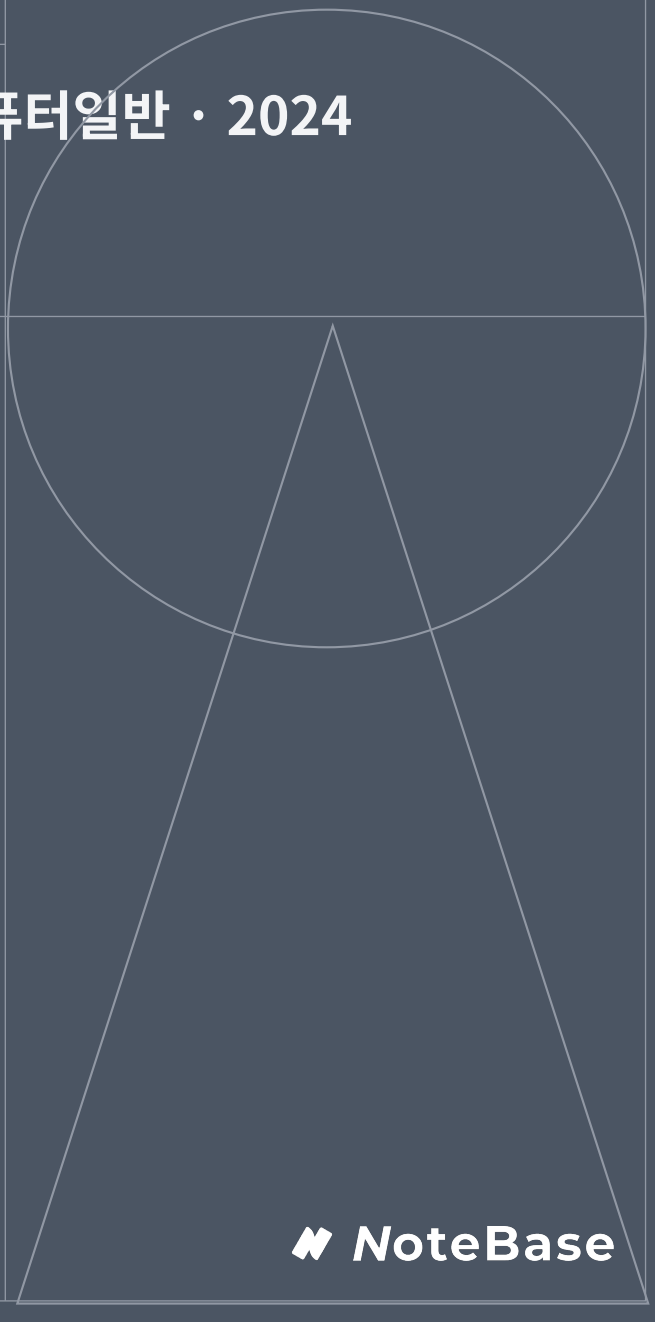
기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2024



NoteBase

기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2024

공무원/소방/경력채용 · 컴퓨터일반 · 2024

문 1. 다음 2진수, 8진수, 16진수의 연산 결과를 10진수로 나타낸 값으로 옳은 것은?

$1101_{(2)} + 21_{(8)} + 1D_{(16)}$

① 57

② 59

③ 61

④ 63

정답 2번

간단한 해설

각 수를 **10진수로 변환한 뒤 더한다**. 2진수 1101은 13, 8진수 21은 17, 16진수 1D는 29이므로 합은 **59**이다. 따라서 정답은 ②이다.

계산 과정

1. 2진수를 10진수로 변환한다. 오른쪽 자리부터 가중치는 1, 2, 4, 8이다.

$$1101_{(2)} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$$

2. 8진수는 각 자릿값에 8의 거듭제곱을 곱한다.

$$21_{(8)} = 2 \times 8^1 + 1 \times 8^0 = 16 + 1 = 17$$

3. 16진수에서 A, B, C, D는 각각 10, 11, 12, 13을 뜻하므로 **D=13**을 대입한다.

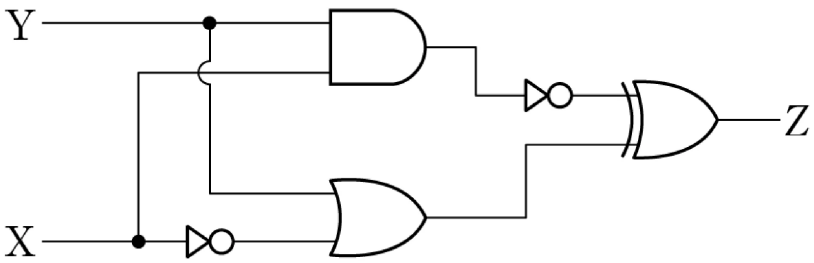
$$1D_{(16)} = 1 \times 16^1 + 13 \times 16^0 = 16 + 13 = 29$$

4. 변환한 10진수를 모두 더하면 $13 + 17 + 29 = 59$이다. 이는 ②의 59와 일치한다.

NoteBase

문 2. 다음 논리회로에 대한 진리표의 ( ) 안에 들어갈 값으로 옳은 것은?

논리회로



진리표

X	0	0	1	1
Y	0	1	0	1
Z	(ㄱ)	(ㄴ)	(ㄷ)	(ㄹ)

① ㄱ: 0, ㄴ: 0, ㄷ: 1, ㄹ: 1

② ㄱ: 0, ㄴ: 0, ㄷ: 0, ㄹ: 0

③ ㄱ: 0, ㄴ: 1, ㄷ: 0, ㄹ: 1

④ ㄱ: 1, ㄴ: 1, ㄷ: 0, ㄹ: 0

정답 1번

간단한 해설

위쪽 NAND 결과와 아래쪽 OR 결과를 **마지막 XOR 게이트에서 비교**한다. 입력 순서 (0,0), (0,1), (1,0), (1,1)에 대한 출력은 각각 0, 0, 1, 1이다.

NoteBase

계산 과정

위쪽 가지는 X와 Y를 AND 연산한 뒤 NOT을 거치므로 $A = \overline{XY}$이다. 아래쪽 가지는 X를 반전한 값과 Y의 OR이므로 $B = \overline{X} + Y$이다. 마지막 게이트는 입력 쪽 곡선이 하나 더 있는 **XOR 게이트**이며, 두 입력이 다르면 1, 같으면 0을 출력한다. 따라서 $Z = A \oplus B$로 계산한다.

ㄱ: $X = 0,\ Y = 0$이면 $A = \overline{0 \cdot 0} = 1, B = \overline{0} + 0 = 1$이다. 두 입력이 같으므로 $Z = 1 \oplus 1 = 0$이다.

ㄴ: $X = 0,\ Y = 1$이면 $A = \overline{0 \cdot 1} = 1, B = \overline{0} + 1 = 1$이다. 따라서 $Z = 1 \oplus 1 = 0$이다.

ㄷ: $X = 1,\ Y = 0$이면 $A = \overline{1 \cdot 0} = 1, B = \overline{1} + 0 = 0$이다. 두 입력이 다르므로 $Z = 1 \oplus 0 = 1$이다.

ㄹ: $X = 1,\ Y = 1$이면 $A = \overline{1 \cdot 1} = 0, B = \overline{1} + 1 = 1$이다. 따라서 $Z = 0 \oplus 1 = 1$이다.

따라서 ㄱ=0, ㄴ=0, ㄷ=1, ㄹ=1로 ①과 일치한다.

NoteBase

문 3. 다음 조건을 고려하여 10진수 -5를 2의 보수 표기법으로 표현한 2진수로 옳은 것은?

· 최상위(맨 왼쪽) 비트 자리에는 부호 비트를 저장한다.
· 부호 비트가 0이면 양수, 1이면 음수를 표현한다.

① $00011010_{(2)}$
② $11111010_{(2)}$
③ $00000101_{(2)}$
④ $11111011_{(2)}$

정답 4번

간단한 해설

음수의 2의 보수는 같은 비트 수로 나타낸 양수의 **모든 비트를 반전한 뒤 1을 더해** 구한다. 5의 8비트 표현인 $00000101_{(2)}$에 이 과정을 적용하면 $11111011_{(2)}$이다.

계산 과정

① 양수의 2진수 표현: 선택지와 같은 8비트로 5를 나타내면 $00000101_{(2)}$이다.

② 비트 반전: 모든 비트의 0과 1을 바꾸면 $11111010_{(2)}$이다. 이는 1의 보수를 구한 중간 결과이므로 여기서 멈추면 안 된다.

③ 1 더하기: 반전한 값에 1을 더하면 $11111010_{(2)} + 1 = 11111011_{(2)}$이다. 최상위 비트가 1이므로 음수를 나타내며, -5의 8비트 2의 보수 표현은 ④이다.

NoteBase

문 4. 다음 조건에서 기억장치의 유효 접근 시간[ns]은?

· 캐시 적중률: 90%
· 캐시 메모리 접근 시간: 30 ns
· 주기억장치 접근 시간: 100 ns

① 20
② 25
③ 35
④ 40

정답 4번

간단한 해설

캐시 미스가 나면 **캐시 확인 시간에 주기억장치 접근 시간이 추가**된다. 적중 시 30 ns, 실패 시 130 ns를 각각의 확률로 가중평균하면 40 ns이다.

계산 과정

캐시를 먼저 확인하고, 실패한 경우 주기억장치에 추가로 접근하는 방식으로 계산한다. **캐시 미스율은 10%**이다.

적중할 때는 캐시 접근 시간 30 ns만 걸린다. 실패할 때는 캐시 확인 30 ns에 주기억장치 접근 100 ns가 더해져 **130 ns**가 걸린다.

유효 접근 시간은 두 경우의 확률 가중평균이다.
$T = 0.9 \times 30 + 0.1 \times (30 + 100) = 40\text{ns}$

따라서 유효 접근 시간은 **40 ns**로, 정답은 ④이다. 미스가 발생해도 캐시를 먼저 확인한 시간이 이미 들었으므로 이를 빼면 안 된다.

NoteBase

문 5. 다음 조건을 고려하여 3,000 × 2,000픽셀로 저장한 이미지 데이터의 크기[MB]는?

