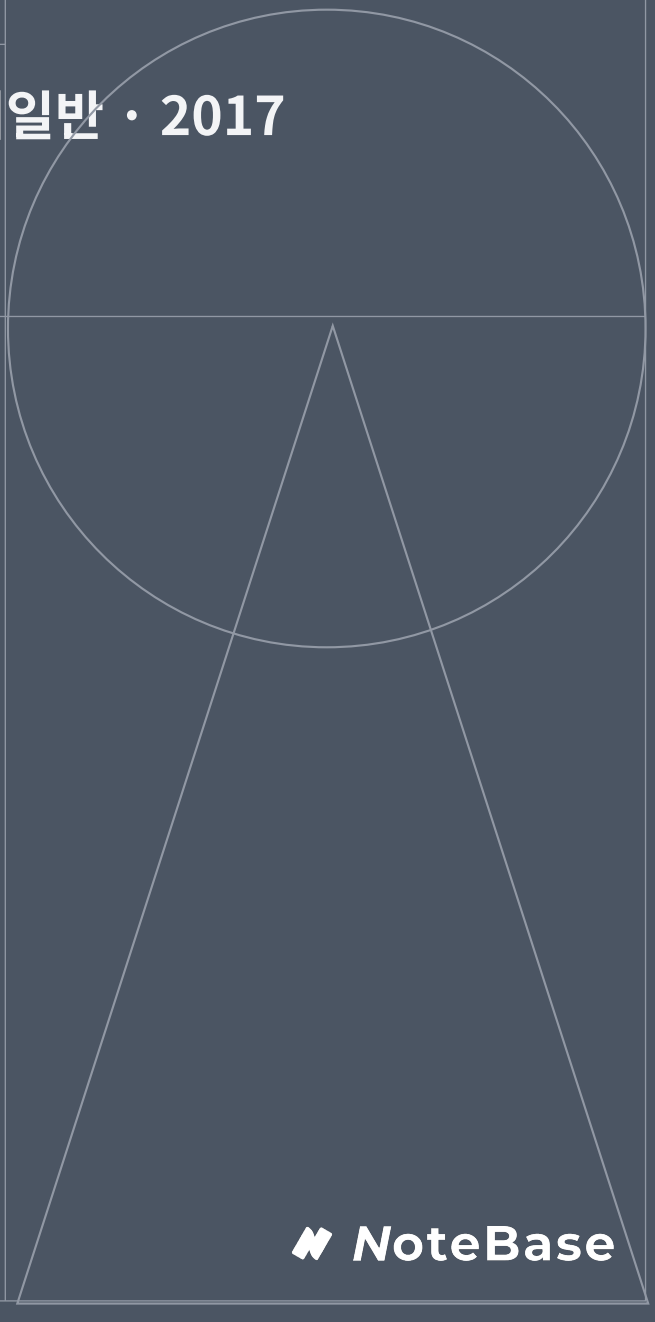
기출OX 해설집

공무원/국가직/9급 · 컴퓨터일반 · 2017



NoteBase

기출OX 해설집

공무원/국가직/9급 · 컴퓨터일반 · 2017

공무원/국가직/9급 · 컴퓨터일반 · 2017

문 1. 컴퓨터 구조에 대한 설명으로 옳지 않은 것은?

① 폰노이만이 제안한 프로그램 내장방식은 프로그램 코드와 데이터를 내부기억장치에 저장하는 방식이다.

② 병렬처리방식 중 하나인 SIMD는 하나의 명령어를 처리하기 위해 다수의 처리장치가 동시에 동작하는 다중처리기 방식이다.

③ CISC 구조는 RISC 구조에 비해 명령어의 종류가 적고 고정 명령어 형식을 취한다.

④ 파이프라인 기법은 하나의 작업을 다수의 단계로 분할하여 시간적으로 중첩되게 실행함으로써 처리율을 높인다.

정답 3번

간단한 해설

③의 '명령어의 종류가 적고 고정 명령어 형식을 취한다'는 **RISC의 전형적인 특징**이다. CISC는 일반적으로 명령어의 종류가 많고 **가변 길이 명령어 형식**을 사용한다.

선지별 해설

① O: 프로그램 내장방식은 **프로그램 명령어와 데이터를 기억장치에 함께 저장**하고, CPU가 저장된 명령어를 가져와 실행하는 방식이다. 폰 노이만 구조에서는 명령어와 데이터가 같은 주기억장치에 저장된다.

② O: SIMD는 단일 명령어·다중 데이터 방식이다. 여러 처리장치가 **같은 명령어를 서로 다른 데이터에 동시에 적용**한다. 예를 들어 여러 수에 각각 같은 값을 더하는 연산을 병렬로 수행할 수 있다.

③ X: CISC는 복잡한 기능을 수행하는 다양한 명령어를 제공하며, 일반적으로 **명령어의 종류가 많고 길이가 가변적**이다. 반면 RISC는 명령어를 단순화하고 대체로 고정 길이 형식을 사용한다. 따라서 선지의 '명령어의 종류가 적고 고정 명령어 형식을 취한다'는 RISC의 전형적인 특징으로 고쳐야 한다.

④ O: 파이프라인은 작업을 여러 단계로 나누고, 서로 다른 작업이 각 단계를 **시간적으로 중첩하여 실행**하게 한다. 예를 들어 첫 번째 명령어를 해독하는 동안 다음 명령어를 인출할 수 있다. 이렇게 단계별 작업을 겹쳐 수행하면 단위 시간에 완료하는 작업 수인 처리율이 높아진다.

NoteBase

문 2. 중앙처리장치 내의 레지스터 중 PC(program counter), IR(instruction register), MAR(memory address register), AC(accumulator)와 다음 설명이 옳게 짝지어진 것은?

ㄱ. 명령어 실행 시 필요한 데이터를 일시적으로 보관한다.
ㄴ. CPU가 메모리에 접근하기 위해 참조하려는 명령어의 주소 혹은 데이터의 주소를 보관한다.
ㄷ. 다음에 인출할 명령어의 주소를 보관한다.
ㄹ. 가장 최근에 인출한 명령어를 보관한다.

