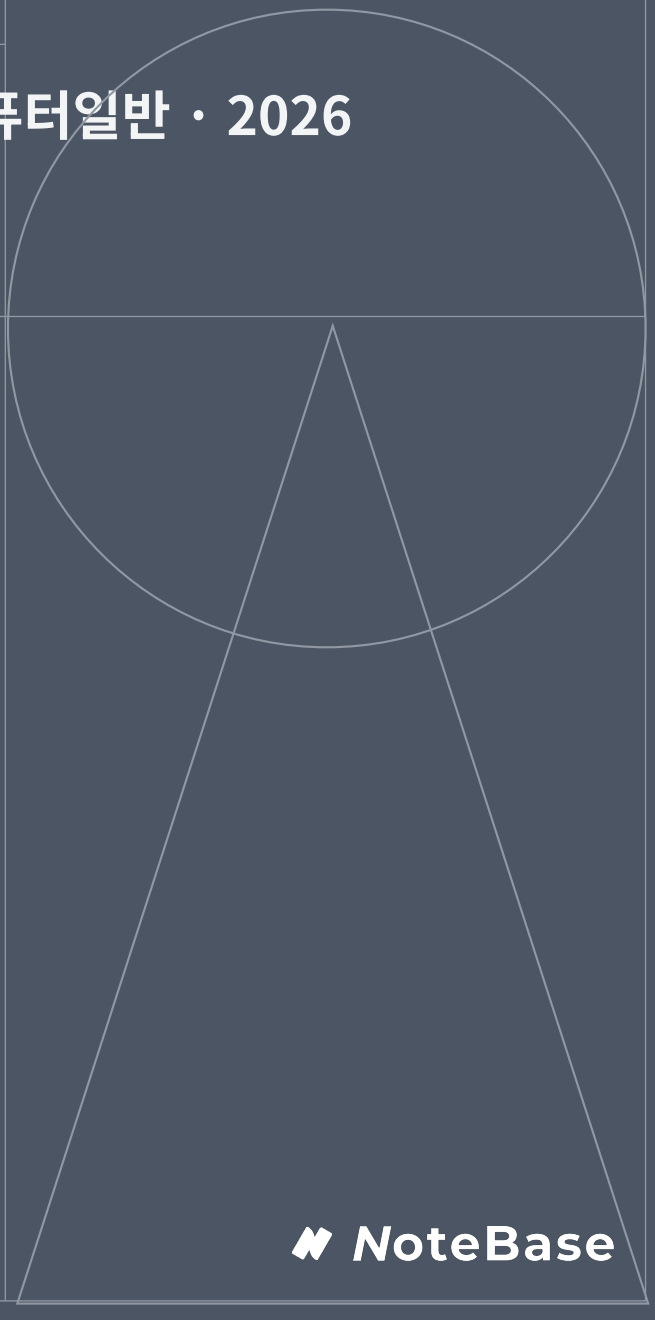
기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2026



NoteBase

기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2026

공무원/소방/경력채용 · 컴퓨터일반 · 2026

문 1. 캐시메모리에 대한 설명으로 옳지 않은 것은?

① CPU와 주기억장치 사이에 위치한 장치로 데이터 접근 속도를 향상시킨다.

② CPU와 주기억장치 간 처리속도의 차이로 인하여 발생하는 병목현상을 해결하기 위해 사용된다.

③ 수준-2(level-2) 캐시메모리가 수준-1(level-1) 캐시메모리보다 CPU에 더 가깝게 있어 접근 속도가 더 빠르다.

④ CPU가 데이터를 읽을 때, 주기억장치보다 캐시메모리를 먼저 조사하여 원하는 데이터가 있으면 바로 CPU로 전달한다.

정답 3번

간단한 해설

캐시 계층에서는 일반적으로 **L1이 CPU 코어에 가장 가깝고 접근 지연이 가장 짧다**. L2는 L1보다 용량이 큰 대신 보통 접근이 더 느리므로, 두 계층의 관계를 반대로 설명한 부분이 틀렸다.

선지별 해설

① (O) 캐시는 메모리 계층상 CPU와 주기억장치 사이에서 자주 사용하거나 곧 사용할 가능성이 높은 데이터와 명령어를 보관한다. 캐시에서 필요한 정보를 찾으면 주기억장치까지 접근하는 시간을 줄일 수 있다.

② (O) CPU의 연산 속도에 비해 주기억장치 접근은 느리므로 CPU가 데이터를 기다리는 병목이 생긴다. 고속 캐시는 이러한 속도 차이를 완화하여 평균 메모리 접근 시간을 줄이는 데 사용된다.

③ (X) 일반적으로 **L1 캐시가 L2 캐시보다 CPU 코어에 가깝고 접근 속도가 빠르다**. L2는 L1보다 용량이 큰 대신 접근 지연이 더 긴 계층이다. 따라서 L2가 L1보다 가깝고 빠르다는 설명은 반대이다.

④ (O) 캐시를 사용하는 일반적인 읽기에서는 필요한 데이터가 캐시에 있는지 확인한다. 있으면 캐시 적중으로 데이터를 제공하고, 없으면 하위 캐시나 주기억장치에 접근한다. 필요한 데이터를 항상 주기억장치에서만 읽는 것은 아니다.

NoteBase

문 2. 인공지능에 대한 설명으로 옳지 않은 것은?

① 인공신경망은 인간 두뇌의 생물학적 뉴런 작용을 모방하였다.

② 머신러닝의 학습 방법에는 지도학습, 비지도학습, 강화학습 등이 있다.

③ 지도학습은 보상을 얻기 위한 정책을 다음 행동에 반영시키는 학습 방법이다.

④ 심층신경망은 입력층과 출력층 사이에 여러 개의 은닉층으로 구성된 인공신경망이다.

정답 3번

간단한 해설

정답이 붙은 데이터로 학습하는 것은 지도학습이고, **보상을 바탕으로 행동 정책을 학습**하는 것은 강화학습이다.

선지별 해설

① O: 인공신경망은 뉴런들이 연결되어 신호를 주고받는 생물학적 작용에서 착안한 계산 모형이다. 입력에 가중치를 적용하고 활성화 함수를 거쳐 출력을 만드는 등 **뉴런의 정보 처리를 추상화**한 것이며, 실제 두뇌를 그대로 재현한 것은 아니다.

② O: 지도학습은 입력과 정답의 쌍으로 예측 관계를 학습하고, 비지도학습은 정답 표지 없이 데이터의 구조나 패턴을 찾는다. 강화학습은 **환경과의 상호작용에서 얻는 보상**을 바탕으로 행동 방식을 학습한다. 이들은 머신러닝의 대표적인 학습 방식이다.

③ X: 보상을 얻기 위한 정책을 학습하여 행동 선택에 반영하는 것은 **강화학습**에 대한 설명이다. 지도학습은 정답이 붙은 예시를 이용하여 입력에 대응하는 출력을 학습하는 방식이므로 구별해야 한다.

④ O: 심층신경망은 입력층과 출력층 사이에 **여러 은닉층**을 둔 인공신경망이다. 각 층의 변환을 거쳐 데이터의 특징을 계층적으로 표현할 수 있다.

따라서 옳지 않은 설명은 ③이다.

NoteBase

문 3. 딥러닝에 대한 설명으로 옳지 않은 것은?

① 심층신경망을 기반으로 한다.

② Tensorflow는 딥러닝을 지원한다.

③ 랜덤 포레스트(random forest)는 인공신경망에 기반한 딥러닝의 한 모델이다.

④ CNN(Convolutional Neural Network)은 이미지 처리 분야 등에 활용되는 딥러닝 모델이다.

정답 3번

간단한 해설

랜덤 포레스트는 **의사결정나무 여러 개를 결합한 앙상블 모델**이다. 여러 층의 인공신경망을 사용하는 딥러닝과는 모델의 기본 구조가 다르다.

선지별 해설

① (O) 딥러닝은 여러 층으로 구성된 **심층 인공신경망**을 이용한다. 각 층이 입력을 변환하면서 단순한 특징부터 복잡한 특징까지 계층적으로 학습할 수 있다.

② (O) TensorFlow는 기계학습 모델의 구성과 학습 등에 사용하는 소프트웨어 프레임워크이다. 인공신경망을 구성하고 학습시키는 기능을 제공하므로 딥러닝을 지원한다.

③ (X) 랜덤 포레스트는 **여러 의사결정나무의 예측을 결합하는 앙상블 기법**이다. 분류에서는 투표, 회귀에서는 예측값의 평균 등을 이용하며, 인공신경망을 기반으로 한 딥러닝 모델이 아니다.

④ (O) CNN은 합성곱 연산으로 입력의 국소적인 특징을 추출하는 신경망이다. 이미지의 공간적 패턴을 학습하는 데 적합하여 이미지 분류·인식 등의 분야에서 활용되는 딥러닝 모델이다.

NoteBase

문 4. 다음은 입력 조합에 대응하는 최소항(minterm)을 보여주는 표이다. ㉠~㉣에 들어갈 부울식으로 옳은 것은?

입력		최소항
A	B	
0	0	(㉠)
0	1	(㉡)
1	0	(㉢)
1	1	(㉣)

① ㉠: AB, ㉡: $A'B'$, ㉢: $A'B$, ㉣: AB'

② ㉠: AB', ㉡: AB, ㉢: $A'B'$, ㉣: $A'B$

③ ㉠: $A'B$, ㉡: AB', ㉢: AB, ㉣: $A'B'$

④ ㉠: $A'B'$, ㉡: $A'B$, ㉢: AB', ㉣: AB

정답 4번

간단한 해설

최소항은 해당 입력 조합에서만 1이 되도록 **0인 입력은 보수로, 1인 입력은 그대로** 써서 AND로 연결한다. 00, 01, 10, 11에 대응하는 항은 차례로 $A'B'$, $A'B$, AB', AB이다.

NoteBase

계산 과정

최소항은 모든 입력 변수를 한 번씩 포함하는 AND 항이며, 지정된 입력 조합에서만 1이 된다. 따라서 **입력이 0인 변수에는 보수를 붙이고, 1인 변수는 그대로** 사용한다. 기호 '는 보수를 뜻한다.

