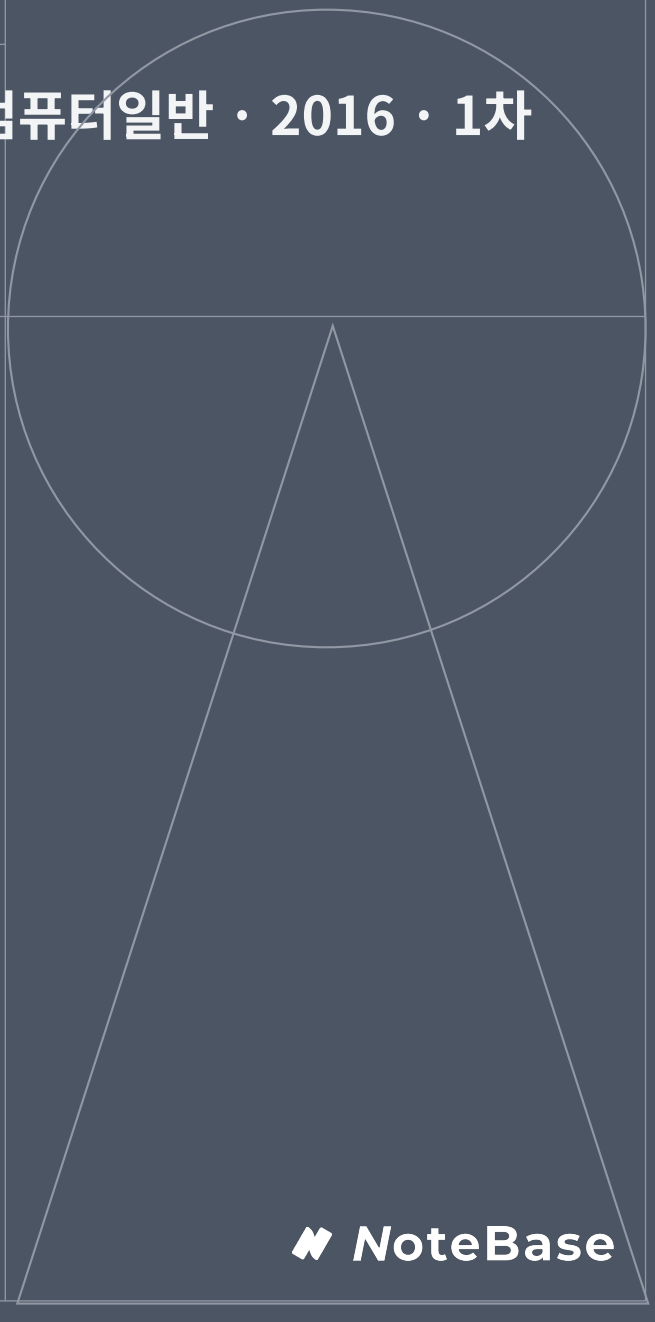
# 기출OX 해설집

공무원/해경/일반직 9급 · 컴퓨터일반 · 2016 · 1차



NoteBase

# 기출OX 해설집

## 공무원/해경/일반직 9급 · 컴퓨터일반 · 2016 · 1차

**공무원/해경/일반직 9급 · 컴퓨터일반 · 2016 · 1차**

**문 1. 인터럽트(interrupt) 발생 시 수행되는 작업순서로 옳은 것은?**

ㄱ. 해당 인터럽트 서비스 루틴을 호출한다.
ㄴ. 현재 프로그램 카운터에 적재된 명령어 주소를 스택에 저장한다.
ㄷ. 해당 인터럽트 서비스를 수행한다.
ㄹ. 스택에 저장된 명령어 주소를 프로그램 카운터에 저장한다.

① ㄱ→ㄴ→ㄹ→ㄷ
② ㄱ→ㄷ→ㄴ→ㄹ
③ ㄴ→ㄱ→ㄷ→ㄹ
④ ㄴ→ㄱ→ㄹ→ㄷ

**정답 3번**

**출제 포인트**

이 문제는 인터럽트가 발생했을 때 복귀 주소를 보존한 뒤 인터럽트 서비스 루틴으로 분기하고, 처리가 끝나면 원래 프로그램으로 복귀하는 순서를 묻는다.

**올바른 순서**

1. ㄴ — 현재 프로그램 카운터에 들어 있는 명령어 주소를 스택에 저장하여 복귀 주소를 보존한다.
2. ㄱ — 해당 인터럽트 서비스 루틴을 호출하여 제어를 넘긴다.
3. ㄷ — 인터럽트 서비스 루틴에서 해당 인터럽트 처리를 수행한다.
4. ㄹ — 처리가 끝나면 스택에 저장된 명령어 주소를 프로그램 카운터에 복원하여 중단된 프로그램으로 돌아간다.

왜 이 순서인가: 서비스 루틴으로 제어를 넘기기 전에 복귀 주소를 저장해야 하며, 인터럽트 서비스를 마친 뒤에야 그 주소를 복원할 수 있기 때문이다.

NoteBase

### 선지별 해설

① X — ㄱ→ㄴ은 복귀 주소를 저장하기 전에 인터럽트 서비스 루틴을 호출하는 순서이므로 잘못이다. 올바르게는 ㄴ을 먼저 수행한 뒤 ㄱ을 수행하며, 전체 순서는 ㄴ→ㄱ→ㄷ→ㄹ이다.
② X — ㄱ→ㄷ은 복귀 주소를 저장하기 전에 서비스 루틴 호출과 인터럽트 처리를 진행하는 순서이므로 잘못이다. 올바르게는 먼저 ㄴ으로 복귀 주소를 저장하고, ㄱ과 ㄷ을 거쳐 마지막에 ㄹ로 복원한다.
③ O — 복귀 주소 저장(ㄴ), 인터럽트 서비스 루틴 호출(ㄱ), 인터럽트 서비스 수행(ㄷ), 복귀 주소 복원(ㄹ)의 순서가 맞다.
④ X — ㄱ 다음에 ㄹ을 배치하여 인터럽트 서비스를 수행하기 전에 원래 프로그램의 주소를 복원한 것이 잘못이다. 올바르게는 ㄷ으로 인터럽트 서비스를 수행한 뒤 ㄹ로 복귀 주소를 복원하므로 ㄴ→ㄱ→ㄷ→ㄹ이다.

### 오답 포인트

호출(ㄱ)을 첫 단계로 생각하거나, 복원(ㄹ)을 서비스 수행(ㄷ)보다 앞에 두기 쉽다. 그러나 중단 지점으로 돌아오려면 호출 전에 복귀 주소를 저장해야 하고, 복원은 인터럽트 처리가 모두 끝난 뒤에 수행해야 한다.

### 다시 나오면 이렇게 판단

인터럽트 순서는 '복귀 정보 저장 → 서비스 루틴으로 제어 이전 → 인터럽트 처리 → 복귀 정보 복원'의 의존관계로 판단한다.

### 암기 포인트

인터럽트의 핵심 흐름은 저장 → 호출 → 처리 → 복원이다.

NoteBase

**문 2. C에서 '^'는 비트 간 XOR 연산을 나타낸다. 5^13의 결과로 옳은 것은?**

① 18

② 13

③ 8

④ 0

**정답 3번**

**출제 포인트**

C의 비트 XOR 연산자 ^를 이용해 두 정수의 대응 비트를 비교하고 결과를 10진수로 변환하는 문제이다.

**조건 정리**

주어진 값은 5와 13이고, 구해야 할 값은 5 ^ 13이다. 두 수를 같은 자릿수의 2진수로 맞춘 뒤 각 비트를 XOR해야 한다.