· 1 KB = 1,000 byte이고 1 MB = 1,000 KB이다.
· 각 픽셀은 RGB 컬러로 구성되며 R, G, B는 각각 8비트로 표현된다.

① 4.8
② 6.2
③ 18
④ 36

정답 3번

간단한 해설

RGB 각 성분이 8비트이므로 **한 픽셀은 24비트, 즉 3바이트**이다. 600만 픽셀에 3바이트를 곱하고 문제의 십진 단위로 환산하면 **18 MB**가 된다.

계산 과정

전체 픽셀 수는 3,000 × 2,000 = 6,000,000개이다.

한 픽셀은 R, G, B 각각 8비트로 이루어지므로 8 × 3 = 24비트이다. 8비트가 1바이트이므로 픽셀당 크기는 24 ÷ 8 = 3바이트이다.

전체 이미지 데이터는 6,000,000 × 3 = 18,000,000바이트이다. 문제에서
1MB = 1,000 × 1,000 = 1,000,000바이트로 정했으므로 18,000,000 ÷ 1,000,000 = 18MB이며, ③에 해당한다.

NoteBase

문 6. gcc 컴파일러가 소스 코드를 이용하여 실행 파일을 생성하는 과정을 순서대로 옳게 나열한 것은?

① Assemble → Pre-process → Link → Compile

② Pre-process → Assemble → Link → Compile

③ Pre-process → Compile → Assemble → Link

④ Assemble → Compile → Pre-process → Link

정답 3번

간단한 해설

각 단계의 산출물이 다음 단계의 입력이 된다. 소스의 전처리 결과를 번역하고, 어셈블리 코드를 목적 파일로 바꾼 뒤 **목적 파일들을 링크하여 실행 파일**을 만든다.

올바른 순서

Pre-process → Compile → Assemble → Link

Pre-process(전처리): #include로 지정한 헤더를 포함하고 매크로 확장·조건부 컴파일 지시문 등을 처리한다. **전처리 결과가 실제 컴파일의 입력**이 되므로 가장 먼저 수행한다.

Compile(컴파일): 전처리된 소스의 문법과 의미를 분석하고, 필요한 최적화 등을 거쳐 **어셈블리 코드**를 생성한다. 어셈블러가 처리할 코드가 이 단계에서 만들어진다.

Assemble(어셈블): 앞 단계의 어셈블리 코드를 기계어와 심벌·재배치 정보 등을 담은 **목적 파일**로 변환한다. 여러 파일에 나뉜 함수나 변수의 연결을 위해 이 결과를 링커에 넘긴다.

Link(링크): 목적 파일들과 필요한 라이브러리를 결합하고, 외부 심벌 참조와 주소 배치를 처리하여 **실행 파일**을 만든다. 연결할 목적 파일이 먼저 필요하므로 어셈블 뒤에 수행한다.

NoteBase

문 7. 프로그램 실행에 관한 설명으로 옳은 것은?

① 로딩(loading)은 프로그램을 실행하려고 메모리로 가져오는 작업이다.

② 라이브러리(library)는 독립적으로 서브루틴(subroutine) 복사본을 실행하는 도구이다.

③ 링킹(linking)은 지정된 기존 주소와 다른 주소에 로딩할 수 있도록 목적 프로그램을 수정하는 작업이다.

④ 재배치(relocating)는 분리된 목적 프로그램을 결합하고, 상호 참조할 수 있도록 정보를 제공하는 작업이다.

정답 1번

간단한 해설

로딩은 실행할 프로그램을 메모리에 적재하는 작업이므로 ①이 옳다. 라이브러리는 재사용 가능한 코드의 모음이며, 여러 목적 모듈을 연결하는 **링킹**과 적재 위치에 맞게 주소를 조정하는 **재배치**를 구별해야 한다.

선지별 해설

① O: 로딩은 저장장치에 있는 실행할 프로그램과 필요한 데이터를 메모리로 가져오는 작업이다. 이 작업을 담당하는 것이 로더(loader)이다.

② X: 라이브러리는 프로그램에서 호출하여 사용할 수 있도록 함수나 서브루틴 등을 모아 놓은 것이다. 서브루틴 복사본을 독립적으로 실행하는 도구가 아니라 **재사용 가능한 코드의 모음**이다.

③ X: 제시된 설명은 **재배치**이다. 재배치는 프로그램이 실제로 적재되는 위치에 맞게 주소 참조를 조정하는 작업이다. 링킹은 분리된 목적 모듈과 필요한 라이브러리를 결합하고 외부 심벌 참조를 해결하는 작업이다.

④ X: 제시된 설명은 **링킹**이다. 링킹은 모듈을 연결하고 다른 모듈의 함수·변수에 대한 참조를 해결한다. 재배치는 모듈 결합 자체가 아니라 적재 위치에 따른 주소 조정에 초점을 둔다.

NoteBase

문 8. 다음 C 프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
int func(int x) {
  int y = 1;

  printf("%d", y++);
  x = x + y;

  return x;
}
void main() {
  int x = 1;
  int y;

  y = func(x);

  printf(" %d %d\n", x, y);
}
```

① 1 1 3

② 1 3 3

③ 2 1 3

④ 2 3 3

정답 1번

간단한 해설

y++는 증가 전 값 1을 출력한 뒤 y를 2로 만든다. func는 3을 반환하지만 **main의 x는 값 전달 때문에 1로 유지**되어 전체 출력은 1 1 3이다.

NoteBase

계산 과정

main에서 x는 1이며, func(x)를 호출하면 그 값 1이 func의 매개변수 x에 복사된다. func의 지역변수 y는 1로 초기화된다.

func 안의 y++는 **후위 증가**이므로 증가 전 값 1이 먼저 출력되고, y는 2가 된다. 다음 문장에서 func의 x는 $1 + 2 = 3$이 되고, 이 값 3을 반환한다.

main의 y에는 반환값 3이 저장된다. C의 값 전달에서는 func의 매개변수 x를 바꿔도 main의 x는 바뀌지 않으므로 main의 x는 여전히 1이다. 마지막 출력은 1과 3이며, 앞서 출력한 1까지 합친 전체 결과는 **1 1 3**으로 ①이다.

NoteBase

문 9. 다음 C 프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
int func(int *x, int *y) {
  *x += 1;
  *y += 1;

  return (*x - *y);
}
void main() {
  int x = 10, y = 20;
  int z;

  z = func(&x, &y);

  printf("%d\n", x + y + z);
}
```

① 20

② 22

③ 24

④ 26

정답 2번

간단한 해설

변수의 주소를 넘겨 역참조로 수정하므로 **main의 x와 y가 각각 11과 21로 바뀐다**. 반환값 z는 −10이어서 x + y + z는 22이다.

NoteBase

계산 과정

처음에는 main의 x가 10, y가 20이다. func(&x, &y)는 **두 변수의 주소를 전달**하므로, 함수 안에서 포인터를 역참조하여 값을 바꾸면 main의 변수도 바뀐다.

*x += 1에 의해 main의 x는 11이 되고, *y += 1에 의해 main의 y는 21이 된다. 포인터 자체의 주소를 증가시키는 연산이 아니다.

func가 반환하는 값은 *x − *y이므로 **11 − 21 = −10**이다. 이 값이 z에 저장된다.

출력식 x + y + z는 11 + 21 − 10 = **22**이다. 따라서 정답은 ②이다.

NoteBase

문 10. 다음 파이썬(Python) 코드에 관한 설명으로 옳지 않은 것은?

라인	소스 코드
1	`data1 = [1,2,3,4,5]`
2	`data1.pop(0);`
3	`print(data1)`
4	
5	`data2 = ('h', 'e', 'l', 'l', 'o')`
6	`print(data2[1:3])`
7	
8	`data3 = {'name':'nam', 'age':25}`
9	`data3['weight'] = 50;`
10	`print(data3)`

① 라인 3의 출력은 [2, 3, 4, 5]이다.

② 라인 6의 출력은 ('e', 'l', 'l')이다.

③ data1은 리스트, data2는 튜플, data3은 딕셔너리이다.

④ 라인 10의 출력은 {'name': 'nam', 'age': 25, 'weight': 50}이다.

정답 2번

간단한 해설

튜플 슬라이싱에서 **끝 인덱스는 포함하지 않는다**. data2[1:3]에는 인덱스 1과 2의 원소만 들어가므로 출력은 ('e', 'l')이다.

NoteBase

선지별 해설

① O: data1.pop(0)은 **인덱스 0에 있는 첫 원소 1을 제거**한다. 따라서 리스트에 2, 3, 4, 5가 남아 라인 3은 [2, 3, 4, 5]를 출력한다.

② X: 슬라이스 data2[1:3]은 **시작 인덱스 1은 포함하고 끝 인덱스 3은 제외**한다. 인덱스 1의 'e'와 2의 'l'만 선택하므로 실제 출력은 ('e', 'l')이다. 인덱스 3의 두 번째 'l'까지 포함한 선지의 출력은 틀렸다.

③ O: data1은 대괄호로 원소를 나열한 **리스트**, data2는 쉼표로 구분한 원소를 소괄호로 묶은 **튜플**이다. data3은 중괄호 안에 'name': 'nam', 'age': 25처럼 키와 값의 쌍을 저장한 딕셔너리이다.

④ O: data3['weight'] = 50은 기존 딕셔너리에 없는 **'weight' 키와 값 50을 추가**한다. 기존 'name'과 'age' 항목은 유지되므로 라인 10은 {'name': 'nam', 'age': 25, 'weight': 50}을 출력한다.

따라서 옳지 않은 설명은 ②이다.

NoteBase

문 11. 자료의 크기가 작은 것에서 큰 순서로 옳게 나열된 것은?

① 파일 < 필드 < 데이터베이스

② 파일 < 데이터베이스 < 필드

③ 필드 < 파일 < 데이터베이스

④ 필드 < 데이터베이스 < 파일

정답 3번

간단한 해설

자료 구성 계층에서는 **필드가 모여 레코드**를 이루고, 관련 레코드가 모여 파일을 이룬다. 데이터베이스는 관련 자료를 통합한 집합이므로, 제시된 세 단위 중 필드가 가장 작고 데이터베이스가 가장 크다.

