 학생들은 여러 과목에 등록하여 수업을 들을 수 있고, 등록과 동시에 성적과 학점이 관리되도록 한다.
 - 학생 정보에는 학번, 이름, 전화번호가 있다.

위의 <업무기술서>로부터 데이터베이스 개념적 설계를 수행하여 E-R 다이어그램(Entity-Relationship Diagram)을 도출하였을 때, 도출되는 개체타입(entity type)은 '과목', '교수', '학과', '학생'이다. 이 개체타입들을 가지고 설계된 E-R 다이어그램과 관련된 내용으로 옳은 것을 <보기>에서 모두 고른 것은?

(단, E-R 다이어그램 설계 시 개체타입, 관계타입, 속성은 <업무기술서>에 기술된 단어만으로 제한하고, 기본키와 외래키는 <업무기술서>에서 가장 적합한 것으로 도출하며, <보기>의 밑줄 친 속성은 기본키를 나타낸다.)

- <보 기> —
- ㄱ. '교수'와 '과목' 개체타입은 '강의' 관계타입으로 다대다 관계를 가지고 있으며, '학과'와 '교수' 개체타입은 일대다 관계를 갖고 있다.
 - ㄴ. '교수' 개체타입은 관계 스키마로 변환될 때 교수(교수번호, 교수명, 학과코드) 스키마로 변환될 수 있으며, '교수' 릴레이션의 '학과코드'는 '학과' 릴레이션을 참조하는 외래키이다.
 - ㄷ. '강의' 관계타입은 관계 스키마로 변환될 때 강의(교수번호, 과목코드, 시간, 장소) 스키마로 변환될 수 있다.
 - ㄹ. '학생'과 '과목' 개체타입이 연결된 '등록' 관계타입은 관계 스키마로 변환될 때 학생(학번, 이름, 전화번호, 과목코드) 스키마로 변환되며, '학생' 릴레이션의 '과목코드'는 '과목' 릴레이션을 참조하는 외래키이다.

- ① ㄱ, ㄴ ② ㄱ, ㄷ ③ ㄴ, ㄹ
 ④ ㄱ, ㄴ, ㄷ ⑤ ㄱ, ㄴ, ㄷ, ㄹ

32. 다음은 배열값을 이용하여 연결리스트를 만들고 그 내용을 확인하는 C 언어 프로그램이다. 밑줄 친 명령문 ㉠가 세 번째로 수행될 때 출력되는 값으로 옳은 것은? (단, malloc() 함수는 정상적으로 메모리를 할당한다.) [1.5점]