**사용할 공식**

XOR은 대응 비트가 서로 다르면 1, 같으면 0이 된다. 따라서 5와 13을 각각 0101과 1101로 나타내어 같은 위치의 비트끼리 비교한다.

**계산 과정**

1. 5를 4비트 2진수로 나타내면 0101이다.
2. 13을 4비트 2진수로 나타내면 1101이다.
3. 각 자리에서 XOR하면 0101 ^ 1101 = 1000이다.
4. 2진수 1000은 10진수 8이다.

**선지 판단**

① X — 18은 5와 13을 산술적으로 더한 값이다. '18'은 틀린 표현이며, 비트 XOR의 결과는 8이다.
② X — 13은 오른쪽 피연산자를 그대로 적은 값으로 XOR 결과가 아니다. '13'은 틀린 표현이며, 올바른 결과는 8이다.
③ O — 0101과 1101의 대응 비트를 XOR하면 1000이고, 이는 10진수 8이다.
④ X — 두 수의 모든 대응 비트가 같아야 XOR 결과가 0이 된다. 5와 13은 최상위 대응 비트가 다르므로 '0'은 틀린 표현이며, 올바른 결과는 8이다.

NoteBase

**오답 포인트**

^를 거듭제곱이나 덧셈으로 해석하는 것이 함정이다. C에서 정수 사이의 ^는 각 대응 비트를 비교하는 XOR 연산자이다.

**다시 나오면 이렇게 판단**

두 정수를 같은 길이의 2진수로 바꾸고, 각 자리에서 서로 다르면 1, 같으면 0을 적은 뒤 결과를 10진수로 변환한다.

**암기 포인트**

XOR은 다르면 1, 같으면 0이다.

NoteBase

**문 3. 다음은 용어에 대한 설명으로 옳지 않은 것은?**

① 텔넷(telnet)은 사용자가 원격지 호스트에 연결하여 이를 자신의 로컬 호스트처럼 사용하는 프로토콜이다.

② TCP/IP 프로토콜 중에서 UDP는 비연결성 데이터 전송방식을 사용하여 신뢰도가 높은 데이터 전송에 사용된다.

③ 텔넷(telenet), FTP, SMTP 등은 TCP/IP의 응용계층에 속하는 대표적인 프로토콜이다.

④ SNMP는 일반 사용자를 위한 응용 프로토콜이 아니고, 망을 관리하기 위한 프로토콜이다.

**정답 2번**

**출제 포인트**

TCP/IP 계층별 대표 프로토콜의 기능과 UDP의 비연결성·비신뢰성 특성을 구별하는 문제이다.

**핵심 개념**

텔넷은 원격 로그인, FTP는 파일 전송, SMTP는 전자우편 전송, SNMP는 네트워크 관리를 위한 응용 계층 프로토콜이다. UDP는 연결을 설정하지 않고 데이터를 보내는 전송 계층 프로토콜로, 전달·순서·중복 방지를 보장하지 않는다. 반면 TCP는 연결 지향적이며 확인 응답과 재전송 등을 통해 신뢰성 있는 전송을 제공한다.

**선지별 해설**

① O — 텔넷은 원격지 호스트에 접속하여 그 시스템을 로컬에서 사용하는 것처럼 명령을 실행할 수 있게 하는 원격 로그인 프로토콜이다.
② X — 틀린 표현은 'UDP가 신뢰도가 높은 데이터 전송에 사용된다'이다. UDP는 비연결형이며 전달 여부나 순서를 보장하지 않는 비신뢰성 전송을 제공한다. 신뢰성 있는 데이터 전송에는 연결 지향형인 TCP가 사용된다.
③ O — 텔넷, FTP, SMTP는 각각 원격 로그인, 파일 전송, 전자우편 전송 기능을 제공하는 TCP/IP 응용 계층의 대표 프로토콜이다.
④ O — SNMP는 네트워크 장비의 상태 정보를 수집하고 설정을 관리하는 망 관리용 응용 계층 프로토콜이므로 일반 사용자의 원격 로그인·파일 전송 같은 응용 서비스와 목적이 다르다.

**오답 포인트**

②는 '비연결성'이라는 올바른 특징 뒤에 '신뢰도가 높은 전송'이라는 TCP의 특징을 붙인 것이 함정이다. UDP는 연결 설정 부담이 적고 빠른 전송에 유리하다는 뜻이지, 신뢰성이 높다는 뜻이 아니다.

**다시 나오면 이렇게 판단**

전송 계층에서는 '연결·확인 응답·재전송·순서 보장'이 보이면 TCP, '비연결·낮은 오버헤드·전달 비보장'이 보이면 UDP로 판단한다. 응용 계층 프로토콜은 텔넷-원격 로그인, FTP-파일 전송, SMTP-메일 전송, SNMP-망 관리로 기능을 대응시킨다.

NoteBase

**암기 포인트**

TCP는 연결과 신뢰성, UDP는 비연결과 비보장이다. SNMP의 M은 Management, 즉 망 관리이다.

NoteBase

**문 4. 분산처리시스템에 대한 설명으로 옳지 않은 것은?**

① 분산되어 있는 자원을 공유할 수 있으며 분산 처리를 통해 컴퓨팅 성능을 향상시킬 수 있다.

② 계층 연결 구조의 경우 인접 형제 노드 간 통신은 부모 노드를 거치지 않고 이루어질 수 있다.

③ 성(star)형 연결 구조의 경우 중앙 노드에 부하가 집중되어 성능이 저하되거나 중앙 노드 고장 시 전체 시스템이 마비될 수 있다.

④ 다중 접근 버스 연결 구조의 경우 한 노드의 고장이 다른 노드의 작동이나 통신에 거의 영향을 주지 않는다.

**정답 2번**

**출제 포인트**

분산처리시스템의 장점과 계층형·성형·다중 접근 버스형 연결 구조의 통신 방식 및 장애 특성을 구별하는 문제이다.

**핵심 개념**

분산처리시스템은 여러 위치의 컴퓨터와 자원을 통신망으로 연결하여 자원을 공유하고 작업을 나누어 처리하는 시스템이다.
계층형은 노드가 부모-자식 관계의 트리 구조로 연결되므로 형제 노드끼리 통신할 때 공통 부모 방향의 경로를 거친다. 성형은 모든 노드가 중앙 노드에 연결되며, 버스형은 여러 노드가 하나의 공통 전송 매체를 공유한다.

**선지별 해설**

① O — 분산된 자원을 공동으로 이용할 수 있고, 작업을 여러 노드에 분담하는 분산 처리로 처리 성능을 높일 수 있다.
② X — 틀린 표현은 '부모 노드를 거치지 않고'이다. 계층 연결 구조에서 인접 형제 노드는 서로 직접 연결된 것이 아니므로 통신할 때 부모 노드를 거쳐야 한다. 올바르게는 '인접 형제 노드 간 통신은 부모 노드를 거쳐 이루어진다'이다.
③ O — 성(star)형은 통신이 중앙 노드에 집중되므로 중앙 노드가 병목이 될 수 있고, 중앙 노드가 고장 나면 전체 통신이 중단될 수 있다.
④ O — 다중 접근 버스형에서는 각 노드가 공통 버스에 독립적으로 접속하므로 일반적으로 한 노드의 고장은 다른 노드의 작동이나 통신에 거의 영향을 주지 않는다. 다만 공통 버스 자체가 고장 나면 전체 통신에 영향을 줄 수 있다.

**오답 포인트**

②의 '인접'이라는 말 때문에 형제 노드가 직접 통신한다고 오해하기 쉽다. 계층도에서 가까이 그려져 있어도 형제 노드 사이에는 직접 연결선이 없으며, 실제 통신 경로는 공통 부모를 통과한다.