① PC: ㄱ, IR: ㄴ, MAR: ㄷ, AC: ㄹ
② PC: ㄴ, IR: ㄹ, MAR: ㄷ, AC: ㄱ
③ PC: ㄷ, IR: ㄴ, MAR: ㄱ, AC: ㄹ
④ PC: ㄷ, IR: ㄹ, MAR: ㄴ, AC: ㄱ

정답 4번

간단한 해설

PC는 **다음에 인출할 명령어의 주소**를 저장하고, IR은 **이미 인출한 명령어 자체**를 저장한다. 주소를 보관하는 것과 그 주소에서 가져온 명령어를 보관하는 것은 서로 다른 역할이다.

대응 관계 해설

PC(program counter, 프로그램 카운터) → ㄷ: **다음에 인출할 명령어의 주소**를 저장한다. CPU는 이 주소를 바탕으로 다음 명령어를 메모리에서 가져온다. 명령어 자체를 저장하는 IR과 구별된다.

IR(instruction register, 명령어 레지스터) → ㄹ: **가장 최근에 인출한 명령어 자체**를 저장한다. 제어장치는 이 명령어를 해독하여 수행할 동작을 결정한다. 따라서 명령어의 주소가 아니라 명령어의 내용을 보관한다.

MAR(memory address register, 메모리 주소 레지스터) → ㄴ: CPU가 메모리에 접근할 때 **접근 대상의 주소**를 저장한다. 그 대상은 명령어일 수도 있고 데이터일 수도 있다. 다음 명령어의 주소를 관리하는 PC보다 메모리 접근 전반에 해당하는 역할이다.

AC(accumulator, 누산기) → ㄱ: 명령어 실행에 필요한 데이터나 연산의 중간 결과를 일시적으로 저장한다. 주소를 보관하는 PC·MAR과 달리 **연산에 사용하는 값**을 담는다.

PC, IR, MAR, AC의 대응은 각각 ㄷ, ㄹ, ㄴ, ㄱ이므로 ④의 짝짓기와 일치한다.

NoteBase

문 3. 트랜잭션이 정상적으로 완료(commit)되거나, 중단(abort)되었을 때 롤백(rollback)되어야 하는 트랜잭션의 성질은?

① 원자성(atomicity)
② 일관성(consistency)
③ 격리성(isolation)
④ 영속성(durability)

정답 1번

간단한 해설

트랜잭션은 일부 작업만 반영된 채 끝나서는 안 된다. 모든 작업을 완료해 확정하거나, 중단되면 수행한 변경을 모두 취소하는 성질이 **원자성**이다.

공통 해설

원자성(atomicity)은 트랜잭션을 하나의 작업 단위로 취급하여, 그 안의 연산이 **모두 반영되거나 전혀 반영되지 않도록** 보장하는 성질이다. 정상적으로 완료되면 커밋(commit)하여 변경을 확정하고, 중단(abort)되면 롤백(rollback)하여 그 트랜잭션이 수행한 변경을 취소한다. 따라서 문제에서 제시한 완료 또는 중단 시의 처리 원칙은 원자성에 해당한다.

예를 들어 계좌 이체를 하나의 트랜잭션으로 처리할 때, A 계좌에서 돈을 차감한 뒤 B 계좌에 입금하기 전에 중단되면 A 계좌의 차감도 취소해야 한다. 차감만 반영된 상태를 남기지 않고 두 작업을 하나로 묶어 처리하는 것이 원자성이다.

NoteBase

문 4. 다음의 설명과 무선 PAN 기술이 옳게 짝지어진 것은?

(가) 다양한 기기 간에 무선으로 데이터 통신을 할 수 있도록 만든 기술로 에릭슨이 IBM, 노키아, 도시바와 함께 개발하였으며, IEEE 802.15.1 규격으로 발표되었다.
(나) 약 10 cm 정도로 가까운 거리에서 장치 간에 양방향 무선 통신을 가능하게 해주는 기술로 모바일 결제 서비스에 많이 활용된다.
(다) IEEE 802.15.4 기반 PAN기술로 낮은 전력을 소모하면서 저가의 센서 네트워크 구현에 최적의 방안을 제공하는 기술이다.

① (가): Bluetooth, (나): NFC, (다): ZigBee
② (가): ZigBee, (나): RFID, (다): Bluetooth
③ (가): NFC, (나): RFID, (다): ZigBee
④ (가): Bluetooth, (나): ZigBee, (다): RFID

정답 1번

간단한 해설

IEEE 802.15.1은 Bluetooth에, **IEEE 802.15.4 기반의 저전력 센서 네트워크**는 ZigBee에 대응한다. 기기를 가까이 대어 모바일 결제에 사용하는 기술은 **NFC**이다.

대응 관계 해설

(가) Bluetooth: 컴퓨터·휴대전화·주변기기 등을 가까운 거리에서 무선으로 연결하는 기술이다. 제시된 에릭슨·IBM·노키아·도시바의 개발 참여와 **IEEE 802.15.1**이 식별 단서이다. IEEE 802.15.1은 초기 Bluetooth 규격을 바탕으로 만든 무선 개인 영역 네트워크 표준이다.

(나) NFC: 매우 가까운 거리에서 장치 사이에 데이터를 주고받는 근거리 무선 통신 기술이다. 제시문의 '약 10 cm 정도로 가까운 거리', '양방향 무선 통신', **'모바일 결제 서비스'**가 NFC를 가리킨다. 휴대전화를 결제 단말기에 가까이 대어 결제하는 것이 대표적인 활용 사례이다.

(다) ZigBee: **IEEE 802.15.4를 기반으로 하는 저전력 통신 기술**이다. 센서가 측정값처럼 적은 양의 데이터를 전달하는 네트워크에 적합하며, 낮은 전력 소모와 저렴한 센서 네트워크 구현이라는 설명에 부합한다. Bluetooth를 가리키는 IEEE 802.15.1과 표준 번호를 구별해야 한다.

따라서 (가) Bluetooth, (나) NFC, (다) ZigBee의 대응이 성립하며, 이 조합은 ①과 일치한다.

NoteBase

문 5. 디스크 헤드의 위치가 55이고 0의 방향으로 이동할 때, C-SCAN 기법으로 디스크 대기 큐 25, 30, 47, 50, 63, 75, 100을 처리한다면 제일 마지막에 서비스 받는 트랙은?

① 50

② 63

③ 75

④ 100

정답 2번

간단한 해설

C-SCAN은 **한 방향으로만 요청을 처리**하고, 끝에 도달하면 반대쪽 끝으로 이동한 뒤 같은 방향으로 처리를 이어간다. 0 방향으로 이동하므로 55보다 큰 요청들은 순환 후 큰 번호부터 처리되어, 그중 가장 작은 **63이 마지막**에 서비스된다.