올바른 순서

필드 → 파일 → 데이터베이스

필드: 이름·주소처럼 하나의 속성을 나타내는 자료 단위이다. 여러 필드가 모여 하나의 레코드를 이룬다.

파일: 관련된 레코드들을 모은 단위이므로, 레코드를 구성하는 필드보다 큰 조직 단위이다.

데이터베이스: 관련된 자료를 통합하여 관리하는 집합으로, 이 문항의 자료 구성 계층에서는 여러 관련 파일을 포괄하는 가장 큰 단위이다.

NoteBase

문 12. 다음 릴레이션 R에 대한 관계대수 연산 결과로 옳은 것은?

릴레이션 R	A | B | C 2 | 5 | 7 3 | 3 | 3 4 | 3 | 2 5 | 2 | 8
관계대수 연산	$\pi_A(\sigma_{B>C}(R))$

① 2

② 3

③ 4

④ 5

정답 3번

간단한 해설

안쪽의 선택 연산으로 **B>C인 행**을 먼저 고르고, 바깥의 투영 연산으로 **A열만 추출**한다. 조건을 만족하는 행은 (4, 3, 2)뿐이므로 결과는 A값 4이며 정답은 ③이다.

NoteBase

계산 과정

1. 관계대수식 $\pi_A(\sigma_{B>C}(R))$에서 σ는 조건에 맞는 행을 고르는 선택, π는 지정한 열을 추출하는 투영이다. 괄호 안의 선택 연산부터 수행한다.

2. 각 행에서 B와 C를 비교한다.

(2, 5, 7): 5>7은 거짓이므로 제외한다.

(3, 3, 3): 3>3은 거짓이므로 제외한다. 조건이 '이상'이 아니라 **'초과'**라는 점에 유의한다.

(4, 3, 2): 3>2가 참이므로 선택한다.

(5, 2, 8): 2>8은 거짓이므로 제외한다.

3. 남은 행 (4, 3, 2)에서 A열을 투영하면 **A값이 4인 한 행**이 남는다. 따라서 결과값 4에 해당하는 ③이 옳다.

NoteBase

문 13. 데이터베이스 뷰(view)에 관한 설명으로 옳은 것은?

① 생성한 뷰에 대해 SELECT 질의어를 사용할 수 없다.

② GROUP BY 절을 포함하여 정의한 뷰는 변경할 수 없다.

③ CREATE VIEW로 정의하며, ALTER VIEW로 갱신한다.

④ DISTINCT 키워드를 포함하여 정의한 뷰는 변경할 수 있다.

정답 2번

간단한 해설

GROUP BY로 여러 원본 행을 묶은 뷰는 결과 행을 어느 원본 행의 변경으로 돌려야 할지 일반적으로 정할 수 없어 **통상적인 뷰 데이터 갱신이 제한**된다. 뷰의 정의·속성을 바꾸는 일과 뷰를 통해 데이터를 갱신하는 일은 다르다.

선지별 해설

① (X) 뷰는 질의 결과를 테이블처럼 사용할 수 있게 한 가상 테이블이므로 **SELECT로 조회할 수 있다**. 생성한 뷰를 조회할 수 없다는 설명은 틀리다.

② (O) GROUP BY가 포함된 뷰는 여러 원본 행을 그룹으로 묶어 나타내므로 일반적인 갱신 가능한 뷰의 조건을 충족하지 않는다. 여기서 변경할 수 없다는 것은 **뷰를 통한 통상적인 데이터 갱신이 제한된다**는 뜻이지, 뷰의 정의 자체를 다시 작성할 수 없다는 뜻이 아니다.

③ (X) CREATE VIEW로 뷰를 정의한다는 부분은 맞지만, ALTER VIEW를 데이터 갱신 명령으로 설명한 부분이 틀리다. 갱신 가능한 뷰의 데이터를 수정할 때는 UPDATE 등을 사용한다. DBMS가 지원하는 ALTER VIEW는 뷰의 정의나 속성을 바꾸는 명령으로 구별해야 한다.

④ (X) DISTINCT로 중복을 제거한 결과 행은 여러 원본 행에 대응할 수 있으므로 일반적인 갱신 가능한 뷰의 조건을 충족하지 않는다. 따라서 DISTINCT가 포함된 뷰를 통상적으로 데이터 갱신할 수 있다는 설명은 틀리다.

NoteBase

문 14. 소프트웨어 개발방법론 중 폭포수 모델에 관한 설명으로 옳은 것은?

① 개발 단계별로 반복하여 소프트웨어의 품질을 개선한다.

② 시제품(prototype)을 활용하여 신속하게 결과물을 개발한다.

③ 소프트웨어 개발 절차보다 실제 동작하는 결과물을 강조한다.

④ 소프트웨어 개발 과정에서 단계별 결과 산출물 관리를 강조한다.

정답 4번

간단한 해설

폭포수 모델은 개발 단계를 순차적으로 진행하며 **단계별 산출물을 작성·검토하고 관리**하는 방식이다. 반복 개선, 시제품 활용, 동작하는 소프트웨어 중심의 가치와 구분한다.

선지별 해설

① (X) 짧은 주기나 여러 차례의 반복으로 결과를 보완하는 것은 반복적 개발의 특징이다. 폭포수 모델의 기본 구조는 **요구사항 분석, 설계, 구현, 시험 등의 단계를 순차적으로 진행**하는 것이다.

② (X) 시제품을 만들어 사용자의 요구를 확인하고 보완하는 방식은 프로토타이핑 모델의 특징이다. 폭포수 모델은 시제품의 반복 개선보다 단계별 작업과 산출물의 완성을 중심으로 진행한다.

③ (X) 절차와 문서보다 실제 동작하는 소프트웨어의 가치를 강조하는 것은 애자일 접근의 특징이다. 폭포수 모델은 미리 정한 개발 단계와 각 단계의 산출물 관리를 중시한다.

④ (O) 폭포수 모델은 **각 단계의 결과를 문서 등의 산출물로 남기고 검토**한 뒤 다음 단계로 진행한다. 요구사항 명세서, 설계서 등 단계별 산출물이 다음 단계의 입력과 관리 기준이 되므로 옳은 설명이다.

NoteBase

문 15. 구조 다이어그램(structural diagram)에 해당하는 것으로 옳은 것은?

① 클래스 다이어그램

② 시퀀스 다이어그램

③ 액티비티 다이어그램

④ 유스케이스 다이어그램

정답 1번

간단한 해설

클래스 다이어그램은 **클래스와 그 사이의 정적인 관계**를 표현하는 구조 다이어그램이다.
시퀀스·액티비티·유스케이스 다이어그램은 행위 다이어그램에 속한다.

선지별 해설

UML의 구조 다이어그램은 시스템을 이루는 요소와 그 관계를 정적인 관점에서 나타내고, 행위 다이어그램은 시스템의 기능이나 동작을 나타낸다.

① O: 클래스 다이어그램은 클래스의 속성·연산과 클래스 사이의 연관·일반화 등의 관계를 나타낸다. 시스템의 **정적인 구성과 관계**를 표현하므로 구조 다이어그램이다.

② X: 시퀀스 다이어그램은 참여 객체나 생명선 사이에서 오가는 **메시지의 시간적 순서**를 나타낸다. 행위 다이어그램에 속하는 상호작용 다이어그램이지, 구조 다이어그램이 아니다.

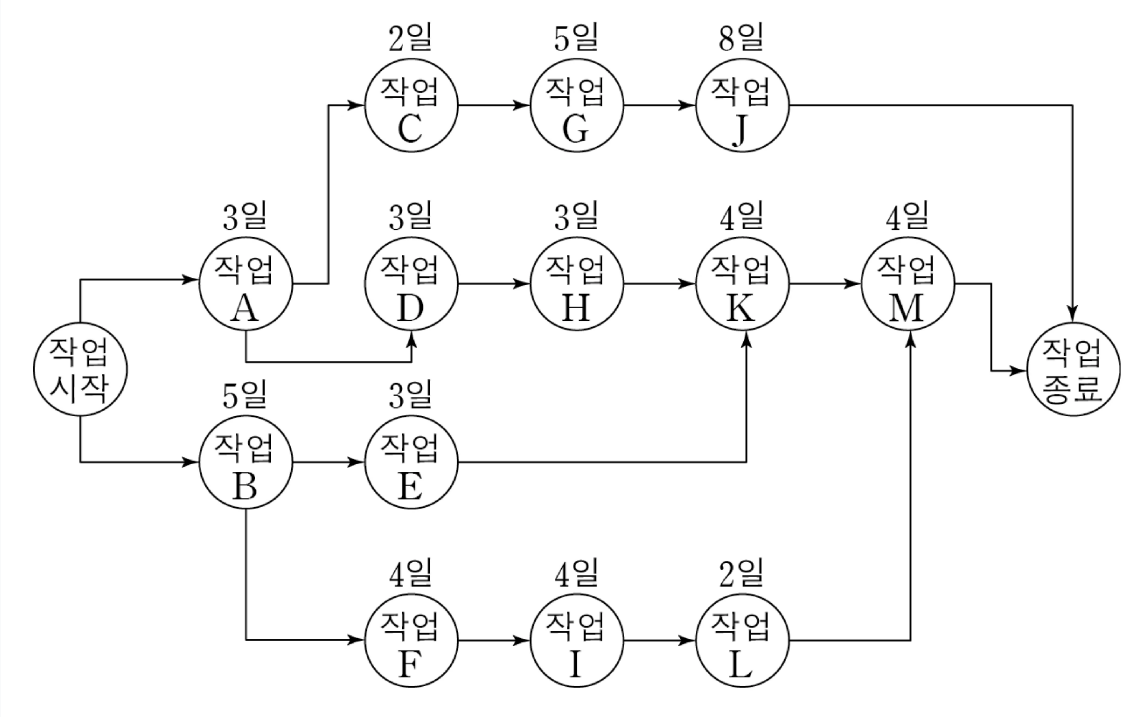
③ X: 액티비티 다이어그램은 작업의 흐름, 분기, 병행 처리 등을 나타낸다. 구성 요소의 정적인 관계보다 **행동이 진행되는 흐름**을 표현하므로 행위 다이어그램이다.

④ X: 유스케이스 다이어그램은 외부 액터와 시스템이 제공하는 기능, 그리고 그 관계를 나타낸다. 사용자의 관점에서 **시스템이 수행해야 할 기능**을 표현하는 행위 다이어그램이다.