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int id;
    struct node *link;
};
void main() {
    int a[5][5] = {{0,1,0,1,1}, {1,0,1,1,0},
                  {0,1,0,0,1}, {1,1,0,0,0}, {1,0,1,0,0}};
    struct node *h[5], *t;
    int i, j;
    for(i = 0; i < 5; i++) {
        h[i] = NULL;
        for(j = 0; j < 5; j++) {
            if(a[i][j]) {
                t = (struct node *)malloc(sizeof(struct node));
                t->id = j;
                t->link = h[i];
                h[i] = t;
            }
        }
    }
    for(i = 0; i < 5; i++) {
        t = h[i];
        while(t) {
            ㉠printf("(%d, %d) ", i, t->id);
            t = t->link;
        }
        printf("\n");
    }
}
```

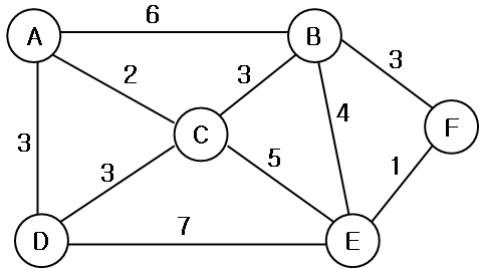
- ① (0, 1) ② (0, 2) ③ (0, 4)
 ④ (2, 1) ⑤ (2, 4)

33. 다음은 인터넷 프로토콜인 IPv6와 IPv4를 비교하여 설명한 것이다. 옳은 것을 <보기>에서 모두 고른 것은? [1.5점]

- <보 기> —
- ㄱ. IPv6의 주소공간은 IPv4보다 2^{28} 개 증가하였다.
 - ㄴ. IPv6는 IPv4보다 라우팅을 고속으로 처리할 수 있도록 데이터그램 헤더 형식을 단순화하였다.
 - ㄷ. IPv6는 데이터그램 헤더에 서비스 유형(type of service) 필드를 사용하여 IPv4보다 실시간 멀티미디어 전송에 적합하도록 설계되었다.
 - ㄹ. IPv6에서 단편화는 발신지 호스트에서만 수행될 수 있으나, IPv4에서는 단편화가 패킷이 전파되는 경로상의 모든 라우터에서 수행될 수 있다.
 - ㅁ. IPv6는 데이터그램 헤더에 인증을 위한 확장헤더를 사용하여 데이터그램의 비밀성과 무결성을 제공하도록 설계되었으나, IPv4의 데이터그램 헤더에서는 제공되지 않는다.

- ① ㄱ, ㄴ ② ㄴ, ㄹ ③ ㄱ, ㄷ, ㄹ
 ④ ㄴ, ㄹ, ㅁ ⑤ ㄴ, ㄷ, ㄹ, ㅁ

34. 다음과 같이 6개의 라우터로 구성된 네트워크 도메인에서 각 라우터는 링크상태(link state) 라우팅 기법을 사용하여 라우팅 테이블(routing table)을 구성한다. 다음 <조건>에서 각 라우터가 생성한 라우팅 테이블로 옳은 것을 <보기>에서 모두 고른 것은?