올바른 순서

50 → 47 → 30 → 25 → 100 → 75 → 63

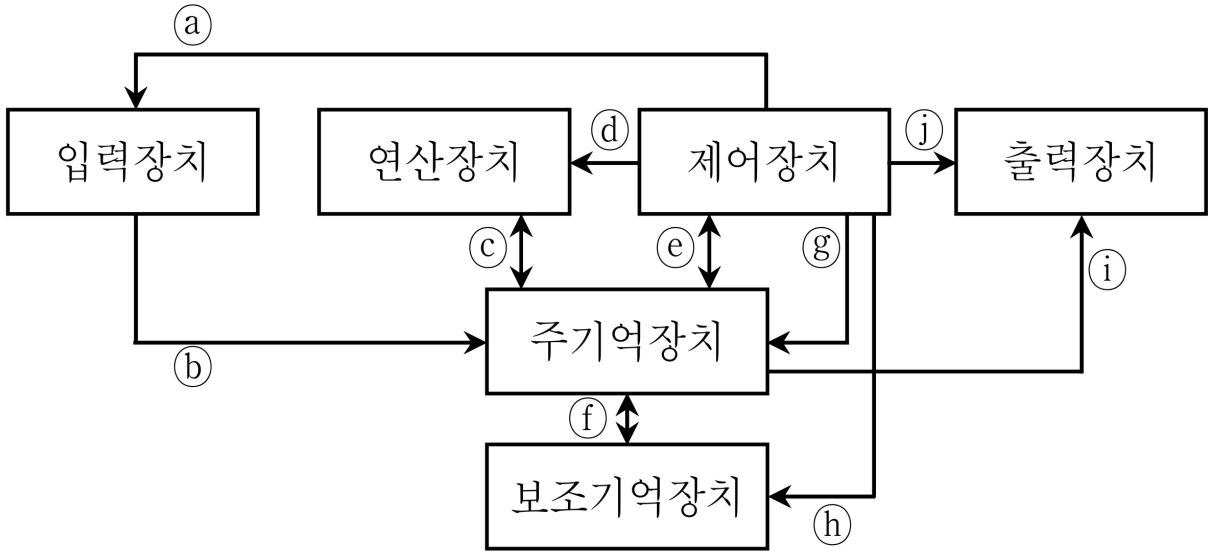
50·47·30·25: 헤드의 시작 위치는 55이고 이동 방향은 0 쪽이므로, 트랙 번호가 감소하는 방향에서 만나는 요청부터 처리한다. 따라서 55보다 작은 요청을 50, 47, 30, 25 순서로 서비스한다.

순환 이동: 25의 요청을 처리한 뒤에도 디스크의 끝인 0까지 이동한다. C-SCAN은 여기서 서비스 방향을 바꾸지 않고 반대쪽 끝인 최대 번호 트랙으로 이동하며, **이 순환 이동 중에는 요청을 처리하지 않는다**. 최대 트랙 번호가 주어지지 않아도 남은 요청의 서비스 순서는 결정할 수 있다.

100·75·63: 반대쪽 끝에 도착하면 다시 트랙 번호가 감소하는 방향으로 요청을 처리한다. 남은 요청은 63, 75, 100이므로 큰 번호부터 100, 75, 63 순서로 서비스한다. **63은 시작 위치 55보다 큰 요청 중 가장 작은 트랙**이어서 이 구간에서 마지막으로 서비스된다.

NoteBase

문 6. 컴퓨터 시스템 구성요소 사이의 데이터 흐름과 제어 흐름에 대한 설명으로 옳은 것은?



① ⓐ와 ⓕ는 모두 제어 흐름이다.

② ⓑ와 ⓖ는 모두 데이터 흐름이다.

③ ⓗ는 데이터 흐름, ⓓ는 제어 흐름이다.

④ ⓒ는 데이터 흐름, ⓖ는 제어 흐름이다.

정답 4번

간단한 해설

ⓒ는 주기억장치와 연산장치 사이에서 **연산할 데이터와 연산 결과**가 오가는 경로이다. ⓖ는 제어장치가 주기억장치에 읽기·쓰기 등의 동작을 지시하는 **제어 흐름**이다.

NoteBase

선지별 해설

데이터 흐름은 처리할 자료나 처리 결과가 이동하는 경로이고, 제어 흐름은 각 장치의 동작을 지시하는 신호가 전달되는 경로이다. 그림에서는 화살표가 연결하는 장치와 전달하는 정보의 역할을 함께 살펴야 한다.

① X: ⓐ는 제어장치에서 입력장치로 동작을 지시하는 제어 흐름이다. 반면 ⓕ는 주기억장치와 보조기억장치 사이에서 저장할 자료나 읽어 온 자료가 오가는 **데이터 흐름**이므로, 둘 다 제어 흐름이라는 설명은 틀리다.

② X: ⓑ는 입력장치가 받아들인 자료를 주기억장치로 보내는 데이터 흐름이다. 그러나 ⓖ는 제어장치에서 주기억장치로 읽기·쓰기 등의 동작을 지시하는 **제어 흐름**이다. 따라서 ⓑ와 ⓖ는 각각 데이터 흐름과 제어 흐름이다.

③ X: ⓗ는 제어장치에서 보조기억장치로 동작을 지시하는 제어 흐름이고, ⓓ는 제어장치에서 연산장치로 수행할 연산을 지시하는 제어 흐름이다. ⓗ를 데이터 흐름이라고 한 부분을 **제어 흐름**으로 고쳐야 한다.

④ O: ⓒ는 주기억장치와 연산장치 사이의 양방향 경로로, 연산에 필요한 데이터를 가져오거나 연산 결과를 저장할 때 이용하는 **데이터 흐름**이다. ⓖ는 제어장치에서 주기억장치로 읽기·쓰기 등의 동작을 지시하는 **제어 흐름**이므로 두 경로의 분류가 모두 맞다.

NoteBase

문 7. 수식의 결과가 거짓(false)인 것은?