NoteBase

### 다시 나오면 이렇게 판단

연결 구조의 선지를 판단할 때 노드 사이에 실제 연결선이 있는지를 먼저 본다. 계층형의 형제 노드는 부모를 거치고, 성형은 중앙 노드가 병목이자 단일 장애점이 되며, 버스형은 개별 노드 장애와 공통 버스 장애를 구분한다.

### 암기 포인트

계층형 형제 통신은 부모 경유, 성형은 중앙 집중, 버스형은 노드 고장보다 공통 버스 고장이 치명적이다.

NoteBase

**문 5. 소프트웨어 개발 과정에서 발생하는 리스크에 대한 관리를 가장 잘 대처할 수 있는 개발 접근 모형은?**

① 폭포수 모형
② 점진적 모형
③ 에자일 모형
④ 나선형 모형

**정답 4번**

**출제 포인트**

이 문제는 소프트웨어 개발 생명주기 모형 가운데 위험 분석과 위험 관리를 핵심 활동으로 삼는 모형을 묻는다.

**핵심 개념**

나선형 모형은 계획, 위험 분석, 개발 및 검증, 고객 평가를 반복하면서 소프트웨어를 점진적으로 완성하는 모형이다. 반복 주기마다 위험 요소를 식별·분석하고 대응 방안을 마련한다는 점이 결정적인 특징이다. 폭포수 모형은 단계를 순차적으로 진행하고, 점진적 모형은 기능을 여러 증분으로 나누어 개발하며, 에자일 모형은 짧은 반복과 고객 피드백을 중시한다. 위험 분석을 개발 주기의 중심에 명시적으로 둔 것은 나선형 모형이다.

**선지별 해설**

① X — 폭포수 모형은 요구 분석부터 유지보수까지 단계를 순차적으로 수행하는 방식이다. 앞 단계로 되돌아가기 어렵고 위험 분석을 반복 과정의 핵심으로 삼지 않으므로 이 문항의 답이 아니다. 틀린 선택은 '폭포수 모형'이며, 올바르게는 '나선형 모형'이다.
② X — 점진적 모형은 시스템을 여러 부분으로 나누어 증분별로 개발·인도하는 방식이다. 조기 인도와 단계적 기능 확장이 핵심이므로 위험 관리에 가장 특화된 모형은 아니다. 틀린 선택은 '점진적 모형'이며, 올바르게는 '나선형 모형'이다.
③ X — 에자일 모형은 짧은 개발 주기, 변화 대응, 고객과의 지속적인 협력을 중시한다. 반복을 사용하지만 위험 분석 자체를 각 주기의 중심 활동으로 정립한 모형은 아니다. 틀린 선택은 '에자일 모형'이며, 올바르게는 '나선형 모형'이다.
④ O — 나선형 모형은 반복할 때마다 위험을 분석하고 그 결과에 따라 개발 방향과 대안을 결정하므로 소프트웨어 개발 위험 관리에 가장 적합하다.

**오답 포인트**

③의 에자일 모형도 반복 개발과 빠른 피드백을 통해 위험을 줄일 수 있어 혼동하기 쉽다. 그러나 문제의 핵심 표현이 '리스크에 대한 관리'이면 위험 분석을 독립적이고 핵심적인 단계로 둔 나선형 모형을 선택한다.

NoteBase

**다시 나오면 이렇게 판단**

개발 모형 문제에서는 대표 키워드를 대응시킨다. 순차적 단계는 폭포수, 증분별 인도는 점진적, 짧은 반복과 변화 대응은 에자일, 반복적인 위험 분석은 나선형 모형이다.

**암기 포인트**

나선형 모형의 핵심은 '반복 + 위험 분석'이다.

NoteBase

**문 6. 다음이 설명하는 것은?**

데이터 집합에서 오류를 발견하거나, 정상적인 데이터 패턴에서 갑작스럽게 변하는 데이터를 찾아내는 방법으로써, 신용카드의 사용 패턴이 갑작스럽게 변하는 것을 식별해 낼 수 있음.

① 클래스 차별화
② 클러스터 분석
③ 연계성 분석
④ 이상치 분석

**정답 4번**

**출제 포인트**

정상적인 데이터 패턴에서 벗어난 관측값을 찾아내는 이상치 분석의 개념과 활용 사례를 묻는 문제이다.

**핵심 개념**

이상치 분석은 대부분의 데이터가 보이는 정상 패턴과 크게 다른 값이나 갑작스러운 변화를 탐지하는 기법이다. 신용카드 사용 패턴의 급격한 변화나 비정상 거래 탐지가 대표적인 활용 사례이다.
클래스 차별화는 미리 정해진 클래스들의 특징 차이를 비교하고, 클러스터 분석은 유사한 데이터를 같은 집단으로 묶으며, 연계성 분석은 항목이나 사건이 함께 나타나는 관계를 찾는다.

**선지별 해설**

① X — 클래스 차별화는 이미 구분된 클래스 사이의 차별적 특징을 찾아 비교하는 방법이다. 지문의 '정상 패턴에서 갑작스럽게 변하는 데이터 탐지'는 클래스 차별화가 아니라 이상치 분석이다.
② X — 클러스터 분석은 사전에 정답 집단을 정하지 않고 유사한 데이터끼리 군집을 형성하는 방법이다. 지문의 설명은 유사성에 따른 군집화가 아니라 정상 패턴에서 벗어난 이상치를 찾는 분석이다.
③ X — 연계성 분석은 여러 항목이나 사건 사이에서 함께 발생하는 규칙과 관련성을 발견하는 방법이다. 지문의 설명은 동시 발생 관계를 찾는 연계성 분석이 아니라 비정상 변화를 찾는 이상치 분석이다.
④ O — 이상치 분석은 데이터 오류나 정상 패턴에서 벗어난 급격한 변화를 찾아낸다. 신용카드 사용 양상의 갑작스러운 변화를 식별하는 사례가 이에 해당한다.

**오답 포인트**

'데이터에서 패턴을 찾는다'는 공통점만 보고 클러스터 분석이나 연계성 분석을 고르기 쉽다. 그러나 지문의 결정적 표현은 '정상적인 데이터 패턴에서 갑작스럽게 변하는 데이터'이며, 이는 정상 집단과 다른 예외를 찾는 이상치 분석을 가리킨다.

NoteBase

### 다시 나오면 이렇게 판단

정상 범위에서 벗어남·갑작스러운 변화·오류·부정 사용 탐지가 제시되면 이상치 분석으로 판단한다. 유사한 것끼리 묶으면 클러스터 분석, 함께 나타나는 관계를 찾으면 연계성 분석으로 구분한다.

### 암기 포인트

정상에서 벗어난 예외는 이상치, 비슷한 것끼리 묶으면 클러스터, 함께 나타나면 연계성이다.

NoteBase

**문 7. 인터넷상에서 사용하지 않는 시간대의 연결된 수 많은 컴퓨터를 하나의 고성능 컴퓨터처럼 활용할 수 있는 기술을 무엇이라고 하는가?**

① 그리드 컴퓨팅(Grid Computing)
② 클라이언트 컴퓨팅(Client Computing)
③ 가상 컴퓨팅(Virtual Computing)
④ 유비쿼터스 컴퓨팅(Ubiquitous Computing)

**정답 1번**

**출제 포인트**

여러 컴퓨터의 유휴 자원을 네트워크로 묶어 하나의 고성능 컴퓨팅 자원처럼 활용하는 기술이 그리드 컴퓨팅임을 묻는 문제이다.