따라서 구조 다이어그램에 해당하는 것은 ① 클래스 다이어그램이다.

NoteBase

문 16. 다음 CPM(Critical Path Method) 네트워크에서 임계 경로의 소요 기간으로 옳은 것은?



① 16일

② 17일

③ 18일

④ 19일

정답 4번

간단한 해설

임계 경로는 시작부터 종료까지 연결되는 경로 중 **소요 기간의 합이 가장 긴 경로**이다. B → F → I → L → M의 기간은 5 + 4 + 4 + 2 + 4 = 19일로, 전체 작업의 완료 시점을 결정한다.

NoteBase

계산 과정

① 경로별 기간 합산: 시작부터 종료까지 연결되는 경로의 작업 기간을 더한다.

A → C → G → J: 3 + 2 + 5 + 8 = 18일

A → D → H → K → M: 3 + 3 + 3 + 4 + 4 = 17일

B → E → K → M: 5 + 3 + 4 + 4 = 16일

B → F → I → L → M: 5 + 4 + 4 + 2 + 4 = 19일

② 합류점 확인: K는 H가 끝나는 9일과 E가 끝나는 8일 중 늦은 9일에 시작하여 13일에 끝난다. M은 K가 끝나는 13일과 L이 끝나는 15일 중 늦은 15일에 시작하여 19일에 끝난다.

③ 임계 경로 결정: 가장 긴 경로는 **B → F → I → L → M**이며, 소요 기간은 19일이다. J가 끝나는 18일보다 M의 종료가 늦으므로 전체 작업은 19일에 완료되며, 정답은 ④이다.

NoteBase

문 17. 다음 소프트웨어 모듈 간 응집 관계에서 ( ) 안에 들어갈 용어로 옳은 것은?

낮은 응집도 → 높은 응집도
우연적 응집, (ㄱ), (ㄴ), (ㄷ), 교환적 응집

① ㄱ: 논리적 응집, ㄴ: 시간적 응집, ㄷ: 절차적 응집
② ㄱ: 논리적 응집, ㄴ: 절차적 응집, ㄷ: 시간적 응집
③ ㄱ: 절차적 응집, ㄴ: 논리적 응집, ㄷ: 시간적 응집
④ ㄱ: 절차적 응집, ㄴ: 시간적 응집, ㄷ: 논리적 응집

정답 1번

간단한 해설

응집도는 **우연적 → 논리적 → 시간적 → 절차적 → 교환적** 순으로 높아진다. 성격의 유사성, 수행 시점, 실행 순서, 자료 공유의 관계를 구분하면 빈칸의 순서를 정할 수 있다.

올바른 순서

우연적 응집 → **논리적 응집** → **시간적 응집** → **절차적 응집** → 교환적 응집

우연적 응집은 뚜렷한 관련이 없는 요소들을 한 모듈에 모은 경우이다.

논리적 응집은 입출력 처리처럼 성격이 유사한 기능들을 분류하여 모은 경우이므로 우연적으로 모은 것보다 관련성이 높다.

시간적 응집은 초기화처럼 같은 시점에 수행되는 작업들을 모은 경우이다.

절차적 응집은 정해진 실행 순서를 따라 수행해야 하는 작업들을 모은 경우로, 단지 실행 시점만 같은 경우보다 작업 사이의 절차적 연결이 강하다.

교환적 응집은 같은 입력 자료를 이용하거나 같은 출력 자료를 생성하는 작업들을 묶은 경우이다. 제시된 순서에서는 절차적 응집보다 응집도가 높다.

NoteBase

문 18. 다음 설명에 해당하는 용어로 옳은 것은?

개발된 소프트웨어의 사용자 요구사항 준수 여부를 확인하려고 생성한 입력값, 실행 조건, 예상된 결과의 집합을 의미한다.

① 코드 정의서
② 테스트 케이스
③ 요구사항 명세서
④ 제품 소프트웨어 매뉴얼

정답 2번

간단한 해설

입력값·실행 조건·예상 결과를 묶은 시험 단위는 테스트 케이스이다. 요구사항이 검증할 기준이라면, 테스트 케이스는 그 기준을 실제로 어떻게 확인할지 구체화한다.

개념 구분

테스트 케이스는 특정 요구사항이나 동작을 확인하기 위해 입력값, 실행 조건, 예상 결과 등을 구체적으로 정리한 시험 단위이다. 실제 실행 결과를 예상 결과와 비교하여 요구사항 충족 여부를 판단하므로, 지문의 설명은 ②에 해당한다.

요구사항 명세서는 소프트웨어가 무엇을 해야 하는지와 충족할 제약을 기술한 문서이다. 이것을 바탕으로 테스트 케이스를 만들 수 있지만, 명세서 자체가 개별 시험의 입력값·실행 조건·예상 결과를 묶은 것은 아니다.

코드 정의서는 시스템에서 쓰는 코드의 값·의미·구성 규칙 등을 정의한다.

제품 소프트웨어 매뉴얼은 설치·사용 방법 등을 안내한다. 둘 다 지문이 설명한 구체적인 시험 단위와 구별된다.

NoteBase

문 19. 형상 관리에 관한 설명으로 옳지 않은 것은?

① CVS, SVN, Git은 형상 관리 도구이다.

② 프로젝트 요구 분석서, 소스 코드는 형상 관리 대상이다.

③ 소프트웨어 생명주기에 발생하는 변경 사항을 체계적으로 관리한다.

④ 체크인(check-in)으로 형상 관리 대상의 최신 버전을 로컬 공간에 저장한다.

정답 4번

간단한 해설

체크인은 **작업한 변경 내용을 저장소에 반영**하는 작업이다. 저장소의 내용을 로컬 작업 공간으로 가져오는 체크아웃과 구별해야 한다.

선지별 해설

① O: CVS, SVN, Git은 파일의 버전과 변경 이력을 관리하는 도구로, **소프트웨어 형상 관리의 버전 관리 기능**을 지원한다. CVS와 SVN은 대표적인 중앙집중식 도구이고, Git은 분산 버전 관리 도구이다.

② O: 형상 관리의 대상은 소스 코드에 한정되지 않는다. 요구 분석서, 설계서, 시험 자료 등도 형상 항목으로 정하여 **산출물의 버전과 변경 관계**를 관리할 수 있다.

③ O: 형상 관리는 개발부터 유지보수까지 생명주기 전반의 산출물을 식별하고 변경을 통제·기록한다. 이를 통해 **어떤 버전에 어떤 변경이 반영되었는지** 추적하고 산출물의 일관성을 유지한다.

④ X: 체크인은 작업 공간에서 수정한 내용을 **저장소에 새 버전으로 등록**하는 작업이다. 저장소의 버전을 로컬 작업 공간으로 가져오는 것은 체크아웃(check-out)에 해당하며, 이미 만든 작업 사본을 최신화하는 작업은 보통 업데이트(update)로 구분한다. 선지는 체크인의 방향을 반대로 설명했다.

따라서 옳지 않은 설명은 ④이다.

NoteBase

문 20. 다음에서 설명하는 리먼(Lehman)이 발견한 소프트웨어 진화의 법칙으로 옳은 것은?

시스템의 요구는 항상 변화하기 때문에 소프트웨어가 사용되려면 계속 좋은 방향으로 진화(evolution)되어야 한다. 따라서 시스템의 유지 보수성을 높이는 일이 중요하다.

① 자기 통제의 법칙
② 품질 저하의 법칙
③ 복잡도 증가의 법칙
④ 지속적인 변경의 법칙

정답 4번

간단한 해설

요구와 환경이 변하므로 소프트웨어도 계속 사용되려면 그 변화에 맞추어 수정되어야 한다는 것은 **지속적인 변경의 법칙**이다. 따라서 정답은 ④이다.

개념 구분

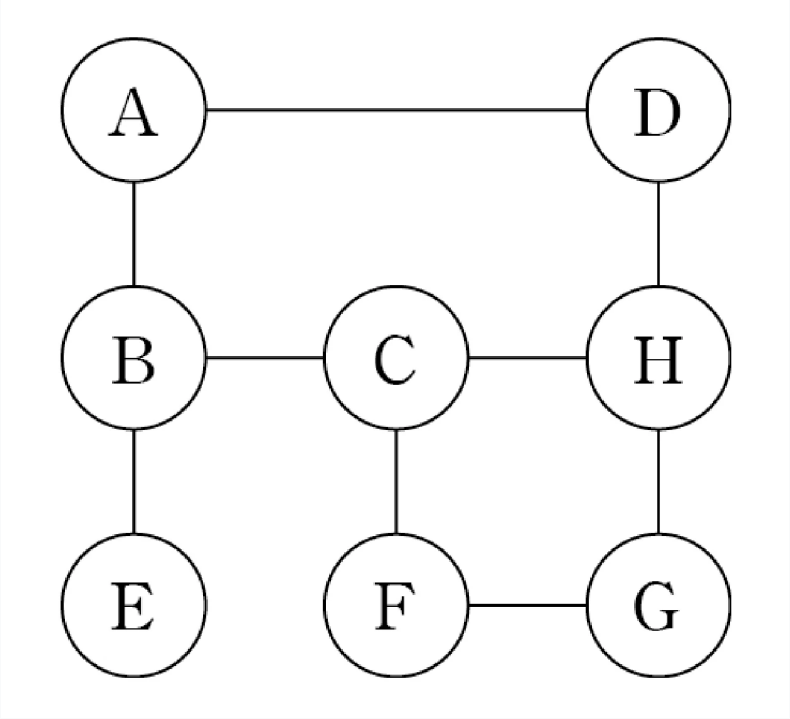
지속적인 변경의 법칙: 실제 환경에서 사용되는 소프트웨어는 변화하는 요구에 계속 적응해야 한다. 그렇지 않으면 사용자의 기대를 점점 충족하지 못하게 된다. 지문의 핵심은 '요구가 변화한다 → 계속 사용하려면 소프트웨어도 변화해야 한다'는 연결이다.