① $20D_{(16)} > 524_{(10)}$

② $0.125_{(10)} = 0.011_{(2)}$

③ $10_{(8)} = 1000_{(2)}$

④ $0.1_{(10)} < 0.1_{(2)}$

정답 2번

간단한 해설

이진 소수에서 소수점 아래 첫째·둘째·셋째 자리의 값은 각각 **1/2, 1/4, 1/8**이다. 따라서 $0.011_{(2)} = 0.375_{(10)}$이며, $0.125_{(10)}$의 이진수 표현은 $0.001_{(2)}$이다.

선지별 해설

수의 아래 첨자는 진법을 나타낸다. 각 숫자에 해당 자리의 자릿값을 곱한 뒤 모두 더하면 십진수 값이 된다. 소수점 왼쪽으로 갈수록 자릿값은 밑의 거듭제곱으로 커지고, 오른쪽으로 갈수록 밑으로 나눈 값이 된다.

① O: 16진수에서 D는 **십진수 13**을 나타낸다. 따라서 $20D_{(16)} = 2 \cdot 16^2 + 13$이며, 십진수 값은 512 + 13 = 525이다. 525 > 524이므로 주어진 부등식은 참이다.

② X: 이진 소수는 소수점 아래 첫째 자리부터 자릿값이 1/2, 1/4, 1/8이다. 따라서 $0.011_{(2)} = \frac{1}{4} + \frac{1}{8} = 0.375_{(10)}$이다. 이는 십진수 0.125와 다르다. 0.125는 1/8이므로 **소수점 아래 셋째 자리만 1**인 $0.001_{(2)}$로 고쳐야 한다.

③ O: $10_{(8)}$은 8의 자리에 1이 있으므로 십진수 8이다. $1000_{(2)}$도 2^3의 자리에 1이 있으므로 십진수 8이다. 두 수는 **같은 값인 8**을 나타내므로 등식은 참이다.

④ O: 십진수 $0.1_{(10)}$은 1/10이고, 이진수 $0.1_{(2)}$은 1/2이므로 십진수로 0.5이다. 따라서 0.1 < 0.5이다. **같은 표기라도 진법에 따라 자릿값이 다르므로** 주어진 부등식은 참이다.

NoteBase

문 8. '인터넷 서점'에 대한 유스케이스 다이어그램에서 '회원등록' 유스케이스를 수행하기 위해서는 '실명확인' 유스케이스가 반드시 선행되어야 한다면 이들의 관계는?

① 일반화(generalization) 관계

② 확장(extend) 관계

③ 포함(include) 관계

④ 연관(association) 관계

정답 3번

간단한 해설

'회원등록'을 수행하는 데 '실명확인'이 **반드시 필요한 기능**이므로, 회원등록이 실명확인을 포함하는 **포함(include) 관계**이다.

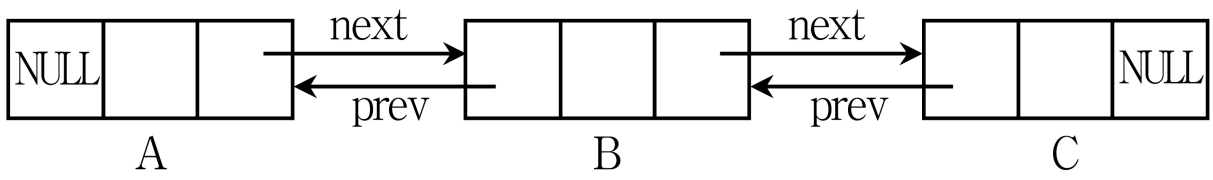
공통 해설

포함(include) 관계는 한 유스케이스가 다른 유스케이스의 기능을 자신의 수행 과정에 필수적으로 사용하는 관계이다. 포함되는 기능은 기본 유스케이스를 올바르게 수행하는 데 필요하다.

이 문제에서는 '실명확인'을 거쳐야 '회원등록'을 수행할 수 있으므로, 실명확인이 **회원등록의 필수 기능**에 해당한다. 따라서 회원등록은 포함하는 유스케이스이고, 실명확인은 포함되는 유스케이스이다.

NoteBase

문 9. 노드 A, B, C를 가지는 이중 연결 리스트에서 노드 B를 삭제하기 위한 의사코드(pseudo code)로 옳지 않은 것은? (단, 노드 B의 메모리는 해제하지 않는다)



① A->next = C;
C->prev = A;

② A->next = B->next;
C->prev = B->prev;

③ B->prev->next = B->next;
B->next->prev = B->prev;

④ A->next = A->next->next;
A->next->next->prev = B->prev;

정답 4번

간단한 해설

④는 첫 문장 실행 후 A->next가 C로 바뀐다는 점을 놓쳤다. 두 번째 문장의 A->next->next는 C의 next인 **NULL**이므로 prev에 접근할 수 없으며, **A->next->prev = B->prev;**로 고쳐야 한다.

NoteBase

선지별 해설

이중 연결 리스트에서 next는 다음 노드, prev는 이전 노드를 가리킨다. 그림의 초기 상태는 A->next가 B, B->prev가 A, B->next가 C, C->prev가 B이며, A->prev와 C->next는 NULL이다. B를 리스트에서 삭제하려면 A->next를 C로, C->prev를 A로 바꾸어 B를 건너뛰도록 연결해야 한다. B의 메모리는 해제하지 않는 조건이므로, B 자체의 prev와 next를 지우지 않아도 된다. 각 선지는 이 초기 상태에서 위에서 아래로 실행한다.

① O: A->next = C;는 A의 다음 노드를 C로 바꾸고, C->prev = A;는 C의 이전 노드를 A로 바꾼다. 두 문장이 **A와 C를 양방향으로 직접 연결**하므로 B가 리스트에서 제외된다.

② O: B->next는 C이므로 첫 문장은 A->next에 C를 저장한다. 이 대입은 B의 필드를 바꾸지 않으므로, 두 번째 문장에서 B->prev는 여전히 A이다. 따라서 C->prev에 A가 저장되어 필요한 두 연결이 완성된다.

③ O: B->prev가 A이므로 B->prev->next는 A의 next 필드이다. 첫 문장은 이 필드에 B->next인 C를 저장한다. 이어서 B->next->prev는 C의 prev 필드이고, 여기에 B->prev인 A를 저장한다. **B의 prev와 next는 그대로 유지**되므로 첫 대입 후에도 두 번째 문장에서 이들을 이용할 수 있다.