**핵심 개념**

그리드 컴퓨팅: 네트워크로 연결된 여러 컴퓨터의 남는 처리 능력과 저장 공간 등을 공동으로 활용하여 대규모 작업을 수행하는 분산 컴퓨팅 방식이다.
클라이언트 컴퓨팅: 사용자의 단말인 클라이언트가 응용 프로그램 실행이나 데이터 처리의 일부 또는 전부를 담당하는 방식이다.
가상 컴퓨팅: 물리적 컴퓨팅 자원을 추상화하여 여러 개의 논리적 컴퓨터나 자원처럼 사용할 수 있게 하는 방식이다.
유비쿼터스 컴퓨팅: 컴퓨터가 생활환경 곳곳에 내재되어 사용자가 언제 어디서나 정보 서비스를 이용할 수 있게 하는 환경이다.

**선지별 해설**

① O — 그리드 컴퓨팅은 인터넷 등 네트워크에 연결된 다수 컴퓨터의 유휴 자원을 결합하여 하나의 고성능 컴퓨터처럼 활용하는 기술이므로 제시된 설명과 일치한다.
② X — 틀린 부분은 '클라이언트 컴퓨팅'이다. 클라이언트 컴퓨팅은 사용자 단말이 응용 프로그램 실행이나 데이터 처리를 담당하는 방식이며, 여러 컴퓨터의 유휴 자원을 통합하는 기술은 그리드 컴퓨팅이다.
③ X — 틀린 부분은 '가상 컴퓨팅'이다. 가상 컴퓨팅은 물리적 자원을 추상화하여 논리적인 가상 머신이나 자원으로 제공하는 방식이며, 다수 컴퓨터의 남는 자원을 모아 공동 처리하는 기술은 그리드 컴퓨팅이다.
④ X — 틀린 부분은 '유비쿼터스 컴퓨팅'이다. 유비쿼터스 컴퓨팅은 컴퓨팅 기능이 생활환경 곳곳에 내재되어 언제 어디서나 서비스를 이용하게 하는 개념이며, 유휴 컴퓨팅 자원의 결합을 가리키는 말은 그리드 컴퓨팅이다.

NoteBase

**오답 포인트**

‘여러 컴퓨터를 하나처럼 사용한다’는 표현 때문에 가상 컴퓨팅과 혼동하기 쉽다. 이 문항의 결정적 단서는 인터넷에 연결된 다수 컴퓨터의 ‘사용하지 않는 시간대’, 즉 유휴 자원을 모아 고성능 연산에 활용한다는 점이다.

**다시 나오면 이렇게 판단**

분산된 여러 컴퓨터의 남는 처리 능력을 결합한다면 그리드 컴퓨팅으로 판단한다. 물리 자원의 추상화는 가상 컴퓨팅, 언제 어디서나 이용 가능한 생활환경은 유비쿼터스 컴퓨팅으로 구별한다.

**암기 포인트**

그리드는 여러 컴퓨터의 유휴 자원을 격자처럼 연결해 하나의 큰 연산 자원으로 활용하는 기술이다.

NoteBase

**문 8. 고급 프로그래밍 언어에 관한 설명 중 옳지 않은 것은?**

① COBOL 언어는 회사의 사무용 자료처리 언어로 개발되었으며, 기계 독립적인 부분과 기계 종속적인 부분을 분리하는 데 성공한 언어이다.

② PASCAL 언어는 간결하면서도 강력한 언어로 손꼽히고 있으며, 교육용 언어로는 뛰어나다는 평가를 받고 있다.

③ C-언어는 고급 언어 프로그래밍과 저급 언어 프로그래밍도 가능한 언어이며, 인터프리터 방식의 대표적 언어이다.

④ FORTRAN은 계산용 언어로서, 뛰어난 실행 효율성으로 성공한 언어이며, 번역기를 구현한 최초의 고급 언어로 평가된다.

**정답 3번**

**출제 포인트**

COBOL, PASCAL, C, FORTRAN의 주된 용도와 번역 방식을 구별하는 문제이다.

**핵심 개념**

COBOL은 기업의 사무·업무 자료 처리를 위해 개발된 고급 언어이다. PASCAL은 구조적 프로그래밍 교육에 적합하도록 설계된 언어이다. C는 고급 언어의 구조와 하드웨어에 가까운 저수준 제어 능력을 함께 제공하며 일반적으로 컴파일러로 번역한다. FORTRAN은 과학·공학의 수치 계산을 위해 개발된 초기 고급 언어이다.

**선지별 해설**

① O — COBOL은 회사의 사무용 자료 처리에 적합하도록 개발되었고, 기계 독립적인 프로그램 부분과 기계 종속적인 환경 기술 부분을 구분하였다.
② O — PASCAL은 문법이 비교적 간결하고 구조적 프로그래밍 개념을 익히기 좋아 교육용 언어로 널리 평가되었다.
③ X — 틀린 표현은 '인터프리터 방식의 대표적 언어'이다. C는 고급 수준의 구조적 프로그래밍과 저수준의 메모리·비트 조작이 모두 가능하지만, 대표적인 컴파일러 방식 언어이다.
④ O — FORTRAN은 과학·공학 계산용으로 개발되었으며 높은 실행 효율을 보였고, 컴파일러를 갖춘 최초의 고급 프로그래밍 언어로 널리 평가된다.

**오답 포인트**

③은 C가 하드웨어에 가까운 저수준 제어도 가능하다는 사실에 '인터프리터 방식'이라는 다른 분류 기준을 섞었다. 저급 언어에 가까운 기능을 제공한다는 것과 인터프리터로 실행된다는 것은 같은 뜻이 아니다.

NoteBase

**다시 나오면 이렇게 판단**

각 언어를 '주된 용도'와 '번역 방식'으로 나누어 판단한다. C는 시스템 프로그래밍과 저수준 제어가 가능하지만 대표적인 컴파일 언어라는 점을 확인한다.

**암기 포인트**

COBOL은 사무 처리, PASCAL은 교육, C는 시스템·컴파일, FORTRAN은 과학 계산이다.

NoteBase

**문 9. 다음 Bit에 대한 설명 중 틀린 것은?**

① 자료 표현의 최소 단위이다.

② 1Bit로 1개의 문자를 표현한다.

③ 2진수 0과 1의 한자리를 의미한다.

④ Binary Digit의 준말이다.

**정답 2번**

**출제 포인트**

비트(Bit)의 정의와 비트가 표현할 수 있는 정보의 범위를 묻는 문제이다.

**핵심 개념**

비트는 컴퓨터에서 자료를 표현하는 최소 단위이며, Binary Digit의 준말이다. 한 비트는 이진수 한 자리로서 0 또는 1 중 하나의 값을 나타낸다. 문자는 문자 인코딩이 정한 여러 비트의 조합으로 표현한다.

**선지별 해설**

① O — 비트는 컴퓨터가 자료를 표현하는 최소 단위이다.
② X — 틀린 부분은 '1Bit로 1개의 문자를 표현한다'이다. 1비트는 0 또는 1 중 하나의 값만 나타내며, 문자는 문자 인코딩이 정한 여러 비트의 조합으로 표현한다.
③ O — 비트는 이진수 0 또는 1을 담는 한 자리를 의미한다.
④ O — Bit는 Binary Digit을 줄인 말이다.

**오답 포인트**

②는 비트와 문자의 저장 단위를 혼동하게 한다. 비트 한 개는 두 가지 상태 중 하나만 나타낼 수 있으므로 일반적인 문자 하나를 표현하기에는 부족하다.

**다시 나오면 이렇게 판단**

비트 관련 선지는 '최소 단위', 'Binary Digit', '0 또는 1의 한 자리'이면 옳다고 판단한다. 문자 표현을 1비트와 바로 연결하면 틀린 설명으로 판단한다.

**암기 포인트**

Bit = Binary Digit = 0 또는 1을 나타내는 정보 표현의 최소 단위이다.