품질 저하의 법칙과의 구별: 품질 저하의 법칙은 변화하는 운영 환경에 맞추어 엄격히 유지·적응시키지 않으면 소프트웨어의 품질이 낮아진 것으로 인식된다는 데 초점을 둔다. 두 법칙은 관련되지만, 지문은 품질 저하라는 결과보다 지속적으로 변경해야 한다는 필요성을 직접 설명한다.

지속적인 변경을 원활하게 수행하려면 코드를 이해하고 수정하기 쉬워야 하므로, 지문에서 강조한 **유지 보수성**을 높이는 일이 중요하다.

NoteBase

문 21. 다음 그래프에서 A 정점을 시작으로 깊이우선탐색 수행 시 방문 순서에 따른 정점을 옳게 나열한 것은? (단, 알파벳순에 따라 방문하지 않은 이웃 정점을 선택한다.)



① A → B → C → F → G → H → D → E

② A → B → C → H → F → G → E → D

③ A → B → E → C → F → G → H → D

④ A → D → B → C → E → H → F → G

정답 1번

간단한 해설

깊이우선탐색은 **한 경로를 끝까지 따라간 뒤 되돌아와 남은 이웃을 탐색**한다. 알파벳순을 적용하면 A → B → C → F → G → H → D를 먼저 방문하고, B로 되돌아온 뒤 E를 방문한다.

NoteBase

계산 과정

깊이우선탐색은 **현재 정점에서 방문하지 않은 이웃을 골라 먼저 깊이 들어가고**, 더 진행할 수 없으면 이전 정점으로 되돌아간다. 이 문제에서는 선택 가능한 이웃 중 알파벳이 앞선 것을 고른다.

A의 이웃 B와 D 중 B를 먼저 방문한다. B에서는 A를 이미 방문했으므로 C와 E 중 C를 고른다. C에서는 F와 H 중 F를 선택한다. 따라서 여기까지의 순서는 **A → B → C → F**이다.

F에서는 G로, G에서는 H로 진행한다. H의 이웃 C와 G는 이미 방문했으므로 아직 방문하지 않은 D를 방문한다. 이때까지의 순서는 **A → B → C → F → G → H → D**이다.

D에서 더 갈 곳이 없으므로 H, G, F, C를 거쳐 B로 되돌아온다. 되돌아가는 과정의 정점을 새 방문으로 다시 세지 않는다. B에 남아 있는 미방문 이웃 E를 마지막으로 방문한다.

따라서 전체 방문 순서는 **A → B → C → F → G → H → D → E**이며, 정답은 ①이다.

NoteBase

문 22. 다음 데이터 정렬에 사용된 알고리즘으로 옳은 것은?

단계	데이터
정렬 전	3 4 1 2
↓	3 4 1 2
	3 1 4 2
	3 1 2 4
	1 3 2 4
	1 2 3 4
정렬 후	1 2 3 4

① 힙(heap) 정렬

② 버블(bubble) 정렬

③ 선택(selection) 정렬

④ 삽입(insertion) 정렬

정답 2번

간단한 해설

이웃한 두 원소를 차례로 비교·교환하면서 큰 값을 뒤로 보내는 버블 정렬이다. 첫 순회에서 최댓값 4가 맨 뒤에 놓이고, 다음 순회에서 3이 그 앞에 놓인다.

NoteBase

도식 해설

첫 번째 순회에서는 이웃한 3과 4는 그대로 두고, 4와 1을 교환하여 3 1 4 2로 만든 뒤, 4와 2를 교환하여 3 1 2 4로 만든다. 이처럼 **인접한 원소를 차례로 비교·교환하여 가장 큰 값 4를 맨 뒤로 보내는 과정**은 버블 정렬의 특징이다.

두 번째 순회에서는 3과 1을 교환하여 1 3 2 4로, 이어 3과 2를 교환하여 1 2 3 4로 만든다. 다음 비교에서는 1과 2가 이미 오름차순이므로 그대로 유지된다. 따라서 사용된 알고리즘은 ② 버블 정렬이다.

선택 정렬은 미정렬 구간에서 최솟값 등을 선택해 제자리에 놓고, 삽입 정렬은 현재 원소를 앞쪽 정렬 구간의 알맞은 위치에 삽입한다. 힙 정렬은 힙의 루트 원소를 꺼내 정렬한다. 제시 과정처럼 이웃한 원소를 연속해서 비교하며 큰 값을 뒤로 밀어내는 방식과 구별한다.

NoteBase

문 23. 정렬 알고리즘에 관한 설명으로 옳지 않은 것은? (단, n은 데이터의 개수이다.)

① 퀵 정렬(quick sort) 알고리즘에서 최악의 경우 시간복잡도는 $O(n^2)$이다.

② 퀵 정렬 알고리즘은 우선순위 큐(priority queue)를 이용한다.

③ 합병 정렬(merge sort) 알고리즘의 시간복잡도는 $O(n\log_2 n)$이다.

④ 합병 정렬 알고리즘은 재귀 함수(recursive function) 기반 분할 정복 기법을 이용할 수 있다.

정답 2번

간단한 해설

퀵 정렬은 **피벗을 기준으로 자료를 나누는 분할 정복 알고리즘**이다. 우선순위 큐에서 우선순위가 높은 원소를 반복해서 추출하는 정렬 방식과는 구별된다.

선지별 해설

① O: 퀵 정렬에서 피벗을 기준으로 분할한 두 부분의 크기가 매번 0과 나머지 전체처럼 극단적으로 불균형하면, 분할 횟수가 많아진다. 비교 횟수가 대략 $(n-1)+(n-2)+\cdots+1$이 되어 최악의 시간복잡도는 $O(n^2)$이다.

② X: 퀵 정렬은 **피벗을 기준으로 자료를 분할**하고 각 부분을 다시 정렬하는 방식이다. 우선순위 큐에서 최댓값이나 최솟값을 반복해서 꺼내는 방식은 힙을 이용한 힙 정렬과 관련되며, 퀵 정렬의 기본 원리가 아니다.

③ O: 표준 합병 정렬은 자료를 절반씩 나누므로 분할 단계가 $O(\log_2 n)$개이고, 각 단계에서 합병에 총 $O(n)$의 시간이 든다. 따라서 전체 시간복잡도는 $O(n\log_2 n)$이다.

④ O: 합병 정렬은 자료를 작은 부분으로 나누고, 각 부분을 정렬한 뒤 합치는 분할 정복 방식이다. 두 부분 배열을 재귀 호출로 정렬한 후 합병하는 형태로 구현할 수 있다.

NoteBase

문 24. 알고리즘 설계 기법에 관한 설명으로 옳지 않은 것은?

① 그리디(greedy) 알고리즘은 부분 문제의 최적해를 계산하더라도 전체 문제의 최적해 계산을 보장할 수 없다.

② 그리디 알고리즘은 동전 교환 문제(coin change problem)의 최적해(최소 동전 개수)를 항상 계산할 수 있다.

③ 동적 계획(dynamic programming) 알고리즘은 부분 문제의 최적해로 전체 문제의 최적해를 찾는 알고리즘이다.

④ 동적 계획 알고리즘은 부분 문제의 해를 찾아 저장한 후 큰 문제를 해결하는 과정에서 이를 사용한다.

정답 2번

간단한 해설

큰 동전부터 고르는 방식은 **동전 종류에 따라 최적해를 보장하지 못한다**. 1원·3원·4원으로 6원을 만들면 그리디의 3개보다 3원짜리 2개가 더 적다.

선지별 해설

① O: 그리디는 매 단계에서 당장 가장 유리한 선택을 하는 방식이다. **지역적으로 최적인 선택이 전체의 최적해를 보장하지는 않는다**. 전체 최적성을 보이려면 그리디 선택이 최적해에 포함될 수 있다는 성질 등을 확인해야 한다.

② X: 큰 동전부터 선택하는 그리디가 항상 최소 개수를 주는 것은 아니다. 예를 들어 동전이 1원·3원·4원이고 6원을 만들 때, 그리디는 4+1+1로 3개를 쓰지만 **3+3은 2개**로 가능하다. 따라서 동전 체계에 따라 그리디는 최적해를 놓칠 수 있다.

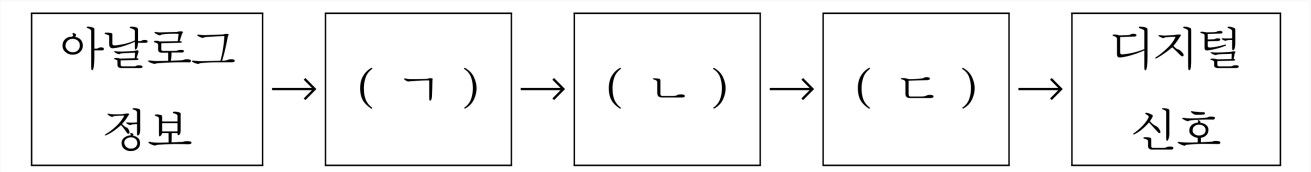
③ O: 최적화 문제에 동적 계획법을 적용하려면 전체 최적해를 부분 문제의 최적해로 구성할 수 있는 **최적 부분 구조**가 필요하다. 적절한 상태와 점화식으로 부분 문제의 최적값을 결합하여 전체 최적값을 구한다.

④ O: 동적 계획법은 계산한 부분 문제의 해를 **저장하여 다시 사용**함으로써 중복 계산을 줄인다. 작은 문제부터 표를 채우는 방식이나, 재귀 호출의 결과를 저장하는 메모이제이션 방식으로 구현할 수 있다.

따라서 옳지 않은 설명은 ②이다.

NoteBase

문 25. 다음 펄스 부호 변조 과정에서 ( ) 안에 들어갈 용어로 옳은 것은?



① ㄱ: 표본화, ㄴ: 양자화, ㄷ: 부호화

② ㄱ: 양자화, ㄴ: 부호화, ㄷ: 표본화

③ ㄱ: 표본화, ㄴ: 부호화, ㄷ: 양자화

④ ㄱ: 양자화, ㄴ: 표본화, ㄷ: 부호화

정답 1번

간단한 해설

펄스 부호 변조는 **표본화 → 양자화 → 부호화** 순서로 아날로그 정보를 디지털 신호로 바꾼다. 시간상의 표본을 추출한 뒤 진폭을 단계화하고, 그 값을 이진 부호로 표현한다.