④ X: 첫 문장의 A->next->next는 대입 전에는 B->next이므로 C를 가리킨다. 따라서 첫 문장 실행 후 A->next는 C가 된다. 이 상태에서 두 번째 문장의 A->next->next는 C->next인 NULL이다. 결국 NULL의 prev 필드에 접근하려 하므로 **C의 prev를 수정할 수 없다**. 두 번째 문장을 A->next->prev = B->prev;로 고치면, 현재 A->next가 가리키는 C의 prev에 A를 저장할 수 있다.

NoteBase

문 10. 이동 애드혹 네트워크(MANET)에 대한 설명으로 옳지 않은 것은?

① 전송 거리와 전송 대역폭에 제약을 받는다.

② 노드는 호스트 기능과 라우팅 기능을 동시에 가진다.

③ 보안 및 라우팅 지원이 여러 노드 간의 협력에 의해 분산 운영된다.

④ 동적인 네트워크 토폴로지를 효율적으로 구성하기 위해 액세스 포인트(AP)와 같은 중재자를 필요로 한다.

정답 4번

간단한 해설

④의 'AP와 같은 중재자를 필요로 한다'는 설명이 잘못되었다. MANET은 **고정된 기반 시설 없이** 이동 노드들이 서로 협력하여 네트워크와 통신 경로를 구성한다.

선지별 해설

① O: 무선 신호가 도달하는 범위가 한정되어 있으므로 **노드 간 직접 전송 거리**에 제약이 있다. 무선 채널의 대역폭도 제한되며, 간섭이나 여러 노드의 채널 공유로 실제 전송 성능이 낮아질 수 있다.

② O: 노드는 데이터를 생성·수신하는 호스트 역할과 다른 노드의 패킷을 전달하는 **라우팅 역할을 함께 수행**한다. 예를 들어 A와 C가 직접 통신할 수 없지만 둘 다 B와 연결되어 있다면, B가 A의 패킷을 C로 중계할 수 있다.

③ O: 중앙 관리 장치에 의존하지 않는 구조이므로 보안 및 라우팅 지원은 **여러 노드의 협력으로 분산 수행**된다. 노드들은 경로 정보를 교환하고 통신 상대를 인증하는 등의 보안 기능을 수행하며 네트워크 운영에 참여한다.

④ X: MANET은 **AP와 같은 고정 중재자 없이 자율적으로 구성**되는 네트워크이다. 노드의 이동으로 연결 관계인 토폴로지가 바뀌면 노드들이 협력하여 경로를 갱신한다. AP를 중심으로 통신하는 기반 구조 방식과 달리, 동적인 토폴로지를 구성하기 위해 AP가 필수인 것은 아니다.

NoteBase

문 11. 공개키 암호화 방법을 사용하여 철수가 영희에게 메시지를 보내는 것에 대한 설명으로 옳지 않은 것은?

① 공개키는 누구에게나 공개된다.

② 공개키의 위조 방지를 위해 인증기관은 인증서를 발급한다.

③ 철수는 자신의 공개키를 사용하여 평문을 암호화한다.

④ 영희는 자신의 개인키를 사용하여 암호문을 복호화한다.

정답 3번

간단한 해설

③의 '자신의 공개키'는 **수신자인 영희의 공개키**로 고쳐야 한다. 영희의 공개키로 암호화해야 그와 짝을 이루는 영희의 개인키로 복호화할 수 있다.

선지별 해설

① O: 공개키 암호화에서는 공개키와 개인키를 한 쌍으로 사용한다. 공개키는 다른 사람이 암호화에 사용할 수 있도록 공개하며, **비밀로 보관해야 하는 것은 개인키**이다.

② O: 인증기관은 **공개키와 그 소유자의 신원을 연결**하는 인증서에 전자서명하여 발급한다. 이를 통해 철수는 받은 공개키가 실제로 영희의 것인지 확인할 수 있어, 공격자가 자신의 공개키를 영희의 것처럼 속이는 일을 방지할 수 있다.

③ X: 철수는 자신의 공개키가 아니라 **영희의 공개키로 암호화**해야 한다. 철수의 공개키로 암호화하면 그와 짝인 철수의 개인키가 복호화에 필요하므로, 영희가 자신의 개인키로 메시지를 복호화하는 목적에 맞지 않는다.

④ O: 영희는 자신의 공개키로 암호화된 메시지를 **자신의 개인키로 복호화**한다. 암호화에 사용한 공개키와 복호화에 사용하는 개인키가 영희의 키 쌍을 이루므로, 개인키를 비밀로 보관한 영희만 메시지의 내용을 읽을 수 있다.

NoteBase

문 12. 네트워크 구성 형태에 대한 설명으로 옳지 않은 것은?

① 메시(mesh)형은 각 노드가 다른 모든 노드와 점 대 점으로 연결되기 때문에 네트워크 규모가 커질수록 통신 회선 수가 급격하게 많아진다.

② 스타(star)형은 각 노드가 허브라는 하나의 중앙노드에 연결되기 때문에 중앙노드가 고장나면 그 네트워크 전체가 영향을 받는다.

③ 트리(tree)형은 고리처럼 순환형으로 구성된 형태로서 네트워크 재구성이 수월하다.

④ 버스(bus)형은 하나의 선형 통신 회선에 여러 개의 노드가 연결되어 있는 형태이다.

정답 3번

간단한 해설

③의 '고리처럼 순환형'은 **링(ring)형**의 특징이다. 트리(tree)형은 나뭇가지처럼 상위 노드에서 하위 노드로 뻗어 나가는 **계층 구조**이며 순환 경로가 없다.

선지별 해설

① O: 선지와 같이 모든 노드 쌍을 직접 연결하는 완전 메시형에서는 **노드를 하나 추가할 때 기존의 모든 노드와 새 회선을 연결**해야 한다. 기존 노드가 많을수록 추가할 회선도 많아지므로, 네트워크 규모가 커질수록 전체 회선 수가 빠르게 증가한다.

② O: 스타형은 각 노드가 **중앙노드를 거쳐 통신**하는 구조이다. 중앙의 허브가 고장 나면 노드 사이의 통신을 이어줄 수 없으므로 네트워크 전체가 영향을 받는다.

③ X: 트리형은 상위 노드에 하위 노드가 가지처럼 연결되는 **순환 경로가 없는 계층 구조**이다. 하위 가지에 노드를 추가하는 방식으로 확장할 수 있다. 노드들이 고리처럼 이어져 순환 경로를 이루는 형태는 링(ring)형이므로, 트리형을 '고리처럼 순환형'이라고 한 부분이 잘못되었다.