㉠: 입력이 $A=0,\ B=0$이면 두 변수를 모두 보수로 써서 $A'B'$가 된다. 이때 $A'=1,\ B'=1$이므로 AND 결과가 1이다.

㉡: 입력이 $A=0,\ B=1$이면 A만 보수로 써서 $A'B$가 된다.

㉢: 입력이 $A=1,\ B=0$이면 B만 보수로 써서 AB'가 된다.

㉣: 입력이 $A=1,\ B=1$이면 두 변수를 그대로 써서 AB가 된다.

따라서 ㉠부터 순서대로 $A'B',\ A'B,\ AB',\ AB$인 ④가 맞다.

NoteBase

문 5. 부호 있는(signed) 2진 정수의 2의 보수 표현법에 대한 설명으로 옳은 것은?

① 비트 수가 고정되었을 때 0에 대한 표현이 2가지이다.

② n비트로 표현할 수 있는 정수의 범위는 $-2^{(n-1)} \sim +2^{(n-1)}$이다.

③ 최상위 비트가 1이면서 양의 값을 가지는 정수가 존재한다.

④ 2진 정수 X에 대한 2의 보수는 X에 대한 1의 보수에 1을 더하여 구할 수 있다.

정답 4번

간단한 해설

고정된 비트 수에서 각 비트를 반전하여 1의 보수를 구한 뒤 **1을 더하면 2의 보수**가 된다. 따라서 ④가 옳다.

선지별 해설

① X: 2의 보수 표현에서 0은 모든 비트가 0인 **한 가지 형태**로만 표현된다. 양의 0과 음의 0이 따로 있는 것은 부호와 절댓값 표현이나 1의 보수 표현의 특징이다.

② X: n비트 2의 보수로 표현하는 정수의 범위는 -2^{n-1}부터 $2^{n-1}-1$까지이다. 예를 들어 8비트에서는 -128부터 127까지이므로 양의 최댓값에 1이 빠져야 한다.

③ X: 2의 보수 표현에서 최상위 비트가 1이면 음수이다. 최상위 비트의 가중치가 -2^{n-1}이므로 나머지 비트가 모두 1이어도 전체 값은 -1이다. 따라서 최상위 비트가 1인 양수는 없다.

④ O: **비트 반전 후 1을 더한다**. 예를 들어 8비트 00000101의 1의 보수는 11111010이고, 여기에 1을 더한 11111011이 2의 보수이다. 고정 비트 수를 넘는 자리올림은 버리며, 이 예에서는 +5의 부호를 바꾼 -5를 나타낸다.

NoteBase

문 6. 여러 개의 입력 신호 중 한 개를 선택하여 출력으로 내보내는 조합논리회로는?

① 디코더

② 전가산기

③ 멀티플렉서

④ 디멀티플렉서

정답 3번

간단한 해설

다수의 입력 중 하나를 선택하여 단일 출력으로 전달하는 회로는 멀티플렉서(MUX)이다.

개념 구분

멀티플렉서(MUX)는 선택 신호가 지정한 입력 하나를 출력으로 전달하는 회로이다. 핵심은 **여러 입력 중 하나를 골라 하나의 출력으로 보내는 것**이다. 예를 들어 4대1 멀티플렉서는 선택 신호의 값에 따라 네 입력 중 하나를 출력한다.

디멀티플렉서(DEMUX)는 방향이 반대이다. **하나의 입력을 여러 출력 중 선택된 하나로 전달**하므로, 여러 입력 중 하나를 선택한다는 문제의 설명과 다르다.

디코더는 이진 입력 코드에 대응하는 출력 선을 활성화하는 회로이다.

전가산기는 두 비트와 입력 자리올림을 더해 합과 출력 자리올림을 구하는 회로이므로, 둘 다 입력 신호를 선택하여 전달하는 멀티플렉서와 기능이 다르다.

따라서 해당하는 회로는 ③ 멀티플렉서이다.

NoteBase

문 7. 10진수 $132.8125_{(10)}$를 8진수로 변환한 결과로 옳은 것은?

① $204.64_{(8)}$

② $204.70_{(8)}$

③ $212.64_{(8)}$

④ $212.70_{(8)}$

정답 1번

간단한 해설

8진수 변환은 **정수부를 8로 나누고 소수부에 8을 곱하는** 방식으로 각각 구한다. 132는 $204_{(8)}$, 0.8125는 $0.64_{(8)}$이므로 결과는 $204.64_{(8)}$이다.

계산 과정

① 정수부 변환: 132를 8로 반복해서 나누고 나머지를 아래에서 위로 읽는다.
132 ÷ 8 = 16, 나머지 4
16 ÷ 8 = 2, 나머지 0
2 ÷ 8 = 0, 나머지 2
따라서 정수부는 $204_{(8)}$이다.

② 소수부 변환: 0.8125에 8을 반복해서 곱하고 생기는 정수 부분을 순서대로 읽는다.
0.8125 × 8 = 6.5 → 첫째 자리 6
0.5 × 8 = 4.0 → 둘째 자리 4
소수 부분이 0이 되었으므로 소수부는 $0.64_{(8)}$이다.

③ 결과 결합 및 검산: 정수부와 소수부를 합치면 $204.64_{(8)}$이다.
$2 \times 8^2 + 0 \times 8 + 4 + 6 \times 8^{-1} + 4 \times 8^{-2} = 132.8125$로 원래 값과 일치하므로 정답은 ①이다.

NoteBase

문 8. $+31.625_{(10)}$에 대한 IEEE 754 표준의 단일 정밀도(single precision) 표현을 16진수로 나타낸 것은?

① $01FB0000_{(16)}$

② $027D0000_{(16)}$

③ $41FD0000_{(16)}$

④ $427B0000_{(16)}$

정답 3번

간단한 해설

31.625를 $1.1111101_{(2)} \times 2^4$로 정규화하면 부호는 0, 저장 지수는 131이다. 숨은 맨 앞 1을 제외한 가수부와 결합하여 네 비트씩 읽으면 16진수 41FD0000이 된다.

계산 과정

단일 정밀도는 **부호 1비트, 지수 8비트, 가수부 23비트**로 구성된다. 양수이므로 부호 비트는 0이다.

정수부 31은 이진수로 11111이고, 소수부 0.625는 0.5 + 0.125이므로 0.101이다. 따라서
$31.625_{(10)} = 11111.101_{(2)} = 1.1111101_{(2)} \times 2^4$

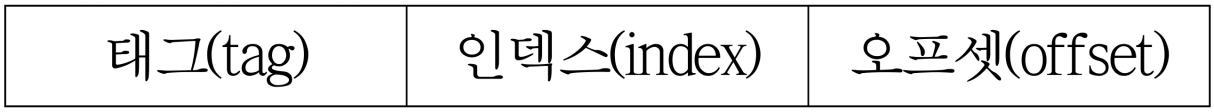
실제 지수 4에 단일 정밀도의 바이어스 127을 더한다.
$E = 4 + 127 = 131 = 10000011_{(2)}$
따라서 저장되는 지수 비트는 10000011이다.

정규화된 값의 맨 앞 1은 저장하지 않는다. 소수점 이하 1111101 뒤에 0을 채워 23비트로 만들면 가수부는 **11111010000000000000000**이다.

부호·지수·가수부를 이어 붙이면 01000001111111010000000000000000이다. 이를 네 비트씩 나누면
0100 0001 1111 1101 0000 0000 0000 0000
이므로 16진수 표현은 **41FD0000**이다. 따라서 정답은 ③이다.

NoteBase

문 9. 32비트 메모리 주소를 사용하는 컴퓨터 시스템에 용량이 128 KB이고 블록 크기가 32바이트인 2-way 집합-연관(set-associative) 캐시메모리가 있다. 캐시메모리와 관련하여 메모리 주소는 다음 그림과 같이 세 부분으로 나누어진다. 태그(tag)의 비트 수로 옳은 것은? (단, 1 KB는 1,024바이트이다.)



① 14

② 15

③ 16

④ 17

정답 3번

간단한 해설

블록 오프셋은 5비트이고, 2-way 캐시의 2,048개 집합을 고르는 인덱스는 11비트이다. 따라서 **태그는 32−11−5=16비트**이다.

계산 과정

주소는 그림처럼 태그·인덱스·오프셋으로 나뉜다. 바이트 단위 주소에서 오프셋은 블록 내부 바이트를, 인덱스는 **집합(set)을 선택**하고, 태그는 그 집합에 들어온 메모리 블록을 구별한다.

오프셋: 블록 크기가 $32 = 2^5$바이트이므로, 블록 내 32개 바이트를 구별하는 오프셋에는 $\log_2 32 = 5$비트가 필요하다.

집합 수와 인덱스: 캐시 용량은 $128 \times 1{,}024 = 131{,}072 = 2^{17}$바이트이다. 전체 블록 수는 $2^{17}/2^5 = 2^{12} = 4{,}096$개이다. 2-way는 **한 집합에 블록 2개가 들어간다는 뜻**이므로 집합 수는 $4{,}096/2 = 2{,}048 = 2^{11}$개이고, 인덱스는 11비트이다.

태그: 주소 전체 32비트에서 집합 인덱스와 블록 오프셋을 빼면 된다.
태그 비트 수 $= 32 - 11 - 5 = 16$비트이다. 두 way 중 하나를 선택하는 별도의 주소 비트를 더 빼지 않고, 선택된 집합의 태그들을 비교한다.

따라서 태그는 16비트로 ③과 일치한다.

NoteBase

문 10. 기억장치 계층구조(memory hierarchy)에 대한 설명으로 옳지 않은 것은?

① CPU에 가까울수록 기억장치의 용량이 증가한다.

② CPU에 가까울수록 기억장치로의 접근 시간이 빨라진다.

③ CPU에 가까울수록 기억장치의 비트당 비용이 증가한다.

④ 지역성의 원리(principle of locality)를 이용하여 구축된다.