올바른 순서

아날로그 정보 → **표본화(ㄱ)** → **양자화(ㄴ)** → **부호화(ㄷ)** → 디지털 신호

표본화는 연속적인 아날로그 신호의 값을 일정한 시간 간격으로 추출하는 과정이다. 먼저 어느 시점의 신호값을 사용할지 정하므로 시간축에서 값을 뽑는 단계가 앞선다.

양자화는 추출한 표본의 진폭을 미리 정한 유한한 단계의 값으로 대응시키는 과정이다. 표본값과 양자화된 값 사이에는 양자화 오차가 생길 수 있다.

부호화는 양자화된 각 단계의 값을 0과 1로 이루어진 이진 부호로 표현하는 과정이다. 표현할 진폭 단계가 정해져야 그 단계에 대응하는 부호를 부여할 수 있다.

NoteBase

문 26. 네트워크 장비에 관한 설명으로 옳지 않은 것은?

① 더미허브(dummy hub)는 호스트가 증가하면 속도가 감소한다.

② 라우터(router)는 독립적인 네트워크를 구성하여 인터넷에 연결할 수 있다.

③ 스위치(switch)는 목적지 주소로 패킷을 스위칭하여 네트워크 효율을 높인다.

④ 리피터(repeater)는 랜(LAN)을 두 개 이상 연결하여 네트워크를 하나로 만든다.

정답 4번

간단한 해설

리피터는 약해진 신호를 재생해 전송 거리를 늘리는 물리 계층 장비이다. 독립된 LAN들을 프레임 단위로 연결하는 브리지의 역할과 구별해야 한다.

선지별 해설

① (O) 더미허브는 들어온 신호를 다른 포트로 전달하며 연결된 호스트들이 하나의 충돌 영역과 전송 대역폭을 공유한다. 동시에 통신하는 호스트가 많아지면 충돌과 경쟁이 증가하여 호스트당 실효 전송 속도가 감소할 수 있다.

② (O) 라우터는 서로 다른 IP 네트워크 사이에서 목적지 IP 주소와 라우팅 정보를 바탕으로 패킷을 전달한다. 내부 네트워크를 외부 네트워크나 인터넷과 연결하는 데 사용할 수 있다.

③ (O) 일반적인 이더넷 스위치는 목적지 MAC 주소를 보고 프레임을 필요한 포트로 전달한다. 모든 포트가 전송 매체를 공유하는 더미허브와 달리 포트별로 충돌 영역을 분리하여 네트워크 효율을 높인다. 선지의 패킷은 여기서 전달되는 데이터 단위를 넓게 표현한 것이다.

④ (X) 리피터의 주된 기능은 **신호 재생과 전송 거리 연장**이며, 주소를 판단해 프레임을 선별 전달하지 않는다. LAN들을 데이터 링크 계층에서 연결하는 역할은 브리지나 스위치로 설명해야 한다. 리피터가 물리적인 LAN 구간을 이어 줄 수 있다는 사실과 독립 LAN을 논리적으로 연결하는 기능을 혼동하면 안 된다.

NoteBase

문 27. IP 주소를 MAC 주소(물리 주소)로 변환하는 프로토콜로 옳은 것은?

① DNS(Domain Name System)

② UDP(User Datagram Protocol)

③ ARP(Address Resolution Protocol)

④ TCP(Transmission Control Protocol)

정답 3번

간단한 해설

IPv4 네트워크에서 IP 주소에 대응하는 **MAC 주소를 알아내는 프로토콜은 ARP**이다. 따라서 정답은 ③이다.

개념 구분

ARP: IP 주소 → MAC 주소

같은 링크에서 프레임을 전달할 상대의 MAC 주소를 알아내는 데 사용한다. 목적지가 다른 네트워크에 있으면 원격 목적지의 MAC 주소가 아니라, 같은 링크에 있는 다음 홉 라우터의 MAC 주소를 알아내어 프레임을 전달한다.

DNS: 도메인 이름 → IP 주소

DNS는 도메인 이름에 대응하는 IP 주소 등의 정보를 조회하는 시스템이다. IP 주소에 대응하는 MAC 주소를 찾는 ARP와는 조회 대상이 다르다.

UDP와 TCP는 응용 프로그램 사이의 데이터 전송을 담당하는 전송 계층 프로토콜이며, IP 주소에서 MAC 주소를 찾는 역할을 하지 않는다.

NoteBase

문 28. 공중망(public network)에 사설망(private network)을 구성하여 전용망 사용 효과를 제공하는 기술로 옳은 것은?

① VPN(Virtual Private Network)

② CDN(Content Delivery Network)

③ GAN(Generative Adversarial Network)

④ HRN(Highest Response-ratio Next) 스케줄링

정답 1번

간단한 해설

VPN은 공중망 위에 논리적인 사설망을 구성하여 전용망을 사용하는 것과 같은 연결 효과를 제공한다. 물리적인 전용 회선을 별도로 두는 대신 기존 공중망을 활용한다.

개념 구분

VPN(가상 사설망): 인터넷 같은 공중망 위에 논리적인 사설 통신 경로를 구성하는 기술이다. 터널링 등을 이용하여 떨어진 지점이나 사용자를 사설망처럼 연결하므로, 전용망 사용 효과를 묻는 조건에 해당한다.

CDN(콘텐츠 전송망): 여러 지역의 서버에 콘텐츠를 분산·캐싱하여 사용자에게 가까운 곳에서 전달하는 기술이다. 콘텐츠 전송 속도와 서버 부하를 개선하는 것이 목적이며, 가상 사설망 구성과는 다르다.

GAN(생성적 적대 신경망): 생성자와 판별자를 경쟁적으로 학습시켜 데이터를 생성하는 인공지능 모형이다. 여기서 네트워크는 신경망을 뜻하며 통신용 사설망을 의미하지 않는다.

HRN 스케줄링: 대기 시간과 서비스 시간을 반영한 응답률이 가장 높은 작업을 먼저 선택하는 CPU 스케줄링 방식이다. 공중망 위의 사설망 구성과는 관계가 없다.

NoteBase

문 29. 5G 이동통신에 관한 내용으로 옳은 것만을 <보기>에서 고른 것은?

<보기>
ㄱ. 저지연 고신뢰 무선 전송
ㄴ. 초광대역 GHz급 전송 지원
ㄷ. 상하향 상이한 주파수 대역 사용
ㄹ. 반이중(half-duplex) 전송 방식 사용

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄴ, ㄹ

정답 1번

출제 포인트

5G의 저지연·고신뢰 및 초광대역 전송 특성과 상향·하향 통신을 구분하는 이중화 방식을 묻는다.

핵심 개념

5G는 고속 데이터 전송, 짧은 통신 지연, 높은 신뢰성 등을 지원하는 5세대 이동통신이다.
FDD는 상향과 하향에 서로 다른 주파수 대역을 배정하는 주파수 분할 이중화이다. TDD는 같은 주파수 대역에서 상향과 하향의 전송 시간을 나누는 시분할 이중화이다.
반이중은 양방향 전송이 가능하지만 동시에 송수신하지 못하는 방식이고, 전이중은 동시에 송수신할 수 있는 방식이다. 이중화 방식과 동시 송수신 가능 여부를 구분한다.

보기 판단

ㄱ O: 저지연·고신뢰 통신(URLLC)은 5G의 주요 지원 기능이다. 데이터를 짧은 지연 시간 안에 높은 성공 확률로 전달하는 것을 목표로 한다.
ㄴ O: 5G는 고주파 대역에서 넓은 대역폭을 확보하여 초광대역 전송을 지원한다. IMT-2020 성능 요구사항에는 고주파 대역에서 최대 1GHz의 대역폭 지원이 포함된다. 여기서 GHz는 대역폭을 나타내며 데이터 전송률의 단위와 다르다.
ㄷ X: “상하향 상이한 주파수 대역 사용”은 FDD의 설명이다. 5G 전체의 공통 특징으로 단정할 수 없다.
올바르게는 5G는 FDD와 TDD를 지원하며, TDD에서는 같은 주파수 대역을 시간으로 나누어 사용한다.
ㄹ X: “반이중 전송 방식 사용”으로 5G 전체의 전송 방식을 한정하는 것이 잘못이다. 올바르게는 5G는 FDD와 TDD를 지원하고, 전이중 FDD에서는 상향과 하향의 동시 송수신이 가능하다. TDD가 시간을 나누어 송수신한다는 사실을 5G 전체의 반이중 특성으로 일반화할 수 없다.

NoteBase

선지별 해설

① O: ㄱ의 저지연·고신뢰 통신과 ㄴ의 초광대역 전송 지원이 모두 옳다.
② X: ㄱ은 옳지만 ㄷ은 FDD의 특징을 5G 전체에 일반화한 설명이다. ㄷ을 빼고 ㄴ을 포함한 "ㄱ, ㄴ"으로 고쳐야 한다.
③ X: ㄴ은 옳지만 ㄷ은 5G가 지원하는 TDD의 특성을 반영하지 못한다. ㄷ을 ㄱ으로 바꾼 "ㄱ, ㄴ"으로 고쳐야 한다.
④ X: ㄴ은 옳지만 ㄹ은 5G를 반이중 방식으로 한정한다. ㄹ을 ㄱ으로 바꾼 "ㄱ, ㄴ"으로 고쳐야 한다.

오답 포인트

ㄷ의 "상이한 주파수 대역"은 FDD를 떠올리게 하지만 5G는 TDD도 지원한다. 반대로 TDD의 시간 분할만 보고 ㄹ의 "반이중"을 5G 전체의 특성으로 받아들이는 것도 함정이다.

다시 나오면 이렇게 판단

저지연·고신뢰와 광대역은 5G의 지원 기능으로 판단한다. 상향·하향에 다른 주파수를 쓰면 FDD, 같은 주파수에서 시간을 나누면 TDD로 구분한다. 특정 방식의 특징을 이동통신 세대 전체의 필수 특징으로 단정했는지 확인한다.

암기 포인트

5G는 저지연·고신뢰·초광대역을 지원한다. FDD는 주파수 분할, TDD는 시간 분할이다.

NoteBase

문 30. 다음 기술에 관한 설명으로 옳은 것은?