④ O: 버스형은 여러 노드가 **하나의 선형 통신 회선을 공유**하는 구조이다. 각 노드가 공통 회선에 연결되어 그 회선을 통해 데이터를 주고받으므로 선지의 설명과 일치한다.

NoteBase

문 13. 다음에서 설명하는 보안공격방법은?

공격자는 여러 대의 좀비 컴퓨터를 분산 배치하여 가상의 접속자를 만든 후 처리할 수 없을 정도로 매우 많은 양의 패킷을 동시에 발생시켜 시스템을 공격한다. 공격받은 컴퓨터는 사용자가 정상적으로 접속할 수 없다.

① 키로거(Key Logger)
② DDoS(Distributed Denial of Service)
③ XSS(Cross Site Scripting)
④ 스파이웨어(Spyware)

정답 2번

간단한 해설

여러 대의 좀비 컴퓨터가 동시에 대량의 패킷을 보내 정상적인 서비스 이용을 막는 것은 **분산 서비스 거부 공격(DDoS)**이다.

공통 해설

DDoS(Distributed Denial of Service)는 여러 컴퓨터를 이용해 대상 시스템이나 네트워크에 부하를 집중시켜 정상적인 사용자가 서비스를 이용하지 못하게 하는 분산 서비스 거부 공격이다. 지문의 좀비 컴퓨터는 악성코드 등에 감염되어 공격자의 명령에 따라 동작하는 컴퓨터를 뜻한다. 이들이 동시에 대량의 패킷을 보내면 네트워크 대역폭이나 시스템의 처리 자원이 소모되어 정상적인 접속 요청을 처리하기 어려워진다.

지문의 '여러 대의 좀비 컴퓨터'는 **공격 발생 지점이 분산되어 있음**을, '사용자가 정상적으로 접속할 수 없다'는 **서비스의 가용성이 침해됨**을 보여 준다. 이 두 특징이 함께 나타나므로 설명된 공격은 DDoS에 해당한다.

NoteBase

문 14. 논리적 데이터 모델에 대한 설명으로 옳지 않은 것은?

① 개체관계 모델은 개체와 개체 사이의 관계성을 이용하여 데이터를 모델링한다.

② 관계형 모델은 논리적 데이터 모델에 해당한다.

③ SQL은 관계형 모델을 따르는 DBMS의 표준 데이터 언어이다.

④ 네트워크 모델, 계층 모델은 레거시 데이터 모델로도 불린다.

정답 1번

간단한 해설

개체관계(E-R) 모델은 현실 세계의 개체와 관계를 표현하는 **개념적 데이터 모델**이다. ①의 모델링 방식 자체는 맞지만, 개념적 모델과 논리적 모델을 구분하는 이 문항에서는 논리적 데이터 모델에 대한 설명으로 적절하지 않다.

선지별 해설

① X: 개체관계(E-R) 모델은 개체, 속성, 개체 사이의 관계를 이용하여 현실 세계의 데이터 요구사항을 표현하는 **개념적 데이터 모델**이다. 예를 들어 학생과 과목을 개체로, 학생이 과목을 수강하는 것을 관계로 표현한다. 따라서 '개체와 개체 사이의 관계성을 이용한다'는 설명 자체는 맞다. 다만 이 문항의 분류에서는 개체관계 모델을 논리적 모델에 포함시키는 것이 부적절하며, 개념적 모델로 구분해야 한다.

② O: 관계형 모델은 데이터를 **행과 열로 이루어진 릴레이션**으로 표현하는 논리적 데이터 모델이다. 릴레이션은 표 형태이며, 행은 하나의 튜플을, 열은 속성을 나타낸다. 개념적 설계에서 파악한 데이터와 관계를 이러한 구조로 구체화한다.

③ O: SQL은 **관계형 DBMS의 표준 데이터 언어**이다. DBMS는 데이터베이스를 관리하는 소프트웨어이며, SQL로 테이블 구조를 정의하거나 데이터를 검색·삽입·수정·삭제할 수 있다. 관계형 모델이 데이터의 구조를 정한다면 SQL은 그 구조의 데이터를 정의하고 다루는 언어이다.

④ O: 계층 모델은 부모와 자식의 관계를 트리 구조로 표현하고, 네트워크 모델은 레코드 사이의 연결을 망 구조로 표현한다. 두 모델은 관계형 모델이 널리 보급되기 전부터 사용된 전통적인 모델로, **레거시 데이터 모델**로도 불린다. 레거시는 기존 시스템에 남아 사용되는 오래된 기술을 가리키므로, 현재 전혀 사용되지 않는다는 뜻은 아니다.

NoteBase

문 15. 다음에서 설명하는 소프트웨어 개발 방법론은?

◯ 애자일 방법론의 하나로 소프트웨어 개발 프로세스가 문서화하는 데 지나치게 많은 시간과 노력이 소모되는 단점을 보완하기 위해 개발되었다.
◯ 의사소통, 단순함, 피드백, 용기, 존중의 5가지 가치에 기초하여 '고객에게 최고의 가치를 가장 빨리' 전달하도록 하는 방법론으로 켄트 벡이 고안하였다.

① 통합 프로세스(UP)
② 익스트림 프로그래밍
③ 스크럼
④ 나선형 모델

정답 2번

간단한 해설

켄트 벡이 고안하고 **의사소통·단순함·피드백·용기·존중**의 다섯 가치를 바탕으로 하는 애자일 방법론은 익스트림 프로그래밍(XP)이다.

공통 해설

익스트림 프로그래밍(XP)은 켄트 벡이 고안한 애자일 소프트웨어 개발 방법론이다. 과도한 문서 작성에 드는 부담을 줄이고, 고객과 개발자가 소통하면서 작동하는 소프트웨어를 자주 확인하고 개선하여 고객에게 가치를 빠르게 전달한다.

XP의 다섯 가치는 **의사소통·단순함·피드백·용기·존중**이다. 의사소통은 요구사항과 개발 상황을 함께 이해하는 것이고, 단순함은 현재 필요한 기능을 단순하게 설계하는 것이다. 피드백은 테스트 결과와 고객의 반응을 개발에 반영하는 것이며, 용기는 필요한 변경과 문제 해결을 회피하지 않는 태도이다. 존중은 팀 구성원 서로의 역할과 기여를 인정하는 것이다. 지문에서 애자일이라는 설명에 더해 이 다섯 가치와 켄트 벡을 제시했으므로, 설명에 해당하는 방법론을 XP로 특정할 수 있다.