정답 1번

간단한 해설

기억장치 계층은 일반적으로 CPU에 가까울수록 **빠르고 비싸지만 용량은 작게** 구성된다. 자주 쓰는 정보를 소량의 고속 기억장치에 두어 속도와 비용을 절충한다.

선지별 해설

① (X) CPU에 가까운 레지스터·캐시는 빠르지만 비트당 비용과 구현 공간의 제약이 커서 보통 용량이 작다. 주기억장치와 보조기억장치처럼 하위 계층으로 갈수록 더 큰 용량을 제공하므로, CPU에 가까울수록 용량이 증가한다는 설명은 반대이다.

② (O) 일반적으로 CPU에 가까운 상위 계층일수록 접근 지연이 짧다. 즉 레지스터나 캐시에서 데이터를 얻는 시간이 주기억장치나 보조기억장치에 접근하는 시간보다 짧은 방향으로 계층을 구성한다.

③ (O) 상위 계층은 높은 접근 속도를 제공하는 대신 **같은 비트 수를 저장하는 데 드는 비용**이 크다. 그래서 모든 데이터를 고속 기억장치에 저장하지 않고, 필요한 일부만 위쪽에 둔다.

④ (O) 시간적 지역성은 최근 사용한 정보를 곧 다시 사용할 가능성이 높은 성질이고, 공간적 지역성은 접근한 주소 주변을 이어서 사용할 가능성이 높은 성질이다. 계층구조는 이러한 성질을 활용하여 자주 필요한 정보가 상위 계층에 있도록 함으로써 평균 접근 시간을 줄인다.

NoteBase

문 11. 함수의 매개 변수, 자동 지역 변수 등의 데이터 저장에 사용되며 함수를 호출할수록 커지고 반환할수록 작아지는 프로세스의 메모리 영역으로 옳은 것은?

① 힙(heap) 영역
② 코드(code) 영역
③ 스택(stack) 영역
④ 데이터(data) 영역

정답 3번

간단한 해설

함수 호출 때 호출 정보를 쌓고 반환할 때 제거하는 영역은 **스택 영역**이다. 일반적인 메모리 구조에서 매개 변수, 자동 지역 변수, 복귀 주소 등을 스택 프레임으로 관리하므로 정답은 ③이다.

개념 구분

스택: 함수 호출과 반환에 따라 관리

함수가 호출되면 해당 호출에 필요한 스택 프레임이 추가되고, 함수가 반환하면 그 프레임이 제거된다. 나중에 호출된 함수가 먼저 반환되는 후입선출(LIFO) 구조이므로, 중첩 호출이 늘어나면 사용 공간이 커지고 반환하면 줄어든다.

힙: 실행 중 동적으로 할당하는 공간

힙의 할당 공간은 개별 함수의 호출·반환에 맞추어 자동으로 쌓였다가 제거되는 스택 프레임과는 다르다. 할당된 객체의 수명과 해제 정책에 따라 관리되므로, 함수 반환 후에도 필요한 데이터를 유지하는 데 사용할 수 있다.

코드 영역은 실행 명령을, 데이터 영역은 전역 변수나 정적 변수 등을 보관한다. 실제 구현에서는 일부 매개 변수나 지역 변수가 레지스터에 놓일 수 있지만, 문항이 묻는 전형적인 호출 관리 영역은 스택이다.

NoteBase

문 12. 파이프라인 위험(pipeline hazard)에 대한 설명으로 옳지 않은 것은?

① 파이프라인 위험으로 인해 파이프라인 지연(stall)이 발생할 수 있다.

② 분기(branch) 명령어는 제어 위험(control hazard)의 요인이 될 수 있다.

③ 데이터 전방전달(forwarding)을 사용하여 데이터 위험(data hazard)으로 인한 성능 저하를 모두 제거할 수 있다.

④ 서로 다른 명령어가 같은 클록(clock) 사이클에 동일한 하드웨어 자원을 사용하려고 할 때 구조적 위험(structural hazard)이 발생한다.

정답 3번

간단한 해설

전방전달은 이미 계산된 값을 뒤 명령어에 빨리 전달하여 지연을 줄이지만, **값이 생성되기 전의 대기까지 없애지는 못한다.** 메모리 읽기 직후 그 값을 사용하는 경우처럼 스톨이 남을 수 있으므로, 데이터 위험에 따른 성능 저하를 모두 제거한다는 설명은 옳지 않다.

선지별 해설

① O: 파이프라인 위험 때문에 다음 명령어가 예정된 단계로 진행할 수 없으면 일부 단계를 정지시키는 스톨을 삽입할 수 있다. 스톨은 올바른 실행을 보장하지만, 명령어 처리율을 낮추는 원인이 된다.

② O: 분기 명령어의 실행 결과나 목표 주소가 확정되기 전에는 다음에 가져올 명령어를 확정하기 어렵다. 이 때문에 명령어 인출을 기다리거나 잘못 가져온 명령어를 취소해야 하는 제어 위험이 발생할 수 있다.

③ X: 전방전달은 앞 명령어의 결과를 레지스터에 기록하기 전에 뒤 명령어로 직접 전달하여 많은 데이터 위험을 완화한다. 그러나 **아직 생성되지 않은 값까지 미리 전달할 수는 없다.** 예를 들어 일반적인 5단계 파이프라인에서 메모리 읽기 직후 그 값을 사용하는 명령어는, 읽은 값이 필요한 시점에 준비되지 않아 전방전달이 있어도 스톨이 필요할 수 있다.

④ O: 같은 사이클에 여러 명령어가 동시에 사용할 수 없는 동일 하드웨어 자원을 요구하면 구조적 위험이 발생한다. 예를 들어 하나의 메모리 포트를 명령어 인출과 데이터 접근이 동시에 요구하면 자원 충돌이 생길 수 있다.

NoteBase

문 13. 프로그래밍 언어에 대한 설명으로 옳지 않은 것은?

① Python은 인터프리터 방식의 대화형 언어이다.

② Java의 시초는 가전제품을 제어하기 위한 언어로 개발된 것이다.

③ C는 유닉스 운영체제의 작성을 위한 시스템 프로그래밍 언어로 설계된 것이다.

④ COBOL은 초보자도 쉽게 배울 수 있도록 만들어진 대화형 언어이며, 게임이나 자연어 처리 등 인공지능 분야에 사용된다.

정답 4번

간단한 해설

COBOL의 핵심 용도는 **회계·급여·금융 등 업무용 데이터 처리**이다. 초보자용 대화형 언어나 인공지능 분야의 대표 언어라는 설명과 구분한다.

선지별 해설

① (O) Python은 인터프리터를 통해 프로그램을 실행하며, 대화형 환경에서 명령을 입력하고 결과를 바로 확인할 수 있다. 내부적으로 바이트코드 등을 거치는 구현도 있지만, 제시된 인터프리터·대화형 언어라는 분류는 옳다.

② (O) Java의 전신인 Oak는 가전제품 등의 장치에서 동작하는 소프트웨어를 염두에 두고 개발되었다. 이후 여러 환경에서 실행할 수 있는 범용 언어로 발전했다.

③ (O) C는 유닉스 운영체제 개발과 밀접하게 관련되어 발전한 **시스템 프로그래밍 언어**이다. 메모리 등 저수준 자원을 다룰 수 있으면서도 이식성 있는 프로그램 작성에 유리하다.

④ (X) COBOL은 **업무용 데이터 처리**를 위해 만들어진 언어이다. 회계·급여·금융 거래처럼 많은 레코드를 처리하는 업무에 사용되어 왔다. 초보자용 대화형 언어라는 성격이나 인공지능 분야의 대표 언어라는 설명으로 COBOL을 규정하는 것은 적절하지 않다.

NoteBase

문 14. 다음 C 프로그램의 출력 결과로 옳은 것은?

```
#include <stdio.h>
int a = 100;
void function()
{

  int a = 200;
  printf("a의 값==>%d\n", a);

}

void main()
{

  function();
  printf("a의 값==>%d\n", a);

}
```

① a의 값 ==> 100
a의 값 ==> 100

② a의 값 ==> 100
a의 값 ==> 200

③ a의 값 ==> 200
a의 값 ==> 100

④ a의 값 ==> 200
a의 값 ==> 200

정답 3번

간단한 해설

function의 a는 전역 a와 별개인 지역 변수이다. **함수 안에서는 200, main에서는 전역 값 100**을 출력한다.

NoteBase

계산 과정

전역 변수: 함수 밖의 int a = 100은 전역 변수 a를 정의한다. 이 값은 프로그램 실행 중 유지되며, 같은 이름의 지역 변수에 가려지지 않는 곳에서 참조할 수 있다.

function 내부: int a = 200은 전역 a에 값을 대입하는 것이 아니라 **별개의 지역 변수 a를 새로 선언**한 것이다. 이 블록에서는 지역 a가 전역 a를 가리므로 function의 printf는 200을 출력한다.

main으로 복귀: function의 지역 a는 함수 실행이 끝나면 수명이 끝난다. main에는 지역 a가 없으므로 다음 printf는 **변경되지 않은 전역 a**를 참조하여 100을 출력한다.

출력 순서: main이 먼저 function을 호출하므로 function 내부의 200이 먼저 출력되고, 복귀 후 main의 100이 그다음 출력된다. 따라서 출력되는 a의 값은 차례로 200, 100이다.

따라서 제시된 출력 결과 중 ③과 일치한다.

NoteBase

문 15. 다음 C 프로그램의 출력 결과로 옳은 것은?

```
#include <stdio.h>
void main()
{

  int aa[]={3, 4, 5, 6, 7};
  int count;
  count = sizeof(aa) / sizeof(int);
  printf("%d", count);

}
```

① 4
② 5
③ 6
④ 7

정답 2번

간단한 해설

배열 전체의 크기를 원소 하나의 크기로 나누면 원소 수가 된다. aa에는 int형 원소가 5개 있으므로 sizeof(aa)/sizeof(int)의 값은 5이다.