· 태그, 안테나, 리더기 등으로 구성된다.
· 제2차 세계 대전 당시 전투기의 피아를 식별하려고 개발하였다.
· 무선 주파수를 이용하여 비접촉으로 정보를 읽는 인식 시스템이다.

① 사각형 2차원 격자무늬를 스캔하여 정보를 얻는다.
② 태그값을 복제할 수 있다는 보안 취약점이 존재한다.
③ 약 2.4 GHz 주파수 대역을 사용하는 무선통신 기술이다.
④ 굵기가 다른 수직 막대의 조합을 광학적 판독으로 읽어 낸다.

정답 2번

출제 포인트

태그·안테나·리더기로 구성되어 무선 주파수로 정보를 비접촉 인식하는 RFID의 특징을 QR 코드·바코드·블루투스와 구분하는 문제이다.

핵심 개념

RFID는 전파를 이용해 태그에 저장된 식별 정보 등을 리더기로 읽는 기술이다. QR 코드는 사각형의 2차원 격자무늬를, 1차원 바코드는 굵기가 다른 막대와 간격을 광학적으로 읽는다. 블루투스는 기기 간 데이터를 주고받는 무선통신 기술이며 2.4GHz 대역을 사용한다.

선지별 해설

① X. '사각형 2차원 격자무늬를 스캔'하는 것은 QR 코드의 설명이다. RFID는 인쇄된 무늬를 광학적으로 스캔하는 대신 태그와 리더기 사이의 전파로 정보를 읽는다.
② O. 정당한 태그에서 식별값 등을 읽어 다른 태그에 복사하면 원래 태그를 모방할 수 있다. 복제 방지나 인증이 충분하지 않은 RFID 시스템에서 발생할 수 있는 보안 취약점이다. 모든 태그가 항상 복제된다는 뜻은 아니다.
③ X(이 문항의 정답 선택 기준). '약 2.4GHz 주파수 대역을 사용하는 무선통신 기술'은 블루투스의 대표적인 설명이다. 기술들을 대비하는 문맥에서 이 선지를 블루투스에 연결하면, RFID의 태그 복제 취약점을 설명한 ②가 가장 직접적인 선택이다. 다만 RFID에도 이 대역을 사용하는 방식이 있으므로 ③의 주파수 자체가 거짓이라고 단정할 수는 없다. 바르게 정리하면 RFID는 방식에 따라 여러 주파수 대역을 사용하며, 특정 대역만으로 블루투스와 구분할 수 없다. NIST는 RFID 사용 대역에 2.45GHz도 제시한다.
④ X. '굵기가 다른 수직 막대의 조합을 광학적 판독'으로 읽는 것은 1차원 바코드의 설명이다. RFID는 막대무늬의 광학적 판독이 아니라 태그의 정보를 전파로 읽는 방식이다.

NoteBase

오답 포인트

①과 ④는 정보를 식별한다는 목적은 비슷하지만 판독 방식이 광학식이다. ③의 주파수는 블루투스에서 사용하는 대역이지만 RFID에서도 사용 가능하므로 'RFID는 이 대역을 쓰지 않는다'라고 외우면 틀린다. 이 문항에서는 지문의 태그·리더기라는 단서와 ②의 복제 취약점을 연결한다.

다시 나오면 이렇게 판단

먼저 태그·리더기와 비접촉 전파 인식이라는 단서로 RFID를 식별한다. 이후 광학적 무늬 판독과 구분하고, 태그 식별값의 복제 가능성을 확인한다. 주파수만 제시된 선지는 다른 기술과 대역을 공유할 수 있는지 함께 판단한다.

암기 포인트

RFID는 전파와 태그, QR 코드는 2차원 격자, 바코드는 막대무늬이다. RFID의 태그값 복제는 보안 취약점이며, 주파수 대역만으로 기술을 단정하지 않는다.

NoteBase

문 31. 다음 설명에 해당하는 네트워크 모델로 옳은 것은?

· 수많은 센서를 서로 무선으로 연결하여 구성하는 네트워크이다.
· 센서가 수집한 정보를 주변의 다른 센서로 전파한다.

① 소셜 네트워크 모델
② 링(ring) 네트워크 모델
③ 버스(bus) 네트워크 모델
④ 애드혹(ad-hoc) 네트워크 모델

정답 4번

간단한 해설

센서들이 무선으로 서로 연결되어 **이웃 센서를 통해 정보를 전달하는 자율적 구조**가 핵심이다. 이는 애드혹 네트워크 모델의 특징이다.

핵심 풀이

애드혹 네트워크는 단말들이 고정된 통신 기반시설에만 의존하지 않고 **서로 직접 통신하여 자율적으로 구성하는 네트워크**이다. 필요한 경우 중간 단말이 정보를 전달해 여러 구간에 걸쳐 통신할 수 있다.

제시문에서 센서들은 무선으로 연결되고 수집한 정보를 이웃 센서에 전파한다. 각 센서가 정보를 생성할 뿐 아니라 다른 센서와 전달 경로를 이루는 구조이므로 **애드혹 네트워크 모델**에 해당한다. 정답은 ④이다.

링과 버스는 각각 고리 형태의 연결과 공통 통신선에 접속하는 형태를 나타내는 네트워크 토폴로지이다. 소셜 네트워크는 사람이나 집단 사이의 관계를 나타내는 개념이므로 제시문의 센서 간 무선 전달 방식과 구분된다.

NoteBase

문 32. 운영체제가 제공하는 세마포(semaphore)의 기능으로 옳은 것은?

① 메모리 할당

② 데이터 암호화

③ 프로세스 비동기화

④ 공유 자원 접근 제어

정답 4번

간단한 해설

세마포는 여러 프로세스나 스레드의 **공유 자원 접근을 제어하는 동기화 도구**이다. 자원을 사용할 수 있는지 확인하고 필요하면 기다리게 하여 동시 접근에 따른 문제를 조절한다.

핵심 풀이

세마포의 핵심 기능: 여러 프로세스나 스레드가 공유 자원에 접근하는 순서와 수를 조절하는 동기화 도구이다. 따라서 제시된 기능 중 공유 자원 접근 제어에 해당한다.

동작 원리: 원자적으로 수행되는 wait(P)와 signal(V) 연산을 이용한다. 사용할 자원이 있으면 접근을 허용하고, 없으면 기다리게 하며, 사용이 끝나면 다른 실행 흐름이 접근할 수 있도록 알린다.

다른 기능과의 구별: 메모리 할당은 프로세스 등에 기억 공간을 배정하는 기능이고, 데이터 암호화는 정보를 보호하기 위해 변환하는 기능이다. 세마포는 이 두 기능 자체를 수행하는 도구가 아니며, 프로세스를 비동기화하는 것이 아니라 **필요한 동기화와 접근 제어**를 제공한다.

NoteBase

문 33. 우선순위 스케줄링(priority scheduling)에 관한 설명으로 옳지 않은 것은?

① 각 프로세스의 상대적 중요성을 정의할 수 있다.

② 실행 시간이 다양하여 실시간 시스템에 사용할 수 없다.

③ 우선순위가 동일한 프로세스는 선입선처리 순서로 실행할 수 있다.

④ 노화(aging) 기법을 이용하여 낮은 우선순위 프로세스의 기아(starvation) 문제를 해결할 수 있다.

정답 2번

간단한 해설

우선순위 스케줄링은 중요한 작업이나 시간 제약이 있는 작업에 높은 우선순위를 부여할 수 있어 **실시간 시스템에도 사용**된다. 실행 시간이 다양하다는 이유만으로 사용할 수 없다는 ②가 옳지 않다.

선지별 해설

① O: 프로세스마다 우선순위를 부여하여 상대적인 중요성이나 긴급성을 나타낼 수 있다. 준비 상태의 프로세스 중 높은 우선순위의 프로세스를 우선 선택한다.

② X: 우선순위 스케줄링은 **실시간 시스템에도 활용할 수 있다**. 예를 들어 주기나 마감 시각을 기준으로 우선순위를 정하여 시간 제약이 있는 작업을 먼저 처리할 수 있다. 다만 실시간 마감 준수를 보장하려면 실행 시간과 작업 부하 등에 대한 별도의 분석이 필요하며, 우선순위 부여만으로 항상 보장되는 것은 아니다.

③ O: 같은 우선순위를 가진 프로세스 사이에는 선입선처리(FCFS)와 같은 보조 기준을 적용할 수 있다. 이 경우 먼저 준비 큐에 들어온 프로세스를 먼저 실행한다.

④ O: 높은 우선순위의 작업이 계속 도착하면 낮은 우선순위의 작업이 무기한 기다리는 기아 현상이 발생할 수 있다. **노화 기법**은 오래 기다린 프로세스의 우선순위를 점차 높여 실행 기회를 주는 방법이다.

NoteBase

문 34. 다음 리눅스(Linux) 명령어 실행 결과에 대한 설명으로 옳지 않은 것은?

```
>> ls -al
drwxr-xr-- 2 user group 4096 Feb 20 12:34 sample
-rwxr-xr-- 1 user group 1024 Feb 20 12:34 script.sh
-rw-r--r-- 1 user group 4096 Feb 20 12:34 document.txt
```

① ls -al 명령어는 2020년 2월에 실행되었다.

② sample은 디렉터리(directory)이다.

③ script.sh는 소유자가 읽기, 쓰기, 실행 권한을 가지고 있다.

④ document.txt는 실행할 수 없는 파일이다.

정답 1번

간단한 해설

ls -l의 날짜·시각은 기본적으로 **파일의 최종 수정 시각**이다. Feb 20의 20은 날짜이므로 2020년이나 명령 실행 연도로 해석할 수 없다.

NoteBase

선지별 해설

ls -al에서 -a는 숨김 항목을 포함하여 나열하는 옵션이고, -l은 파일 종류·권한·소유자·크기·시각 등을 긴 형식으로 표시하는 옵션이다. 맨 앞 문자는 파일 종류이며, 뒤의 아홉 문자는 소유자·그룹·기타 사용자의 권한을 세 문자씩 나타낸다.

① X: Feb 20 12:34는 기본적으로 해당 항목의 **최종 수정 시각인 2월 20일 12시 34분**을 뜻한다. 20은 연도가 아니라 날짜이며, ls 명령을 실행한 시각을 기록한 것도 아니다. 이 출력으로 2020년 2월에 실행했다고 판단할 수 없다.