NoteBase

**문 10. 오디오, 비디오, 이미지 등의 디지털 콘텐츠에 사람의 육안으로는 구별할 수 없도록 저작원의 정보를 삽입하여 불법 복제를 막는 기술을 무엇이라고 하는가?**

① 워터마킹(Watermarking)
② 카피라이트(Copyright)
③ 카피레프트(Copyleft)
④ 스패밍(Spamming)

**정답 1번**

**출제 포인트**

디지털 콘텐츠에 식별 정보를 눈에 띄지 않게 삽입하는 워터마킹의 정의를 다른 저작권 관련 개념과 구별하는 문제이다.

**핵심 개념**

워터마킹은 오디오·비디오·이미지 등 디지털 콘텐츠에 저작권자나 출처를 식별할 수 있는 정보를 사람이 쉽게 알아볼 수 없도록 삽입하는 기술이다. 불법 복제물의 출처 확인과 저작권 보호에 활용된다.
카피라이트는 저작자의 권리를 보호하는 법적 권리 체계이고, 카피레프트는 저작물의 자유로운 이용·수정·재배포를 허용하면서 그 자유를 후속 저작물에도 유지하게 하는 배포 원칙이다. 스패밍은 원치 않는 메시지를 대량으로 보내는 행위이다.

**선지별 해설**

① O — 워터마킹은 디지털 콘텐츠에 육안으로 구별하기 어려운 저작권자·출처 정보를 삽입하여 저작권 보호와 불법 복제 추적에 활용하는 기술이므로 제시된 설명과 일치한다.
② X — 틀린 표현은 '카피라이트'이다. 카피라이트는 저작물에 대한 저작자의 법적 권리를 뜻하며, 정보를 콘텐츠에 삽입하는 기술이 아니다. 올바르게는 '워터마킹'이다.
③ X — 틀린 표현은 '카피레프트'이다. 카피레프트는 일정한 조건 아래 저작물의 이용·수정·재배포를 허용하고 같은 자유를 유지하게 하는 배포 원칙이지, 보이지 않는 정보를 삽입하는 기술이 아니다. 올바르게는 '워터마킹'이다.
④ X — 틀린 표현은 '스패밍'이다. 스패밍은 수신자가 원하지 않는 광고성 메시지 등을 불특정 다수에게 대량 전송하는 행위이다. 올바르게는 '워터마킹'이다.

**오답 포인트**

'저작권 보호'라는 공통 연상어만 보고 카피라이트를 고르기 쉽다. 그러나 선지의 결정적 표현은 '디지털 콘텐츠 내부에 육안으로 구별하기 어려운 정보를 삽입하는 기술'이며, 이는 권리 제도가 아니라 워터마킹을 가리킨다.

NoteBase

**다시 나오면 이렇게 판단**

콘텐츠 안에 소유자·출처 정보를 숨겨 넣는 기술이면 워터마킹, 저작자의 법적 권리이면 카피라이트, 자유로운 이용과 재배포 원칙이면 카피레프트, 원치 않는 메시지의 대량 전송이면 스패밍으로 구분한다.

**암기 포인트**

워터마킹은 콘텐츠 속 숨은 소유 표식이다.

NoteBase

**문 11. SRAM과 DRAM을 설명한 것으로 옳은 것은?**

① SRAM은 재충전이 필요 없는 메모리다.

② DRAM이 SRAM에 비해 속도가 빠르다.

③ SRAM의 소비전력이 DRAM 보다 낮다.

④ DRAM의 Memory Cell은 Flip-Flop으로 구성되어 있다.

**정답 1번**

**출제 포인트**

SRAM과 DRAM의 기억 소자, 재충전 필요 여부, 속도와 소비전력의 차이를 묻는 문제이다.

**핵심 개념**

SRAM은 플립플롭 회로에 데이터를 저장하는 정적 RAM으로, 전원이 공급되는 동안 주기적인 재충전이 필요 없다. DRAM은 트랜지스터와 커패시터로 구성된 셀에 전하를 저장하는 동적 RAM으로, 전하가 누설되므로 주기적인 재충전이 필요하다. 일반적으로 SRAM은 DRAM보다 빠르지만 집적도가 낮고 가격과 소비전력이 높다.

**선지별 해설**

① O — SRAM은 전원이 공급되는 동안 저장 내용을 유지하기 위한 주기적인 재충전이 필요 없는 메모리이다.
② X — 틀린 표현은 'DRAM이 SRAM에 비해 속도가 빠르다'이다. 실제로는 플립플롭 방식의 SRAM이 커패시터 방식의 DRAM보다 빠르다.
③ X — 틀린 표현은 'SRAM의 소비전력이 DRAM보다 낮다'이다. 일반적인 특성 비교에서는 SRAM의 소비전력이 DRAM보다 높고, DRAM의 소비전력이 더 낮다.
④ X — 틀린 표현은 'DRAM의 Memory Cell은 Flip-Flop으로 구성되어 있다'이다. 실제로 DRAM의 메모리 셀은 일반적으로 트랜지스터와 커패시터로 구성되고, 플립플롭은 SRAM의 기억 소자이다.

**오답 포인트**

'재충전이 필요 없다'는 말이 비휘발성을 뜻하는 것은 아니다. SRAM과 DRAM은 모두 전원이 끊기면 내용이 사라지는 휘발성 메모리이지만, SRAM만 전원 공급 중 주기적인 재충전이 필요 없다. 또한 플립플롭과 커패시터의 대응을 서로 바꾸어 제시한 ④에 주의해야 한다.

**다시 나오면 이렇게 판단**

S는 Static으로 연결하여 플립플롭·재충전 불필요·빠른 속도를, D는 Dynamic으로 연결하여 커패시터·주기적 재충전·상대적으로 느린 속도를 기준으로 판단한다.

NoteBase

**암기 포인트**

SRAM은 플립플롭·무재충전·고속, DRAM은 커패시터·재충전·저속이다.

NoteBase

**문 12. 소프트웨어 설계과정에서 고려하는 모듈화에 대한 설명으로 적합한 것은?**

① 모듈화에 대한 척도는 일반적으로 결합력과 응집력으로 나타낸다.

② 모듈화를 달성하기 위해서는 결합력과 응집력은 모두 높아야 한다.

③ 전역변수를 사용한 모듈화는 매개변수 전달 방법보다 우수하다.

④ 제어용 플래그 매개변수의 사용은 좋은 모듈화 달성 방법이다.

**정답 1번**

**출제 포인트**

모듈화의 품질을 판단하는 결합력과 응집력의 바람직한 방향을 묻는 문제이다.

**핵심 개념**

결합력은 서로 다른 모듈이 의존하는 정도이며 낮을수록 변경의 영향이 작아진다. 응집력은 한 모듈 내부의 구성 요소가 하나의 기능을 위해 밀접하게 관련된 정도이며 높을수록 좋다. 따라서 좋은 모듈화의 기본 원칙은 낮은 결합력과 높은 응집력이다.

**선지별 해설**

① O — 모듈화의 정도와 품질은 일반적으로 모듈 간 결합력과 모듈 내부 응집력으로 평가하므로 적합한 설명이다.
② X — 틀린 표현은 '결합력과 응집력은 모두 높아야 한다'이다. 올바르게는 결합력은 낮고 응집력은 높아야 한다.
③ X — 틀린 표현은 '전역변수를 사용한 모듈화는 매개변수 전달 방법보다 우수하다'이다. 전역변수를 여러 모듈이 공유하면 공통 결합이 생겨 의존성과 변경 영향이 커진다. 올바르게는 필요한 데이터만 매개변수로 전달하는 방법이 전역변수 공유보다 결합력을 낮추는 데 유리하다.
④ X — 틀린 표현은 '제어용 플래그 매개변수의 사용은 좋은 모듈화 달성 방법이다'이다. 제어 플래그를 넘겨 다른 모듈의 처리 흐름을 지시하면 제어 결합이 생긴다. 올바르게는 제어 정보보다 필요한 데이터만 전달하고, 모듈이 자신의 기능과 흐름을 스스로 결정하도록 설계하는 것이 바람직하다.