NoteBase

문 16. 다음 프로세스 집합에 대하여 라운드 로빈 CPU 스케줄링 알고리즘을 사용할 때, 프로세스들의 총 대기시간은? (단, 시간 0에 P1, P2, P3 순서대로 도착한 것으로 하고, 시간 할당량은 4밀리초로 하며, 프로세스 간 문맥교환에 따른 오버헤드는 무시한다)

프로세스	버스트 시간(밀리초)
P1	20
P2	3
P3	4

① 16

② 18

③ 20

④ 24

정답 2번

간단한 해설

라운드 로빈에서는 **할당량을 다 쓰고도 끝나지 않은 프로세스를 준비 큐의 맨 뒤로 보낸다**. P1은 P2와 P3이 실행되는 동안 7밀리초를 기다리므로, 각 대기시간의 합은 7 + 4 + 7 = 18밀리초이다.

NoteBase

계산 과정

조건과 실행 원리: 표의 버스트 시간은 P1이 20밀리초, P2가 3밀리초, P3이 4밀리초이다. 세 프로세스는 모두 시각 0에 도착하며, 처음 준비 큐의 순서는 P1 → P2 → P3이다. 라운드 로빈은 한 번에 최대 4밀리초씩 CPU를 배정하고, **실행이 끝나지 않으면 준비 큐의 맨 뒤로 이동**시킨다. 문맥교환 오버헤드는 없으므로 전환에 걸리는 시간은 더하지 않는다.

실행 구간: 0~4밀리초에는 P1이 실행되어 16밀리초가 남는다. 4~7밀리초에는 P2가 실행되며, 버스트 시간이 3밀리초이므로 **할당량 4밀리초를 채우지 않고 종료**한다. 이어서 7~11밀리초에 P3이 실행되어 종료된다. 이후에는 P1만 남으므로 남은 16밀리초를 4밀리초씩 연속으로 실행하여 시각 27밀리초에 종료한다.

P1의 대기시간: 시각 0에 바로 실행되므로 처음 대기시간은 없다. 첫 실행이 끝난 시각 4부터 다시 실행되는 시각 11까지 기다리므로 11 − 4 = 7밀리초이다. 이후에는 다른 프로세스가 없어 추가 대기시간이 없다.

P2의 대기시간: 도착 시각 0부터 처음 실행되는 시각 4까지 기다린다. 한 번의 CPU 배정으로 종료되므로 대기시간은 4 − 0 = 4밀리초이다.

P3의 대기시간: 도착 시각 0부터 처음 실행되는 시각 7까지 기다린다. 한 번의 CPU 배정으로 종료되므로 대기시간은 7 − 0 = 7밀리초이다.

총 대기시간: 대기시간은 프로세스가 준비 큐에서 CPU를 기다린 시간으로, 자신의 실행시간은 포함하지 않는다. 세 프로세스의 대기시간을 합하면 7 + 4 + 7 = 18밀리초이며, 선택지 ②의 18에 해당한다.

NoteBase

문 17. 다음 C 프로그램의 출력 값은?

```
#include <stdio.h>

void funCount();

int main(void) {
        int num;
        for(num=0; num<2; num++)
                funCount();
        return 0;
}

void funCount() {
        int num=0;
        static int count;

        printf("num = %d, count = %d\n",
                ++num, count++);
}
```

① num = 0, count = 0
num = 0, count = 1
② num = 0, count = 0
num = 1, count = 1
③ num = 1, count = 0
num = 1, count = 0
④ num = 1, count = 0
num = 1, count = 1

정답 4번

간단한 해설

funCount의 num은 호출할 때마다 0으로 초기화되고, **전위 증가 ++num**에 의해 1이 출력된다. count는 **static 변수라 이전 값을 유지**하며, 후위 증가 count++는 증가 전 값을 전달하므로 두 번의 출력에서 각각 0과 1이 나온다.

NoteBase

계산 과정

호출 횟수와 변수: main의 for문은 num이 0, 1일 때 실행되므로 funCount를 두 번 호출한다. main의 num과 funCount의 num은 **이름만 같은 별개의 지역 변수**이다. funCount의 int num=0은 호출할 때마다 실행되지만, static int count는 처음에 0으로 초기화된 뒤 함수가 끝나도 값을 유지한다.

증가 연산과 출력: ++num은 값을 먼저 1 증가시킨 뒤 증가한 값을 사용한다. count++는 증가 전 값을 사용하고 count를 1 증가시킨다. 따라서 printf에 전달되는 값은 **num은 증가 후 값, count는 증가 전 값**이다. 두 연산은 서로 다른 변수에 적용되므로 인수 평가 순서에 따라 출력이 달라지지 않는다.

첫 번째 호출: num과 count는 모두 0에서 시작한다. ++num이 전달하는 값은 1이고, count++가 전달하는 값은 0이므로 'num = 1, count = 0'이 출력된다. 이 호출이 끝나면 count에는 1이 남는다.

두 번째 호출: num은 다시 0으로 초기화되지만, count는 이전 호출에서 남은 1로 시작한다. ++num은 1을, count++는 1을 전달하므로 'num = 1, count = 1'이 출력된다. 이후 count는 2가 된다.

출력은 첫 줄 'num = 1, count = 0', 둘째 줄 'num = 1, count = 1'이며, ④의 출력과 일치한다.

NoteBase

문 18. 페이지 크기가 2,000 byte인 페이징 시스템에서 페이지테이블이 다음과 같을 때 논리주소에 대한 물리주소가 옳게 짝지어진 것은? (단, 논리주소와 물리주소는 각각 0에서 시작되고, 1 byte 단위로 주소가 부여된다)

페이지번호(논리)	프레임번호(물리)
0	7
1	3
2	5
3	0
4	8

① 논리주소: 4,300, 물리주소: 2,300

② 논리주소: 3,600, 물리주소: 4,600

③ 논리주소: 2,500, 물리주소: 6,500

④ 논리주소: 900, 물리주소: 7,900

정답 3번

간단한 해설

페이징에서는 페이지 번호를 프레임 번호로 바꾸고 **페이지 내부의 변위는 그대로 유지**한다. 논리주소 2,500은 페이지 1의 변위 500이고, 페이지 1은 프레임 3에 있으므로 물리주소는 3 × 2,000 + 500 = 6,500이다.