NoteBase

계산 과정

배열 aa에는 3, 4, 5, 6, 7의 다섯 값이 들어 있으므로 int형 원소가 5개 있다. sizeof(aa)는 원소 하나가 아니라 **배열 전체의 바이트 크기**를 구한다.

int형 한 개의 크기를 s바이트라고 하면 sizeof(aa)는 $5s$, sizeof(int)는 s이다. 따라서 count에 저장되는 값은 $5s \div s = 5$이다. int형이 반드시 4바이트라고 가정할 필요는 없다.

printf는 정수 count의 값 5를 출력하므로 ②가 맞다. 여기서 aa는 실제 배열이므로 sizeof로 전체 크기를 얻을 수 있으며, 포인터 변수의 크기를 원소 수로 오해해서는 안 된다.

NoteBase

문 16. CPU 스케줄링에서 디스패치(dispatch)에 대한 설명으로 옳은 것은?

① 메모리에 적재되어 프로세스 제어 블록(PCB)을 할당받은 상태이다.

② 준비 상태의 프로세스가 스케줄링되어 CPU를 점유하는 실행 상태로 전환되는 것이다.

③ CPU가 프로그램을 수행하는 동안 컴퓨터 시스템의 내부나 외부에서 발생하는 예기치 못한 사건이다.

④ CPU의 지정된 할당 시간을 모두 사용한 프로세스가 다른 프로세스를 위해 다시 준비 완료 상태로 되돌아가는 것이다.

정답 2번

간단한 해설

디스패치는 준비 큐에서 선택된 프로세스에 CPU를 넘겨 **준비 상태에서 실행 상태로 전환**시키는 것이다. 따라서 정답은 ②이다.

개념 구분

디스패치: 준비 → 실행

스케줄러가 다음에 실행할 프로세스를 선택하면 디스패처가 그 프로세스로 CPU 제어를 넘긴다. 필요한 문맥을 복원하고 실행을 재개하는 절차를 통해 준비 상태의 프로세스가 실행 상태가 된다.

시간 할당량 만료: 실행 → 준비

실행 중인 프로세스가 정해진 CPU 사용 시간을 모두 쓰면 선점되어 준비 큐로 돌아갈 수 있다. 이는 ④의 설명이며 디스패치와 상태 전이의 방향이 반대이다.

③은 실행 중 발생한 사건을 처리하도록 CPU의 제어 흐름을 바꾸는 인터럽트에 관한 설명이다. ①처럼 메모리 적재와 PCB 할당을 언급하는 것만으로는 준비 상태의 프로세스에 CPU를 넘긴다는 디스패치의 핵심을 설명하지 못한다.

NoteBase

문 17. 다음 표는 단일 CPU에 진입한 프로세스의 도착 시간과 처리하는 데 필요한 실행 시간을 나타낸 것이다. 규정 시간량이 10인 라운드 로빈 알고리즘으로 스케줄링할 때, 평균 대기 시간으로 옳은 것은? (단, 프로세스간 문맥교환에 따른 오버헤드는 무시하며, 주어진 5개의 프로세스 외에 처리할 다른 프로세스는 없다고 가정한다.)

프로세스	도착 시간	실행 시간
P_1	0	10
P_2	1	20
P_3	2	6
P_4	3	4
P_5	4	14

① 17.6

② 21.2

③ 26.5

④ 35.3

정답 2번

간단한 해설

시간량 10을 사용한 뒤에도 작업이 남은 **P_2와 P_5를 준비 큐의 뒤로 다시 보낸다**. 각 대기 시간을 완료 시각에서 도착 시각과 실행 시간을 빼서 구하면 평균은 21.2이다.

NoteBase

계산 과정

라운드 로빈은 준비 큐의 앞에서부터 **최대 10만큼 실행**하고, 남은 실행 시간이 있으면 큐의 뒤로 보낸다.
시간량을 다 쓰기 전에 끝나면 즉시 다음 프로세스를 실행한다.

P_1은 0⊠10에 실행을 마친다. 그동안 P_2, P_3, P_4, P_5가 순서대로 도착한다. 이후 실행 구간은
P_2: 10⊠20 → P_3: 20⊠26 → P_4: 26⊠30 → P_5: 30⊠40 → P_2: 40⊠50 → P_5: 50⊠54
이다. P_2와 P_5는 첫 실행 뒤 남은 시간이 있어 큐 뒤로 이동한다.

대기 시간 = 완료 시각 − 도착 시각 − 총 실행 시간으로 구한다.

P_1: 10 − 0 − 10 = 0

P_2: 50 − 1 − 20 = 29

P_3: 26 − 2 − 6 = 18

P_4: 30 − 3 − 4 = 23

P_5: 54 − 4 − 14 = 36

평균 대기 시간은
$\frac{0+29+18+23+36}{5} = \frac{106}{5} = 21.2$
이다. 따라서 정답은 ② **21.2**이다.

NoteBase

문 18. 다음과 같은 특징을 갖는 메모리 할당 방법으로 옳은 것은?

· 외부 단편화 발생
· 비연속 메모리 할당 방법
· 메모리 크기를 동적으로 분할
· 메모리를 프로세스 관점에서 지원하는 관리 방법

① 페이징
② 스와핑
③ 요구 페이징
④ 세그먼테이션

정답 4번

간단한 해설

세그먼테이션은 프로세스를 가변 크기의 논리적 단위로 분할하여 메모리를 관리한다. 세그먼트들을 서로 떨어진 위치에 둘 수 있지만, 가변 크기의 연속 공간을 배정하는 과정에서 외부 단편화가 생길 수 있다.

핵심 풀이

프로세스 관점의 가변 크기 분할: 세그먼테이션은 프로그램을 코드·데이터·스택 같은 논리적 단위로 나눈다. 각 세그먼트의 크기는 서로 다를 수 있으므로, 고정 크기로 나누는 페이징과 구별된다.

비연속 할당과 외부 단편화: 서로 다른 세그먼트를 메모리의 떨어진 위치에 배치할 수 있다. 다만 순수 세그먼테이션에서는 한 세그먼트마다 연속된 공간이 필요하므로, 가변 크기 공간을 할당·해제하다 보면 빈 공간이 흩어지는 외부 단편화가 생길 수 있다.

다른 방식과의 구별: 페이징은 고정 크기 페이지와 프레임을 사용하며 일반적으로 외부 단편화를 없애는 대신 내부 단편화가 생길 수 있다. 요구 페이징은 필요한 페이지를 참조할 때 메모리에 적재하는 방식이고, 스와핑은 프로세스 등을 메모리와 보조기억장치 사이에서 이동시키는 기법이다.

NoteBase

문 19. CPU를 거치지 않고 입출력 장치와 메모리 간에 데이터를 전송하는 입출력 방법으로 옳은 것은?

① 폴링(polling)

② DMA(Direct Memory Access)

③ 프로그램 입출력(programmed I/O)

④ 인터럽트 기반 입출력(interrupt driven I/O)

정답 2번

간단한 해설

DMA 제어기나 장치가 메모리에 직접 접근하여 데이터를 전송하는 방식은 DMA이다. CPU가 매 데이터의 전송을 중계하는 방식과 다르다.

개념 구분

DMA는 입출력 장치와 메모리 사이의 데이터 이동을 DMA 제어기나 DMA 기능을 가진 장치가 수행하는 방식이다. 핵심은 **CPU가 각 데이터를 읽고 써서 중계하지 않는 것**이며, CPU의 전송 처리 부담을 줄인다.

CPU가 전혀 관여하지 않는다는 뜻은 아니다. CPU는 전송할 메모리 주소·크기·방향 등을 설정하여 전송을 시작할 수 있고, 완료 후 인터럽트 등으로 결과를 확인한다. **전송의 설정·완료 처리와 실제 데이터 이동**을 구별해야 한다.

프로그램 입출력에서는 CPU가 명령으로 데이터를 옮기며, 폴링은 CPU가 장치의 준비 상태를 반복 확인하는 방식이다. 인터럽트 기반 입출력은 장치가 준비되었음을 알려 대기를 줄이지만, 전형적인 비교에서는 CPU가 입출력 처리 루틴을 통해 데이터를 옮긴다. 준비 통지를 인터럽트로 한다는 사실만으로 DMA와 같아지는 것은 아니다.

따라서 해당하는 입출력 방법은 ② DMA이다.

NoteBase

문 20. 다음 알고리즘의 시간복잡도를 빅오(O) 표기법으로 표현한 것은? (단, $n > 0$)

```
void complexity(int n) {
      int i, j, count = 0;
      for (i = 1; i <= n; i++) {
            for (j = 1; j <= i; j++)
                  printf(" %d", ++count);
            printf ("\n");
      }
}
```

① $O(n)$

② $O(n^2)$

③ $O(\log_2 n)$

④ $O(n\log_2 n)$

정답 2번

간단한 해설

안쪽 반복은 i가 증가할 때마다 1회, 2회, …, n회 실행된다. 총 횟수 $n(n+1)/2$의 최고차항이 **이차항**이므로 시간복잡도는 $O(n^2)$이다.