**오답 포인트**

②는 응집력의 방향을 결합력에도 그대로 적용한 함정이다. ③의 전역변수와 ④의 제어용 플래그는 전달이 편리해 보여도 모듈 사이의 의존성을 높이므로 좋은 모듈화와 거리가 멀다.

**다시 나오면 이렇게 판단**

모듈 내부의 관련성은 높이고 모듈 사이의 의존성은 낮추는지를 판단한다. 전역 데이터 공유나 제어 정보 전달이 보이면 결합력이 커지는 설계인지 확인한다.

NoteBase

**암기 포인트**

좋은 모듈화는 낮은 결합력, 높은 응집력이다.

NoteBase

**문 13. 다중화 방식에 대한 설명으로 옳지 않은 것은?**

① FDMA : 전송하려는 신호보다 대역폭이 큰 매체를 사용한다.

② TDMA : 한 개의 고속 전송선로를 일정한 시간폭으로 나누어 전송하는 기술이다.

③ ATDMA : 전송할 정보가 존재하는 단말기에만 타임슬롯을 할당한다.

④ CDMA : FDMA와 TDMA의 혼합 방식으로 정보의 압축, 에러 복구가 어렵다.

**정답 4번**

**출제 포인트**

다중화 방식인 FDMA, TDMA, ATDMA, CDMA가 전송 자원을 구분하고 할당하는 기준을 묻는 문제이다.

**핵심 개념**

FDMA는 주파수 대역을 여러 채널로 나누어 사용자별로 할당하는 방식이다.
TDMA는 하나의 전송로를 시간 슬롯으로 나누어 사용자별로 할당하는 방식이다.
ATDMA는 전송할 데이터가 있는 단말기에만 시간 슬롯을 동적으로 할당하는 방식이다.
CDMA는 사용자마다 서로 다른 확산 코드를 부여하여 같은 주파수 대역과 같은 시간에 전송된 신호를 코드로 구분하는 방식이다.

**선지별 해설**

① O — FDMA는 각 신호가 차지하는 대역보다 넓은 매체의 전체 대역폭을 여러 주파수 채널로 분할하여 동시에 전송한다.
② O — TDMA는 하나의 고속 전송선로를 일정한 시간 슬롯으로 나누고 각 단말이 배정된 시간에 전송하도록 한다.
③ O — ATDMA는 전송 요구가 있는 단말기에만 시간 슬롯을 동적으로 할당하므로 고정식 TDMA보다 빈 슬롯의 낭비를 줄일 수 있다.
④ X — 틀린 표현은 ‘FDMA와 TDMA의 혼합 방식’이다. CDMA는 두 방식의 혼합이 아니라 사용자별 고유 확산 코드로 신호를 구분하는 방식이다. 따라서 올바르게는 ‘CDMA는 같은 주파수 대역과 같은 시간에 여러 사용자가 전송하고, 수신 측이 고유 코드로 각 신호를 분리하는 방식이다’라고 해야 한다. 정보 압축과 에러 복구의 어려움도 CDMA를 정의하는 특징이 아니다.

**오답 포인트**

④는 CDMA도 주파수와 시간을 사용한다는 점을 근거로 FDMA와 TDMA의 혼합이라고 착각하게 만든다.
CDMA의 사용자 구분 기준은 주파수나 시간이 아니라 고유 코드이다.

NoteBase

**다시 나오면 이렇게 판단**

사용자를 무엇으로 나누는지 확인한다. 주파수로 나누면 FDMA, 시간으로 나누면 TDMA, 전송 수요에 따라 시간 슬롯을 동적으로 배정하면 ATDMA, 고유 코드로 나누면 CDMA이다.

**암기 포인트**

F는 주파수, T는 시간, AT는 필요한 단말에 시간 슬롯, C는 코드로 구분한다.

NoteBase

**문 14. 빅 데이터의 특성을 설명하는 것이 아닌 것은?**

① Verify (검사)

② Volume (규모)

③ Velocity (속도)

④ Variety (다양성)

**정답 1번**

**출제 포인트**

빅 데이터의 대표적 특성인 3V와 철자가 비슷한 비표준 용어를 구별하는 문제이다.

**핵심 개념**

빅 데이터는 기존 방식으로 처리하기 어려울 정도로 데이터의 규모가 크고, 빠르게 생성·처리되며, 형태가 다양한 데이터를 말한다. 대표적인 3V는 Volume(규모), Velocity(속도), Variety(다양성)이다. 확장된 특성으로 신뢰성·진실성을 뜻하는 Veracity를 더하기도 하지만, Verify(검사)는 3V에 포함되지 않는다.

**선지별 해설**

① X — Verify(검사)는 빅 데이터의 대표적 3V가 아니다. 틀린 표현은 Verify이며, 3V는 Volume, Velocity, Variety이다. 확장 특성을 묻는 경우에는 Verify가 아니라 Veracity(신뢰성·진실성)가 사용된다.
② O — Volume은 수집·저장·처리해야 하는 데이터의 방대한 규모를 뜻한다.
③ O — Velocity는 데이터가 빠르게 생성·유입되고 신속하게 처리되어야 하는 특성을 뜻한다.
④ O — Variety는 정형 데이터뿐 아니라 문서, 이미지, 영상 등 다양한 형태의 데이터가 존재하는 특성을 뜻한다.

**오답 포인트**

①의 Verify는 V로 시작하고 데이터 검증과 관련 있어 보이지만, 빅 데이터의 대표 특성 용어가 아니다. 특히 확장된 V인 Veracity와 혼동하기 쉽다.

**다시 나오면 이렇게 판단**

빅 데이터의 기본 특성을 물으면 먼저 3V인 Volume·Velocity·Variety에 해당하는지 확인한다. 추가 V가 제시되면 Verify가 아니라 Veracity 등 실제 확장 개념인지 구별한다.

**암기 포인트**

빅 데이터 3V는 규모(Volume)·속도(Velocity)·다양성(Variety)이다. Verify가 아니라 Veracity이다.

NoteBase

**문 15. 버퍼링(buffering)과 스풀링(spooling)에 대한 설명으로 옳지 않은 것은?**

① 스풀링은 디스크 일부를 매우 큰 버퍼처럼 사용한다.

② 버퍼링은 한 레코드를 읽어서 CPU가 그것에 대한 작업을 시작함과 동시에 입출력 장치가 필요한 레코드를 미리 읽어 버퍼에 저장해 둔다.

③ 스풀링은 어떤 작업의 입출력과 다른 작업의 계산을 병행 처리한다.

④ 버퍼링은 보조기억장치를 버퍼로 사용한다.

**정답 4번**

**출제 포인트**

버퍼링과 스풀링이 사용하는 저장 공간과 병행 처리의 대상을 구별하는 문제이다.

**핵심 개념**

버퍼링은 주기억장치의 버퍼에 데이터를 임시로 저장하여 CPU와 입출력 장치의 속도 차이를 완화하는 기법이다.
스풀링은 디스크의 일부를 큰 버퍼처럼 사용하여 한 작업의 입출력과 다른 작업의 계산을 겹쳐 처리하는 기법이다.
핵심 차이는 버퍼링은 주기억장치와 한 작업 내부의 입출력·처리 중첩, 스풀링은 보조기억장치와 작업 간 중첩이 중심이라는 점이다.