<조 건>

- 원은 라우터이고, 선은 링크를 표시한다.
- 각 링크상의 숫자는 전송비용(metric)을 나타내며, 양방향 동일하다.
- 자기 자신으로의 전송비용은 0이다.
- 라우터를 경유할 때 각 라우터에서의 지연은 없다.
- 라우터와 링크는 항상 정상적으로 동작한다.
- 링크상태 패킷은 모든 라우터에 안전하게 도달된다.
- 라우팅 테이블은 라우터, 비용, 다음라우터 필드로 구성된다.
- 라우터 필드는 목적지 라우터를, 비용 필드는 목적지까지의 최소 경로비용을, 다음라우터 필드는 최초로 경유할 라우터를 나타내며 인접 라우터로 직접 전달이 가능하면 '-'로 표시한다.

<보 기>

ㄱ. 라우터 A의 라우팅 테이블 ㄴ. 라우터 B의 라우팅 테이블

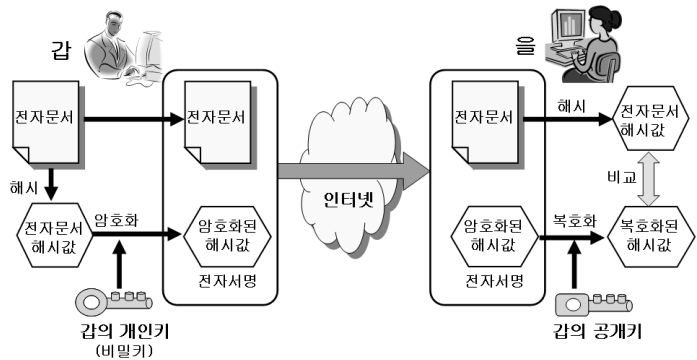
라우터	비용	다음라우터	라우터	비용	다음라우터
A	0	-	A	5	C
B	5	C	B	0	-
C	2	-	C	3	-
D	3	-	D	6	C
E	7	C	E	4	-
F	9	B	F	3	-

ㄷ. 라우터 D의 라우팅 테이블 ㄹ. 라우터 E의 라우팅 테이블

라우터	비용	다음라우터	라우터	비용	다음라우터
A	3	-	A	7	C
B	6	C	B	4	-
C	3	-	C	5	-
D	0	-	D	7	-
E	7	-	E	0	-
F	8	E	F	1	-

- ① ㄱ, ㄴ ② ㄱ, ㄹ ③ ㄴ, ㄷ
 ④ ㄱ, ㄴ, ㄷ ⑤ ㄴ, ㄷ, ㄹ

35. 다음 그림은 공개키 암호 알고리즘을 이용한 전자서명 시스템의 개념도로서, 송신자 '갑'과 수신자 '을'이 전자문서를 송수신하는 과정을 나타낸 것이다. 다음 <조건>에서 그림과 같이 동작할 때 옳은 것을 <보기>에서 모두 고른 것은?



<조 건>

- 해시 함수의 해시값은 충돌을 일으키지 않는다.
- 인터넷상에서 전송 에러가 발생하지 않는다.
- '갑'의 개인키는 안전하며, 문서 송수신 동안에 '갑'의 개인키와 공개키는 변경되지 않는다.

<보 기>

- ㄱ. '을'은 전자서명의 서명자가 '갑'임을 확인할 수 있다.
- ㄴ. 전자서명된 전자문서의 무결성을 보장할 수 없다.
- ㄷ. 서명자인 '갑'이 서명한 사실을 부인할 수 없다.
- ㄹ. '을'은 서명자인 '갑'이 전자문서를 송신하였음을 확인할 수 있다.
- ㅁ. 전자문서의 해시값을 전자서명에 사용하였으므로 전자문서의 비밀성을 보장할 수 있다.

- ① ㄱ, ㄴ, ㄷ ② ㄱ, ㄴ, ㄹ ③ ㄱ, ㄷ, ㄹ
 ④ ㄴ, ㄷ, ㅁ ⑤ ㄱ, ㄷ, ㄹ, ㅁ

36. 다음은 Pascal 유형의 의사코드로 작성된 프로그램이다. fun() 프로시저를 호출할 때 값-결과-전달(pass-by-value-result), 참조-전달(pass-by-reference), 이름-전달(pass-by-name) 매개변수 전달 기법을 각각 사용할 경우 밑줄 친 명령문 ㉠의 출력 결과로 옳은 것은?