NoteBase

계산 과정

페이지와 프레임의 크기는 모두 2,000 byte이다. 주소가 0부터 시작하므로 논리주소를 2,000으로 나눈 몫은 페이지 번호, 나머지는 페이지 안에서의 위치인 변위이다. 페이지테이블에서 해당 페이지의 프레임 번호를 찾은 뒤, **프레임 번호 × 2,000 + 변위**로 물리주소를 구한다. 프레임 번호 × 2,000은 그 프레임의 시작 주소이며, 주소 하나가 1 byte를 가리키므로 변위를 그대로 더한다.

논리주소 4,300: 4,300 = 2 × 2,000 + 300이므로 페이지 2, 변위 300이다. 표에서 페이지 2는 프레임 5에 대응하므로 물리주소는 5 × 2,000 + 300 = 10,300이다.

논리주소 3,600: 3,600 = 1 × 2,000 + 1,600이므로 페이지 1, 변위 1,600이다. 표에서 페이지 1은 프레임 3에 대응하므로 물리주소는 3 × 2,000 + 1,600 = 7,600이다.

논리주소 2,500: 2,500 = 1 × 2,000 + 500이므로 페이지 1, 변위 500이다. 표에서 페이지 1은 프레임 3에 대응하므로 물리주소는 3 × 2,000 + 500 = 6,500이다.

논리주소 900: 첫 페이지의 주소 범위는 0~1,999이므로 페이지 0, 변위 900이다. 표에서 페이지 0은 프레임 7에 대응하므로 물리주소는 7 × 2,000 + 900 = 14,900이다.

계산한 물리주소와 제시된 값을 대조하면, 논리주소 2,500과 물리주소 6,500을 짝지은 ③이 일치한다.

NoteBase

문 19. HTML5의 특징에 대한 설명으로 옳지 않은 것은?

① 플러그인의 도움 없이 음악과 동영상 재생이 가능하다.

② 쌍방향 통신을 제공하여 실시간 채팅이나 온라인 게임을 만들 수 있다.

③ 디바이스에 접근할 수 없어서 개인정보 보호 및 보안을 철저히 유지할 수 있다.

④ 스마트폰의 일반 응용프로그램도 HTML5를 사용해 개발할 수 있다.

정답 3번

간단한 해설

③의 '디바이스에 접근할 수 없다'는 설명이 잘못되었다. HTML5 관련 웹 API는 사용자 허가 등 조건에 따라 **기기의 위치 정보나 카메라·마이크에 접근**할 수 있으므로, 기기 접근 자체가 불가능해서 보안이 보장되는 것은 아니다.

선지별 해설

① O: HTML5의 audio와 video 요소를 사용하면 브라우저가 음악과 동영상을 직접 재생할 수 있다. 따라서 플래시 같은 **별도 플러그인 없이** 멀티미디어 콘텐츠를 제공할 수 있다.

② O: HTML5와 함께 활용되는 WebSocket은 브라우저와 서버 사이의 **양방향 통신**을 지원한다. 연결을 유지한 상태에서 양쪽이 데이터를 보낼 수 있으므로 실시간 채팅 메시지나 온라인 게임의 상태 변화를 주고받는 데 이용할 수 있다.

③ X: HTML5 관련 웹 기술은 기기 기능에 대한 접근을 모두 금지하지 않는다. 예를 들어 Geolocation API로 기기의 위치 정보를 얻을 수 있고, 관련 미디어 API로 카메라·마이크를 사용할 수 있다. 다만 이러한 접근에는 **사용자 허가 등 보안 조건**이 적용된다. 따라서 '디바이스에 접근할 수 없다'는 '허용된 범위에서 기기 기능과 정보에 접근할 수 있다'로 고쳐야 하며, 접근이 불가능하다는 이유로 개인정보 보호와 보안이 철저히 보장된다고 할 수 없다.

④ O: HTML5를 CSS·JavaScript와 함께 사용하면 화면을 구성하고 사용자 입력을 처리하는 스마트폰용 응용프로그램을 개발할 수 있다. 모바일 브라우저에서 실행하는 **웹 앱**으로 만들거나, 웹 콘텐츠를 앱 실행 환경에 넣는 하이브리드 앱 형태로 구현할 수 있다.

NoteBase

문 20. 컴퓨터의 발전 과정에 대한 설명으로 옳지 않은 것은?

① 포트란, 코볼같은 고급 언어는 집적회로(IC)가 적용된 제3세대 컴퓨터부터 사용되었다.

② 애플사는 1970년대에 개인용 컴퓨터를 출시하였다.

③ IBM PC라고 불리는 컴퓨터는 1980년대에 출시되었다.

④ 1990년대에는 월드와이드웹 기술이 적용되면서 인터넷에 연결되는 컴퓨터의 사용자가 폭발적으로 증가하였다.

정답 1번

간단한 해설

포트란과 코볼 같은 고급 언어는 **제3세대 이전부터 사용**되었다. 고급 언어의 보급은 트랜지스터를 사용한 **제2세대 컴퓨터**의 주요 특징이다.

선지별 해설

① X: 고급 언어를 '제3세대부터' 사용했다는 부분이 틀렸다. 포트란은 1957년에 등장했으며, 포트란·코볼은 **제2세대 컴퓨터에서 이미 사용**되었다. 제2세대는 트랜지스터, 제3세대는 집적회로(IC)를 주요 회로 소자로 사용했다.

② O: 애플은 **1977년 애플 II**를 출시하였다. 개인이 사용할 수 있는 컴퓨터를 1970년대에 출시했다는 설명과 일치한다.

③ O: IBM PC의 대표적인 초기 모델인 IBM 5150은 **1981년**에 출시되었다. 따라서 IBM PC의 출시 시기는 1980년대이다.

④ O: 월드와이드웹은 인터넷을 통해 **하이퍼링크로 연결된 문서를 열람**할 수 있게 하는 서비스이다. 1990년대 웹과 웹 브라우저가 보급되면서 일반인의 정보 접근이 쉬워졌고, 인터넷 이용자가 급격히 증가하였다.

NoteBase