NoteBase

계산 과정

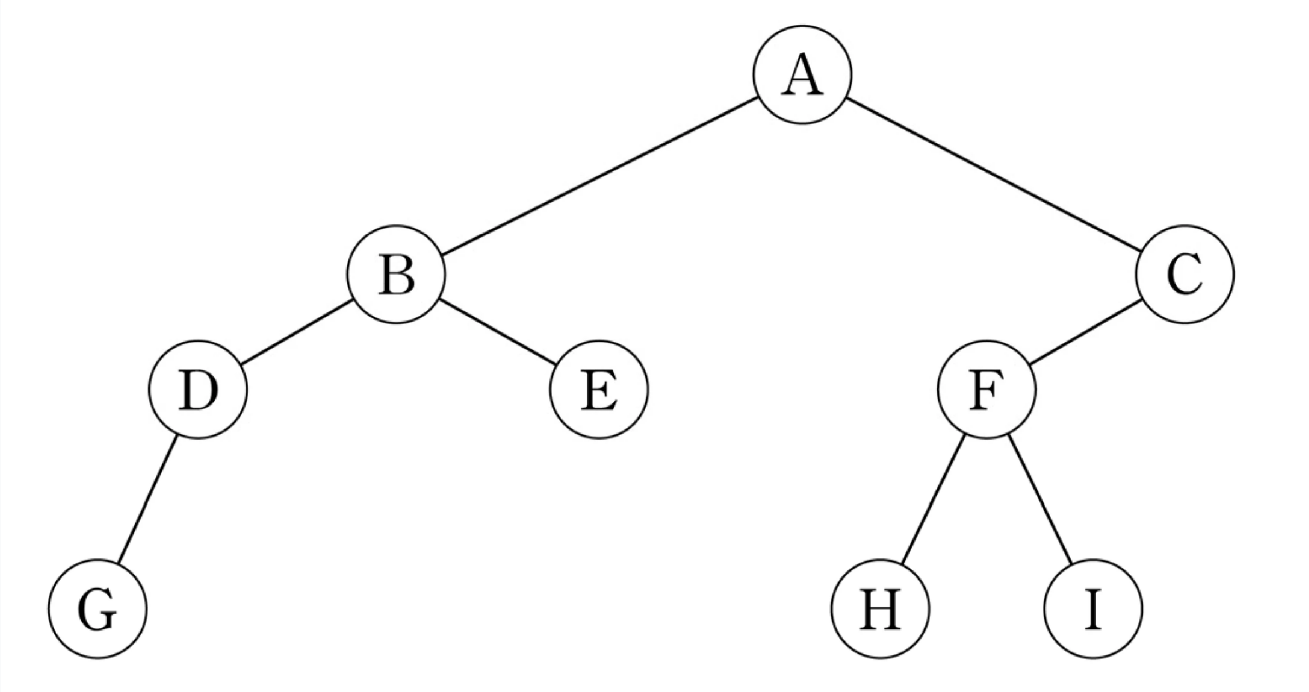
바깥 반복문은 i를 1부터 n까지 증가시키므로 n회 실행된다. 안쪽 반복문은 매번 j를 1부터 i까지 증가시키므로, 해당 바깥 반복에서 i회 실행된다.

안쪽 출력문의 총 실행 횟수는 $1 + 2 + \cdots + n = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$이다. 안쪽 반복이 매번 n회인 것은 아니지만, 반복 횟수의 합은 **이차식**으로 증가한다.

통상적인 기본 연산 횟수 분석에서 반복문 제어와 정수 출력 한 번을 상수 시간 작업으로 센다. 바깥의 개행 출력은 n회로 선형 항을 더할 뿐이므로, 상수 계수와 낮은 차수 항을 제외한 시간복잡도는 $O(n^2)$이며 ②가 맞다.

NoteBase

문 21. 다음 이진 트리를 중위 순회(inorder traversal)할 때 마지막 방문 노드로 옳은 것은?



① C

② E

③ G

④ I

정답 1번

간단한 해설

중위 순회에서는 왼쪽 서브트리를 처리한 뒤 현재 노드를 방문한다. C의 왼쪽 F 서브트리를 모두 방문한 뒤 **오른쪽 자식이 없는 C를 마지막으로 방문**한다.

NoteBase

계산 과정

중위 순회는 각 노드에서 **왼쪽 서브트리 → 현재 노드 → 오른쪽 서브트리**의 규칙을 재귀적으로 적용한다. 루트 A에서는 왼쪽 B 서브트리를 모두 방문한 뒤 A를 방문하고, 마지막으로 오른쪽 C 서브트리를 방문한다.

왼쪽 서브트리: B의 왼쪽 D는 왼쪽 자식 G를 먼저 방문하므로 G → D가 된다. 이후 B를 방문하고 오른쪽 E를 방문하여 B 서브트리의 순서는 G → D → B → E이다.

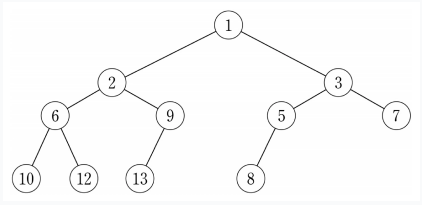
오른쪽 서브트리: C의 왼쪽 자식 F는 왼쪽 H, 자기 자신 F, 오른쪽 I 순서로 방문한다. 그다음 C를 방문하는데 **C에는 오른쪽 자식이 없으므로**그곳에서 전체 순회가 끝난다.

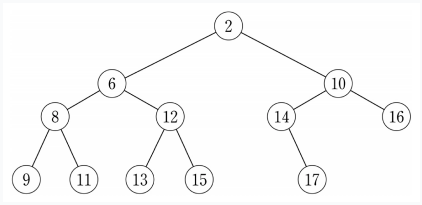
전체 확인: G → D → B → E → A → H → F → I → C 순서이므로, 마지막에 방문하는 노드는 C이다. 가장 깊은 잎 I가 아니라 오른쪽 서브트리의 루트 C가 I 다음에 방문된다는 점에 주의한다.

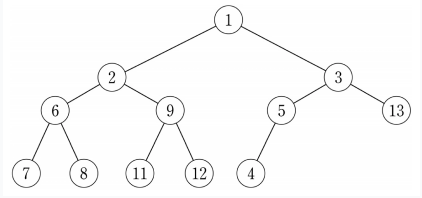
따라서 마지막 방문 노드는 C로 ①과 일치한다.

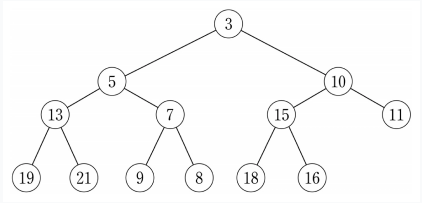
NoteBase

문 22. 최소 힙(min heap)에 해당하는 것은?

① 

② 

③ 

④ 

정답 4번

간단한 해설

최소 힙은 **완전 이진 트리**이면서 모든 부모의 값이 자식의 값보다 작거나 같아야 한다. 두 조건을 모두 만족하는 것은 ④이다.

NoteBase

선지별 해설

① X: 부모의 값은 자식보다 작지만 완전 이진 트리가 아니다. 마지막 층에서 노드 9의 오른쪽 자리가 비어 있는데 그보다 오른쪽에 있는 노드 5의 왼쪽 자식 8이 존재한다. 완전 이진 트리는 마지막 층을 **왼쪽부터 빈자리 없이** 채워야 한다.

② X: 노드 14는 왼쪽 자식이 없고 오른쪽 자식 17만 갖는다. 마지막 층에 왼쪽 빈자리가 생기므로 완전 이진 트리 조건을 위반한다. 값의 대소관계가 맞더라도 최소 힙이 아니다.

③ X: 트리의 모양은 완전 이진 트리이지만 **부모 5가 자식 4보다 크다**. 최소 힙에서는 모든 부모 값이 자식 값보다 작거나 같아야 하므로 힙 순서 조건을 위반한다.

④ O: 마지막 층이 왼쪽부터 연속하여 채워진 완전 이진 트리이다. 또한 3≤5,10; 5≤13,7; 10≤15,11; 13≤19,21; 7≤9,8; 15≤18,16으로 모든 부모·자식 관계가 조건을 만족한다. 형제 노드 사이의 값이 왼쪽부터 오름차순일 필요는 없다.

NoteBase

문 23. 다음과 같이 선언된 배열을 행 우선(row major) 순서로 저장하는 경우 원소 arr[3][3]의 주소로 옳은 것은? (단, char의 크기는 1바이트이고, 배열 arr의 시작 주소는 3,000이라고 가정한다.)

```
char arr[20][15];
```

① 3,048

② 3,096

③ 3,192

④ 3,384

정답 1번

간단한 해설

행 우선 저장에서 arr[3][3] 앞에는 **3개 행의 45개 원소와 같은 행의 3개 원소**가 있다. 원소당 1바이트이므로 시작 주소 3,000에 48을 더해 3,048을 얻는다.

계산 과정

행 우선 저장은 한 행의 원소들을 차례로 저장한 뒤 다음 행으로 넘어간다. arr[20][15]에서는 **한 행에 15개 원소**가 있고, C 배열의 인덱스는 0부터 시작한다.

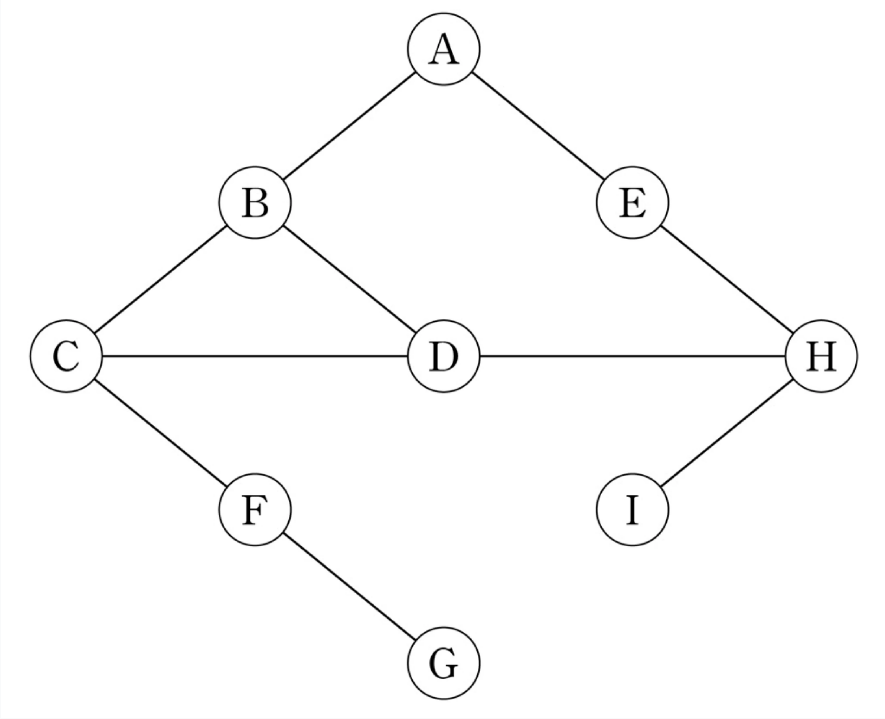
arr[3][3] 앞에는 0·1·2행의 3개 행과, 3행 안의 0·1·2열 원소가 있다. 따라서 앞선 원소 수는
$3 \times 15 + 3 = 48$
개이다.