```

program test;
var c : array[1..5] of integer;
    t : integer;
procedure fun(a, b : integer);
begin
    a := a * 2; b := b + a;
    a := a + 1; b := b + 1
end;
begin
    c[1] := 3; c[2] := 1; c[3] := 4;
    c[4] := 6; c[5] := 5;
    t := 1;
    fun(t, c[t]);
    ㉠ write(c[t])
end.
    
```

	값-결과-전달	참조-전달	이름-전달
①	3	4	5
②	3	5	5
③	4	4	5
④	4	5	5
⑤	4	5	6

37. 어떤 회사가 인터넷 관리 기관으로부터 200.20.64.0/22로 시작하는 클래스 없는(classless) IP 주소 블록을 부여받았다. 다음과 같이 24개의 서브넷이 필요하여 가변길이 서브넷을 구성하고자 한다. 부여받은 주소 블록을 첫 번째 그룹부터 시작하여 낮은 주소부터 차례로 할당할 때, 옳은 것을 <보기>에서 고른 것은? [2.5점]

- 첫 번째 그룹은 각각 64개의 주소를 가지는 2개의 서브넷
- 두 번째 그룹은 각각 32개의 주소를 가지는 2개의 서브넷
- 세 번째 그룹은 각각 16개의 주소를 가지는 4개의 서브넷
- 네 번째 그룹은 각각 4개의 주소를 가지는 16개의 서브넷

—<보 기>—

- ㄱ. 첫 번째 그룹의 첫 번째 서브넷은 서브넷 마스크가 255.255.254.0이고, 마지막 주소가 200.20.64.63/23이다.
- ㄴ. 두 번째 그룹의 두 번째 서브넷은 시작 주소가 200.20.64.160/27이다.
- ㄷ. 세 번째 그룹의 두 번째 서브넷은 시작 주소가 200.20.64.208/28이며, 마지막 주소가 200.20.64.223/28이다.
- ㄹ. 네 번째 그룹의 마지막 서브넷은 서브넷 마스크가 255.255.255.240이다.
- ㅁ. 부여받은 주소 블록에서 모든 서브넷의 주소를 할당한 후 여분의 주소 개수는 704개이다.

- ① ㄱ, ㄴ, ㄷ ② ㄱ, ㄴ, ㅁ ③ ㄴ, ㄷ, ㄹ
- ④ ㄴ, ㄷ, ㅁ ⑤ ㄷ, ㄹ, ㅁ

38. 문법 G가 생성하는 언어 L(G)를 다음과 같이 정의한다.
 $L(G) = \{\omega \mid S \xrightarrow{*} \omega, \omega \text{는 종단(terminal) 기호로 만들어진 문장}\}$
 L(G)는 문법의 시작 기호 S로부터 유도하여 생성할 수 있는 모든 문장들의 집합이다. L(G)와 같은 언어를 생성하는 문법을 <보기>에서 모두 고른 것은? (단, 문법의 생성규칙에서 종단 기호는 소문자, 비종단(nonterminal) 기호는 대문자로 표현한다.)

G :
 S → aA | bB
 A → aA | C
 B → bB | b

—<보 기>—

ㄱ. S → bB B → bB b	ㄴ. S → aA X X → bX b
ㄷ. S → aA bC C → b bB B → bB b	ㄹ. S → bb bX X → bX b

- ① ㄱ, ㄴ, ㄷ ② ㄱ, ㄴ, ㄹ ③ ㄱ, ㄷ, ㄹ
- ④ ㄴ, ㄷ, ㄹ ⑤ ㄱ, ㄴ, ㄷ, ㄹ

39. 다음은 비트별 논리곱(&) 연산자를 사용하지 않고 두 양의 정수에 대한 비트별 논리곱 연산을 구하는 C 언어 프로그램이다. ㉠과 ㉡에 알맞은 것은? (단, 정수형 변수의 크기는 32비트이며, 입력되는 두 양의 정수는 $2^{31}-1$ 보다 작다.) [2.5점]

```
#include <stdio.h>
void main() {
    int a, b, c, i, x, y;
    scanf("%d %d", &a, &b);
    x = 1; y = 2; c = 0;
    for(i = 0; i < 31; i++) {
        if( ㉠ ) ㉡ ;
        x = x * 2;
        y = y * 2;
    }
    printf("%d & %d => %d\n", a, b, c);
}
```

- | | | |
|---|----------------|-----------|
| | ㉠ | ㉡ |
| ① | a/x%y && b/x%y | c = c + 1 |
| ② | a/x%y && b/x%y | c = c + x |
| ③ | a/x%y && b/x%y | c = c + y |
| ④ | a%y/x && b%y/x | c = c + 1 |
| ⑤ | a%y/x && b%y/x | c = c + x |

40. 다음은 정렬된 배열에서 재귀함수를 이용하여 원하는 자료를 찾는 C 언어 프로그램이다. 이 프로그램을 실행하였을 때 출력된 결과로 옳은 것은? (단, 배열의 원소 값은 서로 다르다.)

```
#include <stdio.h>
int s = 0;
int search(int a[], int low, int high, int f) {
    int n, m;
    n = high - low;
    if(n < 0) return -1;
    if(n == 0) {
        if(f == a[high]) return high;
        else return -1;
    }
    m = n * (f - a[low]) / (a[high] - a[low]) + low;
    s = s + a[m];
    if(f == a[m]) return m;
    else {
        if(f < a[m]) return search(a, low, m-1, f);
        else return search(a, m+1, high, f);
    }
}
void main() {
    int a[11] = {2,3,5,12,22,42,43,62,72,92,105};
    int p, find = 22;
    p = search(a, 0, 10, find);
    if(p != -1) printf("%d\n", s);
    else printf("0\n");
}
```

- ① 0 ② 27 ③ 37
- ④ 42 ⑤ 81

- 수 고 하 셧 습 니 다 -

출 제 : 한국교육과정평가원