**선지별 해설**

① O — 스풀링은 디스크 일부를 대용량 임시 저장 공간으로 사용하므로 옳다.
② O — CPU가 현재 레코드를 처리하는 동안 입출력 장치가 다음 레코드를 미리 버퍼에 읽어 두면 두 장치의 동작을 겹칠 수 있으므로 옳다.
③ O — 스풀링은 한 작업의 입출력 데이터를 디스크에 모아 두고, 그 입출력과 다른 작업의 계산을 병행할 수 있게 하므로 옳다.
④ X — 틀린 표현은 '보조기억장치를 버퍼로 사용한다'이다. 올바르게는 버퍼링은 주기억장치의 버퍼를 사용하고, 보조기억장치인 디스크를 큰 버퍼처럼 사용하는 것은 스풀링이다.

**오답 포인트**

④는 '임시 저장 공간을 사용한다'는 공통점만 이용하여 스풀링의 저장 매체인 보조기억장치를 버퍼링의 특징으로 바꿔 놓았다.

NoteBase

### 다시 나오면 이렇게 판단

저장 공간이 주기억장치이고 한 작업의 처리 흐름을 완충하면 버퍼링, 디스크를 이용하여 서로 다른 작업의 입출력과 계산을 겹치면 스풀링으로 판단한다.

### 암기 포인트

버퍼링은 주기억장치·작업 내부, 스풀링은 디스크·작업 사이이다.

NoteBase

**문 16. 레지스터와 그 사용 목적이 올바르지 않은 것은?**

① ACC - 연산 결과를 임시적으로 저장하는 레지스터

② MSR - 주소 변환에 필요한 수를 저장하는 레지스터

③ MBR - 주기억 장치에서 읽은 정보를 저장하는 레지스터

④ MAR - 접근하는 메모리의 주소 값을 저장하는 레지스터

**정답 2번**

**출제 포인트**

CPU의 주요 레지스터가 주소, 데이터, 연산 결과, 상태 정보 중 무엇을 저장하는지 구별하는 문제이다.

**핵심 개념**

ACC는 산술·논리 연산의 피연산자나 중간 결과 및 최종 결과를 임시로 저장하는 누산기이다.
MBR은 주기억 장치에서 읽어 온 데이터 또는 주기억 장치에 기록할 데이터를 임시로 저장한다.
MAR은 CPU가 읽거나 쓰려는 주기억 장치의 주소를 저장한다.
MSR은 시스템이나 메모리의 상태 정보를 나타내는 상태 레지스터로 쓰이며, 주소 변환용 수를 저장하는 레지스터가 아니다.

**선지별 해설**

① O — ACC는 연산 과정에서 피연산자와 연산 결과를 임시로 저장하는 누산기이므로 설명이 옳다.
② X — 틀린 표현은 ‘MSR - 주소 변환에 필요한 수를 저장’이다. MSR은 상태 정보를 저장하는 레지스터이고, 주소 변환에 더하는 기준값이나 변위값은 베이스 레지스터·인덱스 레지스터 등 주소 계산용 레지스터에 저장된다.
③ O — MBR은 주기억 장치에서 읽어 온 정보를 임시로 저장하며, 반대로 주기억 장치에 쓸 정보도 임시로 보관한다.
④ O — MAR은 현재 CPU가 접근하려는 주기억 장치의 주소를 저장하므로 설명이 옳다.

**오답 포인트**

②의 ‘주소 변환’이라는 말 때문에 주소와 관련된 레지스터처럼 보이지만, MSR의 핵심은 상태 정보이다. 주소 그 자체는 MAR에, 유효주소 계산에 필요한 값은 베이스 레지스터나 인덱스 레지스터 등에 저장된다는 차이가 함정이다.

**다시 나오면 이렇게 판단**

레지스터의 약어를 먼저 풀어 기능과 연결한다. ACC는 연산 결과, MBR은 메모리와 주고받는 데이터, MAR은 접근할 메모리 주소, 상태 레지스터는 시스템 상태로 구분한다.

NoteBase

**암기 포인트**

ACC는 연산, MBR은 데이터, MAR은 주소, MSR은 상태이다.

NoteBase

**문 17. IP(Internet Protocol)의 주요임무가 아닌 것은?**

① 호스트의 주소지정

② 패킷 전달

③ 전송 경로의 논리적 관리

④ 전송 패킷의 흐름제어

**정답 4번**

**출제 포인트**

이 문제는 네트워크 계층의 IP가 담당하는 주소 지정·패킷 전달·경로 관리와 전송 계층의 TCP가 담당하는 흐름제어를 구분하는 문제이다.

**핵심 개념**

IP는 출발지에서 목적지까지 데이터그램을 전달하기 위한 네트워크 계층 프로토콜이다. 논리적 IP 주소를 사용하고, 목적지 주소에 따라 패킷이 거쳐 갈 경로를 정해 전달한다. 반면 TCP의 흐름제어는 수신 측의 처리 능력에 맞추어 송신량을 조절하는 전송 계층 기능이다.

**선지별 해설**

① O — 호스트를 식별하기 위한 논리적 IP 주소를 사용하는 주소 지정은 IP의 주요 기능이다.
② O — IP는 목적지 IP 주소를 바탕으로 라우터를 거쳐 패킷을 목적지 네트워크로 전달한다.
③ O — IP는 라우팅을 통해 패킷이 이동할 논리적 전송 경로를 선택·관리한다.
④ X — 틀린 표현은 '전송 패킷의 흐름제어'이다. IP는 비연결형으로 패킷 전달을 담당하며 송신량을 수신 측 처리 능력에 맞게 조절하지 않는다. 올바르게는 종단 간 흐름제어는 TCP가 수신 윈도우 등을 이용하여 수행한다.

**오답 포인트**

IP도 패킷을 전달하므로 패킷과 관련된 모든 제어를 담당한다고 혼동하기 쉽다. 그러나 ④의 '흐름제어'는 단순한 패킷 전달이나 라우팅이 아니라 송신 속도·송신량을 수신 측에 맞추는 기능이므로 IP의 임무가 아니다.

**다시 나오면 이렇게 판단**

주소 지정·라우팅·패킷 전달이면 IP를, 신뢰성 있는 전송·순서 제어·오류 제어·종단 간 흐름제어이면 TCP를 먼저 대응한다.

**암기 포인트**

IP는 주소와 길 찾기, TCP는 신뢰성과 흐름 조절이다.

NoteBase

**문 18. 운영체제에서 교착상태(deadlock)가 발생하기 위한 필요 조건에 해당되지 않는 것은?**

① 상호배제(mutual exclusion)

② 점유와 대기(hold and wait)

③ 선점(preemption)

④ 순환대기(circular wait)

**정답 3번**

**출제 포인트**

교착상태가 발생하기 위한 네 가지 필요조건을 구별하는 문제이다. 네 조건이 모두 동시에 성립해야 교착상태가 발생할 수 있다.

**핵심 개념**

상호배제: 한 번에 하나의 프로세스만 자원을 사용할 수 있는 상태이다.
점유와 대기: 프로세스가 이미 자원을 가진 채 다른 자원을 기다리는 상태이다.
비선점: 다른 프로세스가 사용 중인 자원을 강제로 빼앗을 수 없는 상태이다.
순환대기: 여러 프로세스가 원형으로 서로가 가진 자원을 기다리는 상태이다.

**선지별 해설**

① O — 상호배제는 교착상태의 필요조건이다. 공유할 수 없는 자원을 한 프로세스가 사용하면 다른 프로세스는 그 자원이 해제될 때까지 기다려야 한다.
② O — 점유와 대기는 교착상태의 필요조건이다. 프로세스가 하나 이상의 자원을 점유한 상태에서 추가 자원을 기다릴 때 자원 대기가 얽힐 수 있다.
③ X — 틀린 표현은 '선점'이다. 교착상태의 필요조건은 자원을 강제로 회수할 수 없는 '비선점'이다. 선점이 가능하면 운영체제가 자원을 회수하여 대기 고리를 해소할 수 있으므로 선점 자체는 교착상태의 필요조건이 아니다.
④ O — 순환대기는 교착상태의 필요조건이다. 프로세스들이 원형으로 다음 프로세스가 점유한 자원을 기다리면 어느 프로세스도 진행하지 못할 수 있다.