한 원소가 1바이트이므로 시작 주소에서의 바이트 간격도 48이다.
$\text{addr}(arr[3][3]) = 3000 + (3 \times 15 + 3) \times 1 = 3048$

따라서 주소는 **3,048**이며 정답은 ①이다. 행 번호에 곱하는 값은 전체 행 수 20이 아니라 **한 행의 열 수 15**이다.

NoteBase

문 24. 다음 그래프의 정점 A에서부터 시작하여 너비우선탐색을 수행할 때, 방문 순서로 옳은 것은? (단, 방문하지 않은 인접 정점이 2개 이상인 경우 알파벳 오름차순으로 방문한다.)



① A → B → C → D → E → F → G → H → I

② A → B → C → D → H → E → I → F → G

③ A → B → E → C → D → F → G → H → I

④ A → B → E → C → D → H → F → I → G

정답 4번

간단한 해설

너비우선탐색은 **먼저 발견한 정점부터 큐의 순서대로 처리**한다. A의 이웃 B, E를 먼저 방문한 뒤 C, D, H를 처리하므로, 방문 순서는 A → B → E → C → D → H → F → I → G이다.

NoteBase

계산 과정

① A에서 시작: 너비우선탐색은 큐를 사용하여 먼저 발견한 정점부터 처리한다. A를 방문하고 인접한 B, E를 알파벳순으로 큐에 넣는다. 큐의 앞부터 B, E이다.

② 첫 번째 이웃 처리: B를 꺼내 아직 발견하지 않은 C, D를 넣으면 큐는 E, C, D가 된다. 이어서 E를 꺼내 H를 넣으면 큐는 C, D, H이다.

③ 다음 정점 처리: C를 꺼내 F를 넣으면 큐는 D, H, F이다. D의 인접 정점은 이미 방문했거나 큐에 들어 있으므로 다시 넣지 않는다. H를 꺼내 I를 넣으면 큐는 F, I이다.

④ 나머지 처리: F를 꺼내 G를 넣으면 큐는 I, G가 된다. 이후 I와 G를 차례로 방문한다. 전체 방문 순서는 **A → B → E → C → D → H → F → I → G**이므로 정답은 ④이다.

NoteBase

문 25. DIKW 계층 구조에서 관찰 또는 측정으로 수집된 사실이나 값을 지칭하는 용어는?

① 데이터(data)

② 지혜(wisdom)

③ 지식(knowledge)

④ 정보(information)

정답 1번

간단한 해설

관찰·측정으로 얻은 개별 사실이나 값은 데이터이다. 이를 정리하고 맥락을 부여해 의미 있게 만든 정보와 구별한다.

개념 구분

데이터는 관찰·측정 등을 통해 수집한 개별 사실이나 값이다. 아직 특정 목적에 맞게 의미를 해석하거나 관계를 정리하기 전의 기초 자료에 해당하므로, 질문의 용어는 ① 데이터이다.

정보는 데이터를 정리하고 맥락을 부여하여 의미 있게 만든 것이다. 예를 들어 여러 시각에 측정한 온도값들이 데이터라면, 이를 비교해 '오늘 오후 기온이 오전보다 높았다'고 정리한 내용은 정보로 볼 수 있다.

지식은 정보 사이의 관계나 규칙을 이해하여 활용할 수 있게 된 것이고, 지혜는 그러한 이해에 경험과 판단을 더하여 상황에 맞는 선택을 하는 것이다. 따라서 DIKW에서는 데이터, 정보, 지식, 지혜를 구별한다.

NoteBase

문 26. 관계 데이터 모델에서 릴레이션의 특징으로 옳지 않은 것은?

① 속성 값으로 다중 값을 허용할 수 있다.

② 하나의 릴레이션에는 동일한 튜플(tuple)이 존재할 수 없다.

③ 하나의 릴레이션에서 속성 사이의 순서는 바뀌어도 상관없다.

④ 하나의 릴레이션에서 튜플 사이의 순서는 바뀌어도 상관없다.

정답 1번

간단한 해설

전통적인 관계 모델에서 한 속성 값은 **단일한 원자 값**이어야 한다. 다중 값을 한 속성에 담는다는 설명은 릴레이션의 특징에 맞지 않는다.

선지별 해설

이 문제는 구현상의 SQL 테이블이 아니라 전통적인 관계 데이터 모델의 릴레이션을 묻는다. 릴레이션은 정해진 속성들로 이루어진 튜플의 집합이며, 각 속성 값은 원자 값으로 취급한다.

① X: 하나의 튜플에서 한 속성에는 해당 도메인의 **단일한 원자 값**이 들어가야 한다. 한 칸에 여러 값을 모아 넣는 다중 값이나 반복 그룹을 허용한다는 설명은 전통적인 릴레이션의 원자성에 어긋난다.

② O: 릴레이션은 튜플의 **집합**이므로 모든 속성 값이 같은 중복 튜플은 존재하지 않는다. 특정 속성 값 하나가 같다는 것과 전체 튜플이 같다는 것은 구별해야 한다.

③ O: 속성은 이름으로 식별하므로 **속성의 표시 순서**를 바꾸어도 릴레이션의 논리적 의미는 달라지지 않는다. 단, 각 속성과 그 값의 대응은 유지되어야 한다.

④ O: 튜플들은 집합의 원소이므로 **튜플 사이에 고정된 순서가 없다**. 행의 표시 위치를 바꾸어도 같은 튜플 집합이면 같은 릴레이션이다.

따라서 릴레이션의 특징으로 옳지 않은 것은 ①이다.

NoteBase

문 27. 데이터 관리를 위한 파일 시스템의 문제점으로 옳지 않은 것은?

① 응용 프로그램을 개발하기가 쉽지 않다.

② 응용 프로그램이 데이터 파일에 종속적이다.

③ 같은 내용의 데이터는 단일 파일에만 저장해야 한다.

④ 데이터 파일에 대한 동시 공유, 보안, 회복 기능이 부족하다.

정답 3번

간단한 해설

전통적인 파일 처리 방식에서는 **같은 데이터가 여러 파일에 중복되고 값이 불일치**할 수 있다. 같은 내용을 단일 파일에만 저장해야 하는 제약이 있다는 설명은 실제 문제점과 반대이다.

선지별 해설

① (O) 전통적인 파일 처리 방식에서는 응용 프로그램마다 파일의 구조와 읽기·쓰기·검사 절차를 직접 처리해야 한다. 공통 데이터 관리 기능을 각각 구현해야 하므로 개발과 유지보수 부담이 커진다.

② (O) 프로그램이 파일의 레코드 형식이나 저장 구조를 알고 접근하므로 **프로그램과 데이터 사이의 종속성**이 높다. 파일 구조가 바뀌면 이를 이용하는 프로그램도 함께 수정해야 할 수 있다.

③ (X) 파일 시스템이 같은 데이터를 반드시 한 파일에만 저장하도록 강제하는 것은 아니다. 오히려 여러 응용 프로그램의 파일에 **같은 데이터가 중복 저장**되기 쉽고, 일부만 수정하면 서로 다른 값이 남는 불일치가 생길 수 있다는 것이 문제점이다.

④ (O) DBMS와 비교한 전통적 파일 처리 방식에서는 여러 프로그램의 동시 접근을 조정하거나 데이터 단위의 권한과 장애 복구를 일관되게 관리하기 어렵다. 운영체제의 기본 파일 보안 등이 전혀 없다는 뜻이 아니라, 통합된 데이터 관리 기능이 부족하다는 의미이다.

NoteBase

문 28. 음성 데이터의 주파수를 4 kHz 이내로 제한하여 이를 Nyquist 대역폭에 따라 디지털로 변조하여 전송하고자 한다. 각 샘플(sample)을 4비트로 표현한다면, 최대 전송률로 옳은 것은?

① 32 kbps

② 64 kbps

③ 128 kbps

④ 256 kbps

정답 1번

간단한 해설

4 kHz에 나이퀴스트의 2배 기준을 적용하면 **초당 8,000샘플**이다. 샘플당 4비트를 곱하면 **32,000 bps, 즉 32 kbps**가 된다.

계산 과정

주어진 대역폭은 $B = 4\text{kHz} = 4{,}000\text{Hz}$이다. 나이퀴스트 기준에서 이 대역폭에 대응하는 기본 표본화율은 $2B = 8{,}000$샘플/초이다.

각 샘플을 4비트로 표현하므로 전송할 비트율은 $8{,}000 \times 4 = 32{,}000\text{bps} = 32\text{kbps}$이다. 따라서 ①이 맞다.

같은 수치는 이상적인 무잡음 채널의 나이퀴스트 식 $R_{\max} = 2B\log_2 M$으로도 얻을 수 있다. 4비트에 대응하는 신호 단계 수를 $M = 2^4 = 16$으로 두면 $R_{\max} = 2 \times 4{,}000 \times \log_2 16 = 32{,}000\text{bps}$이다. 표본화 이론의 2배 기준은 최소 표본화율에 대한 것이므로, 실제로 더 높은 표본화율을 선택할 수 없다는 뜻은 아니다.

NoteBase

문 29. OSI 참조 모델의 전송 계층(transport layer) 프로토콜로 옳은 것은?

① ARP

② UDP

③ HDLC

④ HTTP

정답 2번

간단한 해설

UDP는 포트 번호로 응용 프로세스를 구분하는 전송 계층 프로토콜이다. ARP는 주소 결정, HDLC는 데이터 링크 제어, HTTP는 응용 계층의 요청·응답을 담당한다.

선지별 해설

① X: ARP는 같은 링크에서 IPv4 주소에 대응하는 MAC 주소를 알아내는 주소 결정 프로토콜이다. 링크에서 실제 프레임을 전달할 목적지 주소를 찾는 기능이며, 응용 프로세스 사이의 전송을 담당하는 프로토콜은 아니다.

② O: UDP는 **전송 계층의 비연결형 프로토콜**이다. 포트 번호를 이용하여 송수신 응용 프로세스를 구분하며, 연결 설정 없이 데이터그램을 전송한다. UDP 자체는 재전송이나 순서 보장 기능을 제공하지 않는다.

③ X: HDLC는 데이터 링크 계층의 비트 지향 프로토콜이다. 프레임을 구성하고 링크상의 데이터 전달을 제어하는 기능을 수행하므로 전송 계층과 구별된다.

④ X: HTTP는 웹 자원에 대한 요청과 응답을 규정하는 응용 계층 프로토콜이다. 하위 전송 기능을 이용하는 응용 프로토콜이지, 그 자체가 전송 계층 프로토콜은 아니다.

NoteBase

문 30. 다음에서 ㉠~㉢에 해당하는 내용으로 모두 옳은 것은?

㉠
· 데이터 링크 계층에서 생성된다.
· LAN 내의 수신자와 송신자를 특정할 수 있다.
· 네트워크 인터페이스마다 부여되는 48비트 길이의 주소이다.

㉡
· 네트워크 계층에서 생성된다.
· 인터넷에 연결된 기기를 식별할 수 있는 정보이다.
· 32비트로 표현되며, 8비트씩 구분하여 0~255 범위 안에 있는 네 개의 10진수로 표기되기도 한다.

㉢
· 전송 계층에서 생성된다.
· 프로세스를 식별하기 위한 정보이다.
· 0~65,535의 범위를 가지는 16비트로 표현되는 정수이다.

① ㉠: Port 번호, ㉡: IPv4 주소, ㉢: MAC 주소
② ㉠: MAC 주소, ㉡: Port 번호, ㉢: IPv4 주소
③ ㉠: MAC 주소, ㉡: IPv4 주소, ㉢: Port 번호
④ ㉠: Port 번호, ㉡: MAC 주소, ㉢: IPv4 주소

정답 3번

간단한 해설

계층과 비트 수를 대응시키면 ㉠ **MAC 주소(데이터 링크 계층, 48비트)**, ㉡ **IPv4 주소(네트워크 계층, 32비트)**, ㉢ **Port 번호(전송 계층, 16비트)**이다. 따라서 정답은 ③이다.

NoteBase

대응 관계 해설

㉠ **MAC 주소**: 이더넷 LAN에서 프레임을 주고받는 네트워크 인터페이스를 식별하는 48비트 주소이다. 같은 링크에서 송신자와 수신자를 구분하므로 데이터 링크 계층과 대응한다. Port 번호는 전송 계층의 16비트 식별 정보이므로 ㉠에 맞지 않는다.

㉡ **IPv4 주소**: 네트워크 계층에서 사용하는 32비트 논리 주소이다. 8비트씩 네 부분으로 나누어 점으로 연결한 10진수로 표기하며, 각 부분의 범위는 0부터 255까지이다. 라우터는 IP 주소를 이용하여 패킷의 전달 경로를 결정한다.

㉢ **Port 번호**: TCP나 UDP 등 전송 계층에서 통신 종단의 서비스나 응용 프로그램을 구분하는 16비트 번호이다. 표현 범위는 0부터 65,535까지이며, 실제 통신에서는 IP 주소 및 전송 프로토콜 등과 함께 사용한다.

따라서 ㉠ MAC 주소, ㉡ IPv4 주소, ㉢ Port 번호의 대응 관계를 제시한 ③이 모두 옳다.

NoteBase

문 31. 주로 2.4 GHz, 5 GHz 등의 대역을 사용하며, 버전에 따라 사용되는 주파수뿐만 아니라 전송 속도, 대역폭, 전송 가능한 거리, 변조 방식 등이 달라지는 대표적인 무선랜 표준으로 옳은 것은?

① IEEE 802.3

② IEEE 802.4

③ IEEE 802.5

④ IEEE 802.11

정답 4번

간단한 해설

IEEE 802.11은 무선랜(WLAN) 표준 계열이다. 이더넷의 IEEE 802.3, 토큰 버스·토큰 링 표준과 구별한다.

개념 구분

문제의 결정적 단서는 **무선랜(WLAN) 표준**이라는 점이다. IEEE 802.11 계열은 무선랜의 매체 접근 제어와 물리 계층을 규정하며, Wi-Fi 기술의 기반이 된다. 2.4 GHz·5 GHz 등의 대역을 사용하는 규격들이 이 계열에 속한다.

802.11에는 여러 세부 규격이 있으므로 사용하는 주파수 대역, 채널 폭, 변조 방식과 지원 속도 등이 달라질 수 있다. 문제는 특정 세부 버전의 속도를 계산하는 것이 아니라 **이들을 포괄하는 표준 계열**을 식별하는 것이다.

IEEE 802.3은 이더넷, IEEE 802.4는 토큰 버스, IEEE 802.5는 토큰 링에 해당한다. 이들은 제시된 무선랜 표준인 IEEE 802.11과 구별된다.

따라서 대표적인 무선랜 표준은 ④ IEEE 802.11이다.

NoteBase

문 32. QAM(Quadrature Amplitude Modulation)에서 변조되는 두 가지 특성으로 옳은 것은?

① 위상 - 진폭

② 진폭 - 주파수

③ 주파수 - 위상

④ 대역폭 - 주파수

정답 1번

간단한 해설

QAM은 두 직교 반송파 성분을 조합하여 **합성 신호의 진폭과 위상**으로 정보를 나타낸다. 이름의 진폭만 보고 주파수 변조와 혼동하지 않는다.

핵심 풀이

QAM은 **직교 진폭 변조**이다. 서로 위상이 90° 다른 두 반송파 성분을 이용하고, 각 성분의 크기를 조절하여 합성 신호를 만든다.

두 직교 성분을 합성하면 신호점마다 결과 신호의 **진폭과 위상**이 달라질 수 있다. 이 진폭·위상의 조합에 정보를 대응시키므로 문제에서 묻는 두 특성은 위상과 진폭이며 정답은 ①이다.

주파수 변화로 정보를 나타내는 방식과 구분해야 한다. QAM은 반송파 주파수를 여러 값으로 바꾸어 구별하는 방식이 아니며, 대역폭도 이 문항에서 묻는 두 변조 특성에 해당하지 않는다.

NoteBase

문 33. 파이프라이닝 기법을 활용하여 여러 세그먼트를 전송하고, 도중에 잘못 전송된 세그먼트가 발생하면 해당 세그먼트부터 전부 다시 전송하는 ARQ 방식으로 옳은 것은?

① Go back N

② Stop and Wait

③ Fast Recovery

④ Selective Repeat

정답 1번

간단한 해설

오류가 난 지점부터 후속 미확인 데이터까지 함께 재전송하는 방식은 Go-Back-N ARQ이다. 오류가 난 데이터만 골라 다시 보내는 Selective Repeat와 구별한다.

개념 구분

Go-Back-N ARQ는 확인 응답을 매번 기다리지 않고 송신 윈도 크기 범위에서 여러 데이터를 연속 전송한다. 재전송이 필요해지면 문제가 된 데이터부터 그 뒤의 아직 확인되지 않은 데이터들을 함께 다시 보내므로, 질문의 설명은 ①에 해당한다.

Selective Repeat ARQ도 여러 데이터를 연속 전송하지만, 수신 측에서 정상적으로 받은 후속 데이터를 보관하고 오류·손실이 난 데이터만 선택적으로 재전송한다. **문제 지점 이후를 함께 재전송하는지, 필요한 것만 재전송하는지**가 두 방식의 핵심 차이이다.

Stop-and-Wait ARQ는 데이터 하나를 보낸 뒤 확인 응답을 기다리므로 여러 데이터를 연속으로 보내는 파이프라이닝 방식이 아니다.

Fast Recovery는 TCP의 혼잡 제어에서 손실 후 혼잡 윈도를 조절하는 절차로, 지문이 설명한 ARQ 방식의 이름과 구별된다.

NoteBase

문 34. 리눅스에서 로그인 계정 이름, 로그인한 환경, 로그인한 디바이스, 로그인한 셸(shell)의 프로세스 ID 등을 기록하는 로그로 옳은 것은?

① utmp
② syslog
③ secure
④ history

정답 1번

간단한 해설

utmp는 현재 로그인한 사용자와 세션의 정보를 기록한다. 계정 이름, 터미널, 로그인 관련 프로세스 ID 등이 주요 단서이다.

개념 구분

utmp는 시스템의 **현재 로그인 세션 정보**를 담는 기록이다. 사용자 이름, 터미널 장치, 원격 접속 호스트, 로그인 관련 프로세스 ID와 시각 등의 필드를 포함하므로 문제의 단서와 일치한다.

syslog는 시스템과 서비스의 다양한 이벤트를 기록하는 일반적인 로깅 체계 또는 그 로그를 가리킨다. secure는 일부 배포판에서 인증·보안 관련 메시지를 모으는 로그로 쓰인다. 이들은 **세션별 사용자·터미널·PID를 구조화한 utmp**와 구별된다.

history는 셸에서 입력한 명령의 이력을 다룬다. 로그인 사용자와 터미널 등의 세션 정보를 기록하는 것이 아니라, 사용자가 어떤 명령을 입력했는지를 확인하는 데 쓰인다.

따라서 해당하는 로그는 ① utmp이다.

NoteBase

문 35. 다음에서 설명하는 HTTP Request 메서드(method)로 옳은 것은?

메시지에 포함된 데이터를 지정한 URI(Uniform Resource Identifier) 장소에 해당 URI를 이름으로 하여 저장한다.

① GET
② PUT
③ DELETE
④ TRACE

정답 2번

간단한 해설

PUT은 지정한 URI의 자원을 요청 메시지의 내용으로 생성하거나 대체하도록 요청하는 메서드이다. 따라서 데이터를 보낼 대상 URI를 지정해 저장한다는 설명과 일치한다.