**오답 포인트**

③의 '선점'을 익숙한 운영체제 용어라는 이유로 필요조건으로 고르기 쉽다. 교착상태를 만드는 조건은 선점이 가능한 상태가 아니라 선점이 불가능한 '비선점' 상태이다.

NoteBase

### 다시 나오면 이렇게 판단

교착상태 필요조건은 상호배제, 점유와 대기, 비선점, 순환대기의 네 가지로 확인한다. 선지에 '선점'이 제시되면 '비선점'과 반대되는 표현인지 먼저 판별한다.

### 암기 포인트

교착상태의 네 조건은 '상호배제·점유와 대기·비선점·순환대기'이며, 선점이 아니라 비선점이다.

NoteBase

**문 19. 다음에 설명하는 방법에 해당되는 것은?**

일반적으로 30일간의 주기를 기반으로 실제 동작하는 제품을 반복적으로 개발하면서 프로젝트를 진행하며, 매일 수행하는 프로젝트 미팅을 통하여 개발 상황을 공유하는 방법

① 프로토타입(Prototype)
② 스크럼(Scrum)
③ 스파이크(Spike)
④ 스테이지 게이트(Stage-Gate)

**정답 2번**

**출제 포인트**

이 문제는 짧고 고정된 개발 주기를 반복하고 매일 진행 상황을 공유하는 애자일 방법론이 스크럼임을 식별하는 문제이다.

**핵심 개념**

스크럼은 스프린트라는 짧고 고정된 주기마다 실제로 동작하는 제품 증분을 만들고, 데일리 스크럼을 통해 진행 상황과 장애 요인을 공유하는 애자일 프레임워크이다.
프로토타입은 요구사항이나 설계의 타당성을 확인하려고 미리 만드는 시험용 모형이고, 스파이크는 불확실성을 줄이기 위한 조사·실험 작업이며, 스테이지 게이트는 개발 과정을 여러 단계와 심사 관문으로 나누어 다음 단계 진행 여부를 결정하는 방식이다.

**선지별 해설**

① X — 프로토타입은 아이디어나 요구사항을 검증하기 위한 시험용 모형이다. 설명의 '30일간의 반복 주기'와 '매일 수행하는 미팅'에 해당하는 용어는 프로토타입이 아니라 스크럼이다.
② O — 스크럼은 보통 한 달 이내의 스프린트를 반복하여 동작 가능한 제품 증분을 만들고, 데일리 스크럼에서 개발 상황을 공유한다. 지문의 30일 주기와 매일 미팅이라는 단서가 모두 일치한다.
③ X — 스파이크는 기술적·요구사항상의 불확실성을 해소하기 위해 일정 시간을 정해 조사하거나 실험하는 작업이다. 실제 제품을 반복 개발하면서 매일 미팅하는 전체 방법은 스파이크가 아니라 스크럼이다.
④ X — 스테이지 게이트는 개발을 단계별로 구분하고 각 게이트에서 계속 진행할지를 심사하는 방식이다. 30일 단위 반복 개발과 매일 진행 상황 공유를 중심으로 하는 방식은 스테이지 게이트가 아니라 스크럼이다.

**오답 포인트**

'실제 동작하는 제품을 반복적으로 개발한다'는 표현만 보고 프로토타입을 고르기 쉽다. 그러나 '30일간의 주기'는 스프린트, '매일 수행하는 프로젝트 미팅'은 데일리 스크럼을 가리키므로 두 단서를 함께 보아야 한다.

NoteBase

**다시 나오면 이렇게 판단**

짧은 고정 주기의 반복 개발, 동작 가능한 제품 증분, 매일 진행하는 짧은 공유 회의가 함께 제시되면 스크럼으로 판단한다. 단계별 심사 관문이면 스테이지 게이트, 시험용 모형이면 프로토타입, 불확실성 해소용 조사·실험이면 스파이크이다.

**암기 포인트**

스크럼은 스프린트와 데일리 스크럼, 프로토타입은 시험용 모형, 스파이크는 불확실성 해소, 스테이지 게이트는 단계별 심사이다.

NoteBase

**문 20. 데이터베이스 시스템에서 데이터 독립성에 대한 설명으로 옳은 것은?**

① 논리적 데이터 독립성은 개념 스키마와 내부 스키마 사이의 사상(mapping) 관계에 의하여 제공한다.

② 물리적 데이터 독립성이 제공 될 때, 외부 스키마는 내부 스키마의 변경에 영향을 받는다.

③ 논리적 데이터 독립성은 외부 스키마에 영향을 주지 않고 데이터베이스의 논리적 구조를 변경 시킬 수 있는 능력을 말한다.

④ 물리적 데이터 독립성은 내부 스키마를 변경하지 않으면서 개념 스키마를 변경할 수 있는 능력을 말한다.

**정답 3번**

**출제 포인트**

이 문제는 3단계 스키마 구조에서 논리적 데이터 독립성과 물리적 데이터 독립성이 각각 어느 스키마 사이의 사상으로 보장되는지를 묻는다.

**핵심 개념**

논리적 데이터 독립성: 외부 스키마와 응용 프로그램에 영향을 주지 않고 개념 스키마를 변경할 수 있는 능력이다. 외부 스키마와 개념 스키마 사이의 사상이 이를 지원한다.
물리적 데이터 독립성: 개념 스키마에 영향을 주지 않고 저장 구조인 내부 스키마를 변경할 수 있는 능력이다. 개념 스키마와 내부 스키마 사이의 사상이 이를 지원한다.

**선지별 해설**

① X — 틀린 표현은 '논리적 데이터 독립성'이다. 개념 스키마와 내부 스키마 사이의 사상은 물리적 데이터 독립성을 제공한다. 논리적 데이터 독립성은 외부 스키마와 개념 스키마 사이의 사상으로 제공된다.
② X — 틀린 표현은 '외부 스키마는 내부 스키마의 변경에 영향을 받는다'이다. 물리적 데이터 독립성이 보장되면 내부 스키마가 변경되어도 개념 스키마가 영향을 받지 않으며, 그 위의 외부 스키마도 변경할 필요가 없다.
③ O — 논리적 데이터 독립성은 외부 스키마에 영향을 주지 않고 데이터베이스의 논리적 구조인 개념 스키마를 변경할 수 있는 능력이다.
④ X — 내부 스키마와 개념 스키마의 변경 방향을 거꾸로 설명했다. 물리적 데이터 독립성은 개념 스키마를 변경하지 않으면서 내부 스키마를 변경할 수 있는 능력이다.

**오답 포인트**

①과 ④는 '논리적·물리적'이라는 명칭과 변경 대상·보호 대상을 서로 바꾼 선지이다. 특히 물리적 데이터 독립성은 물리적 저장 구조인 내부 스키마를 변경하되 논리적 구조인 개념 스키마를 보호한다는 점이 핵심이다.

NoteBase

**다시 나오면 이렇게 판단**

개념 스키마를 바꾸고 외부 스키마를 보호하면 논리적 데이터 독립성이고, 내부 스키마를 바꾸고 개념 스키마를 보호하면 물리적 데이터 독립성으로 판단한다.

**암기 포인트**

논리적 독립성은 외부-개념 사상, 물리적 독립성은 개념-내부 사상이다.

NoteBase