개념 구분

PUT: **지정한 URI의 자원 상태를 요청 내용으로 생성하거나 대체**하도록 요청한다. 클라이언트가 저장 대상을 URI로 지정하고 데이터를 보내는 것이 핵심이므로, 지문의 설명에 해당한다.

GET: 지정한 자원의 표현을 가져오는 조회 요청이다. 요청에 포함한 데이터로 대상 자원을 저장하거나 대체하는 PUT과는 목적이 다르다.

DELETE: 지정한 URI와 현재 기능 사이의 연결을 제거하도록 요청하는 메서드이다. 일반적으로 대상 자원을 삭제하는 요청으로 이해한다.

TRACE: 요청 메시지가 전달되는 경로에서 진단에 사용할 수 있도록, 대상 서버가 받은 요청을 되돌려 주는 루프백 검사를 요청한다. 자원을 저장하는 기능은 아니다.

NoteBase

문 36. 인터넷을 이용한 공격 유형 중 사용자가 자신의 브라우저에서 정확한 웹페이지 주소를 입력해도 가짜 웹페이지에 접속하게 하여 개인 정보를 훔치는 공격 방법으로 옳은 것은?

① 파밍(pharming)
② 스미싱(smishing)
③ 백도어(backdoor)
④ 스파이웨어(spyware)

정답 1번

간단한 해설

주소를 정확히 입력했는데도 위조 사이트로 연결된다는 점이 파밍의 핵심이다. 사용자를 문자메시지로 속이는 스미싱과 구분한다.

핵심 풀이

파밍은 **정상적인 주소를 입력해도 위조된 웹사이트로 연결되도록 유도**하는 공격이다. 주소를 실제 서버 위치로 연결하는 정보나 사용자 환경을 변조하는 등의 방식으로 접속 대상을 바꾼다.

사용자는 정상 사이트에 접속했다고 생각해 개인정보나 인증정보를 입력하지만, 실제로는 공격자의 가짜 사이트에 정보를 넘기게 될 수 있다. 따라서 '정확한 주소를 입력했는데도 가짜 페이지로 접속'한다는 핵심 단서에 해당하는 것은 ① **파밍**이다.

스미싱은 문자메시지를 매개로 속이는 공격이다.

백도어는 정상적인 인증 절차 등을 우회하는 숨은 접근 통로이고,

스파이웨어는 사용자 정보를 몰래 수집하는 소프트웨어이다.

제시문의 접속 대상 변조 방식과 구분한다.

NoteBase

문 37. 암호화 알고리즘인 RSA에 대한 설명으로 옳지 않은 것은?

① 비대칭 암호화 알고리즘이다.

② 소인수 분해의 어려움에 기반을 둔 공개키 암호이다.

③ 리베스트(Rivest), 샤미르(Shamir), 애들먼(Adleman)이 고안하였다.

④ 미국 NIST에서 DES를 대신할 수 있는 새로운 암호 알고리즘을 공모하여 개발되었다.

정답 4번

간단한 해설

DES를 대체하기 위한 NIST의 공개 선정 과정을 거친 것은 **AES**이며, 선정된 알고리즘은 Rijndael이다. RSA에 대한 설명이 아니므로 ④가 옳지 않다.

선지별 해설

① O: RSA는 서로 다른 **공개키와 개인키**를 사용하는 비대칭 암호 알고리즘이다. 암호화에서는 수신자의 공개키로 암호화한 내용을 해당 개인키로 복호화한다.

② O: RSA는 큰 정수를 소인수분해하기 어렵다는 성질과 밀접하게 관련된 공개키 암호이다. 두 큰 소수의 곱을 공개 모듈러스로 사용하며, 그 소인수들을 알면 개인키 계산에 필요한 정보를 구할 수 있다. 다만 모든 RSA 공격이 소인수분해 문제와 수학적으로 동등하다고 증명된 것은 아니다.

③ O: RSA라는 이름은 고안자인 **Rivest, Shamir, Adleman**의 이름 첫 글자에서 유래한다.

④ X: DES를 대체하기 위한 공개 경쟁을 통해 NIST가 선정한 것은 **Rijndael을 바탕으로 한 AES**이다. AES는 같은 비밀키를 사용하는 대칭키 블록 암호이며, 공개키 방식인 RSA와 구별된다.

NoteBase

문 38. 보안 시스템에 대한 설명으로 옳지 않은 것은?

① OTP는 외부 네트워크로부터 내부 네트워크를 보호한다.

② NAC는 접근제어 및 인증을 수행하여 네트워크 접속을 통제한다.

③ VPN은 공중 네트워크를 이용하여 사설 네트워크를 구성하는 솔루션이다.

④ DLP는 사용자 수준에서 정보가 유출되는 것을 막는 솔루션으로 사용자의 다양한 데이터 전송 인터페이스를 제어한다.

정답 1번

간단한 해설

OTP는 일회용 비밀번호를 이용한 인증 수단이다. 외부와 내부 네트워크 사이의 통신을 정책에 따라 걸러 보호하는 방화벽의 역할과 구별해야 한다.

선지별 해설

① (X) OTP(One-Time Password)는 한 번의 인증에 사용하는 비밀번호로, 같은 비밀번호의 재사용 위험을 줄이는 인증 수단이다. 외부 네트워크와 내부 네트워크 사이의 트래픽을 통제하는 기능 자체는 방화벽 등의 역할이다. OTP가 네트워크 로그인 보안에 활용될 수 있다는 점과 네트워크 경계 보호 장비의 기능은 구별한다.

② (O) NAC(Network Access Control)는 사용자나 단말의 인증 결과, 보안 상태와 정책 등을 확인하여 네트워크 접속을 허용·차단·제한한다. 승인되지 않거나 기준을 충족하지 못한 단말의 접속을 통제하는 데 사용된다.

③ (O) VPN(Virtual Private Network)은 인터넷 등 공중 네트워크 위에 논리적인 사설 연결을 구성한다. 보안 VPN은 터널링과 암호화·인증 등을 이용하여 원격 사용자나 떨어진 네트워크 간의 통신을 보호한다.

④ (O) DLP(Data Loss Prevention)는 중요 정보가 허가되지 않은 경로로 나가는 것을 탐지하거나 차단하는 솔루션이다. 엔드포인트 DLP는 USB 저장매체, 인쇄, 파일 전송 등 사용자 단말의 다양한 반출 경로를 정책에 따라 제어할 수 있다.

NoteBase

문 39. 다음과 같은 특징을 갖는 접근제어 모델로 옳은 것은?

· 데이터 무결성에 초점을 둔 모델이다.
· 주체는 자신보다 낮은 무결성 수준의 객체를 읽을 수 없다.
· 주체는 자신보다 높은 무결성 수준의 객체를 수정할 수 없다.

① 비바(Biba)
② 벨-라파듈라(Bell-LaPadula)
③ 임의적 접근 통제(Discretionary Access Control)
④ 역할 기반 접근 통제(Role Based Access Control)

정답 1번

간단한 해설

무결성 보호, 낮은 수준 읽기 금지, 높은 수준 쓰기 금지를 결합한 설명은 비바(Biba) 모델의 엄격한 무결성 정책이다.

개념 구분

비바(Biba) 모델은 낮은 신뢰 수준의 정보나 주체가 높은 신뢰 수준의 데이터를 오염시키지 않도록 **데이터 무결성**을 보호한다. 제시된 두 규칙은 엄격한 무결성 정책의 특징이다.

낮은 무결성 객체 읽기 금지(No Read Down): 높은 무결성의 주체가 **더 낮은 무결성의 정보를 읽어 영향을 받는 것**을 막는다. 따라서 주체는 자신보다 낮은 무결성 수준의 객체를 읽을 수 없다는 단서와 일치한다.

높은 무결성 객체 쓰기 금지(No Write Up): 낮은 무결성의 주체가 **더 높은 무결성의 데이터를 수정하는 것**을 막는다. 따라서 주체는 자신보다 높은 무결성 수준의 객체를 수정할 수 없다는 단서와 일치한다.

벨-라파듈라는 기밀성 보호를 중심으로 하며 기본 규칙은 상위 보안등급 읽기 금지와 하위 보안등급 쓰기 금지이다.

임의적 접근 통제는 소유자 등의 재량에 따른 권한 부여,

역할 기반 접근 통제는 역할에 따른 권한 부여가 중심이므로 제시된 무결성 수준 규칙을 특징으로 하는 비바와 다르다.

따라서 해당하는 접근제어 모델은 ① 비바이다.

NoteBase

문 40. 가짜 ID와 패스워드를 네트워크에 뿌려 이를 이용하여 접속을 시도하는 공격자를 탐지하는 방법으로 옳은 것은?

① ARP를 이용한 스니퍼 탐지

② 유인을 이용한 스니퍼 탐지

③ ping을 이용한 스니퍼 탐지

④ DNS를 이용한 스니퍼 탐지

정답 2번

간단한 해설

가짜 인증정보를 미끼로 두고 **그 정보를 사용한 접속 시도를 관찰**하므로 유인을 이용한 탐지이다. 프로토콜 응답 검사와 달리 공격자의 미끼 사용 행동이 탐지 단서가 된다.

핵심 풀이

유인을 이용한 탐지는 공격자가 관심을 가질 만한 **미끼 정보**를 두고, 그 정보가 이용되는지를 관찰하는 방식이다. 제시문에서는 정상 사용자가 쓸 이유가 없는 가짜 ID와 패스워드가 미끼이다.

누군가 네트워크에서 이 정보를 수집한 뒤 실제 접속에 사용하면, 그 시도가 정보의 탈취·악용을 의심할 단서가 된다. 따라서 해당 방법은 ② **유인을 이용한 스니퍼 탐지**이다.

ARP·ping·DNS를 이용하는 탐지는 각 프로토콜과 관련된 응답이나 조회 같은 네트워크 동작을 관찰하는 접근이다. 이 문제는 특정 프로토콜의 반응이 아니라 가짜 인증정보를 사용하도록 유도한다는 점으로 판별한다. 다만 미끼를 사용하지 않는 수동 도청까지 반드시 탐지한다는 뜻은 아니다.

NoteBase