② O: sample의 권한 문자열은 drwxr-xr--이며, 맨 앞의 **d는 디렉터리**를 뜻한다. 일반 파일을 나타내는 첫 문자 -와 구별한다.

③ O: script.sh의 문자열 -rwxr-xr--에서 첫 문자 - 다음의 세 문자 rwx가 소유자 권한이다. **r은 읽기, w는 쓰기, x는 실행**이므로 소유자는 세 권한을 모두 가진다.

④ O: document.txt의 문자열 -rw-r--r--는 소유자 rw-, 그룹 r--, 기타 사용자 r--로 나뉜다. 어느 묶음에도 x가 없으므로 **표시된 권한으로는 파일 자체를 직접 실행할 수 없다**. 이는 읽기 가능한 내용을 인터프리터 등에 입력하는 것과는 구별되는 실행 권한 판단이다.

따라서 옳지 않은 설명은 ①이다.

NoteBase

문 35. 다음 설명에 해당하는 공격의 명칭으로 옳은 것은?

· 공격자는 감염된 여러 좀비 컴퓨터로 공격을 수행한다.
· 허용치를 초과하는 통신 데이터나 메시지를 목표 시스템으로 전송한다.

① 스미싱(smishing) 공격
② IP 스푸핑(spoofing) 공격
③ APT(Advanced Persistent Threat) 공격
④ DDoS(Distributed Denial-of-Service) 공격

정답 4번

간단한 해설

여러 좀비 컴퓨터가 동시에 과도한 트래픽을 보내 서비스 이용을 방해한다는 점이 DDoS의 핵심이다. 공격원이 여러 곳에 분산되어 있다는 단서를 놓치지 않는다.

핵심 풀이

DDoS는 **분산 서비스 거부 공격**이다. 여러 장치에서 목표 시스템으로 많은 통신 요청이나 데이터를 보내 네트워크 대역폭 또는 서버 자원을 소진시키고, 정상 사용자가 서비스를 이용하지 못하게 한다.

제시문의 여러 좀비 컴퓨터는 공격원이 분산되어 있다는 단서이고, 허용치를 넘는 데이터 전송은 서비스 자원에 과부하를 준다는 단서이다. 두 특징을 합치면 **DDoS 공격**이므로 정답은 ④이다.

스미싱은 문자메시지를 이용해 악성 링크 접속이나 정보 입력 등을 유도하는 공격이다.

IP 스푸핑은 출발지 IP 주소를 위조하는 기법이며,

APT는 특정 대상을 상대로 은밀하고 지속적으로 침투하는 공격 양상이다.

이들은 제시문의 분산된 대량 트래픽에 의한 서비스 방해라는 핵심과 구분된다.

NoteBase

문 36. 개인정보 비식별 방법으로 옳지 않은 것은?

① 가명 처리

② 총계 처리

③ 데이터 복호화

④ 데이터 범주화

정답 3번

간단한 해설

복호화는 암호문을 원래 데이터로 되돌리는 과정이므로 개인의 식별 가능성을 낮추는 비식별 방법이 아니다. 가명 처리·총계 처리·범주화는 식별 정보를 대체하거나 상세도를 줄이는 데 활용된다.

선지별 해설

① (O) 가명 처리는 이름 등 식별 가능한 값을 다른 이름이나 기호로 대체하여 그 정보만으로 개인을 바로 알아보기 어렵게 하는 방법이다. 추가 정보와 결합하면 다시 식별될 가능성이 있으므로 완전한 익명화와 같다고 볼 수는 없다.

② (O) 총계 처리는 개인별 값을 그대로 드러내는 대신 집단의 합계·평균 등 통계값으로 나타내어 개별 정보의 노출을 줄이는 방법이다. 특정 개인이 드러나지 않도록 집단의 크기와 구성도 고려해야 한다.

③ (X) 데이터 복호화는 암호화된 데이터를 해독하여 원래의 평문으로 복원하는 과정이다. **개인 식별 정보를 대체하거나 상세도를 낮추는 처리가 아니므로** 비식별 방법에 해당하지 않는다.

④ (O) 데이터 범주화는 정확한 값을 일정 범위나 상위 범주로 바꾸는 방법이다. 예를 들어 나이를 정확한 숫자 대신 연령대로 나타내면 정보의 상세도가 낮아져 개인을 특정하기 어려워진다.

NoteBase

문 37. 드라이브 미러링(drive mirroring)을 사용하는 RAID 레벨로 옳은 것은?

① RAID 0

② RAID 1

③ RAID 4

④ RAID 5

정답 2번

간단한 해설

RAID 1은 같은 데이터를 중복 저장하는 드라이브 미러링을 사용한다. 한 드라이브에 장애가 발생해도 다른 드라이브에 남은 동일한 데이터로 서비스를 유지할 수 있다.

개념 구분

RAID 1: **동일한 데이터를 여러 드라이브에 중복 저장하는 미러링**을 사용한다. 한 드라이브가 고장 나도 정상 드라이브의 복사본으로 데이터를 사용할 수 있다. 같은 용량의 드라이브 두 개를 미러링하면 사용 가능한 저장 용량은 한 개의 용량에 해당한다.

RAID 0: 데이터를 나누어 여러 드라이브에 분산 저장하는 스트라이핑 방식이다. 미러링이나 패리티를 두지 않으므로, 드라이브 고장에 대비한 데이터 중복성이 없다.

RAID 4와 RAID 5: 두 방식 모두 블록 단위 스트라이핑과 패리티를 이용한다. RAID 4는 패리티를 전용 드라이브에 저장하고, RAID 5는 패리티를 여러 드라이브에 분산 저장한다. 데이터를 그대로 복제하는 미러링과는 구별된다.

NoteBase

문 38. 다음 ( ) 안에 들어갈 용어로 옳은 것은?

양자컴퓨터는 (ㄱ)과 (ㄴ) 등 양자물리학의 원리를 이용하여 다수의 정보를 동시에 연산할 수 있도록 구현된 새로운 개념의 컴퓨터이다.

① ㄱ: 간섭, ㄴ: 변형

② ㄱ: 간섭, ㄴ: 얽힘

③ ㄱ: 중첩, ㄴ: 얽힘

④ ㄱ: 중첩, ㄴ: 변형

정답 3번

간단한 해설

지문에서 여러 정보를 함께 다루는 양자 상태의 특징에 대응하는 ㄱ은 **중첩**, 여러 큐비트 사이의 양자적 상관관계에 해당하는 ㄴ은 **얽힘**이다. 따라서 출제 의도에 맞는 조합은 ③이다.

빈칸별 해설

ㄱ: 중첩

고전적인 비트는 0 또는 1의 값을 갖지만, 큐비트는 0과 1의 기저상태가 선형결합된 중첩 상태로 표현될 수 있다. 여러 큐비트의 상태에는 여러 비트열에 대응하는 성분이 포함될 수 있으므로, 지문의 '다수의 정보를 동시에 연산'이라는 설명은 이 중첩의 특징을 가리킨다.

ㄴ: 얽힘

여러 큐비트의 전체 상태가 각 큐비트의 독립적인 상태로 분리되지 않는 양자적 상관관계이다. 양자컴퓨터는 이러한 상태를 연산에 활용한다.

따라서 ㄱ은 중첩, ㄴ은 얽힘이며 ③과 연결된다. 다만 **간섭도 실제로 활용되는 양자컴퓨팅의 원리**이다. 간섭은 상태의 확률진폭을 강화하거나 상쇄하는 역할로 구별한다. 또한 중첩을 이용한다고 해서 모든 계산 결과를 한 번의 측정으로 동시에 읽을 수 있는 것은 아니다.

NoteBase

문 39. 빅데이터의 특징을 나타내는 3V로 옳지 않은 것은?

① Variety

② Velocity

③ Volume

④ Verification

정답 4번

간단한 해설

빅데이터의 기본 3V는 **Volume(규모), Velocity(속도), Variety(다양성)**이다. Verification은 이 세 특성에 포함되지 않는다.

선지별 해설

① O: Variety는 **데이터 형태와 출처의 다양성**이다. 표 형태의 정형 데이터뿐 아니라 문서·이미지·영상 등 비정형 데이터와 여러 구조의 데이터를 함께 다루는 특징을 뜻한다.

② O: Velocity는 **데이터가 생성·유입되고 처리되어야 하는 속도**이다. 센서나 온라인 서비스에서 빠르게 쌓이는 데이터를 필요한 시점에 처리해야 한다는 특성과 관련된다.

③ O: Volume은 **데이터 양의 규모**이다. 대량의 데이터를 저장하고 분석하기 위해 기존 방식만으로는 부족할 수 있다는 빅데이터의 대표적인 특징이다.

④ X: Verification은 검증을 뜻하는 말로, 기본 **3V인 Volume·Velocity·Variety에 포함되지 않는다**. 확장된 특성으로 쓰이는 Veracity(진실성·신뢰성)와도 다른 용어이다.

따라서 빅데이터의 3V로 옳지 않은 것은 ④ Verification이다.

NoteBase

문 40. 인공지능의 지도학습 알고리즘으로 옳은 것만을 나열한 것은?

① 선형 회귀, KNN(K-Nearest Neighbors)

② KNN, K-means 군집화

③ K-means 군집화, 차원 축소

④ 선형 회귀, K-means 군집화

정답 1번

간단한 해설

지도학습은 **정답이 주어진 데이터로 입력과 출력의 관계를 학습**한다. 선형 회귀와 KNN은 이에 해당하지만, K-means는 정답 없이 군집을 찾는 비지도학습이다.

선지별 해설

① (O) 선형 회귀는 입력과 정답 수치의 관계를 이용해 연속적인 값을 예측한다. KNN은 가까운 이웃 데이터의 정답 정보를 이용해 분류하거나 회귀값을 예측한다. 따라서 **둘 다 지도학습에 해당**한다.

② (X) KNN은 지도학습이지만, K-means는 정답 레이블 없이 데이터의 유사성에 따라 군집을 만드는 **비지도학습**이다. 지도학습만 나열하려면 K-means 대신 선형 회귀 등의 지도학습 알고리즘을 넣어야 한다.

③ (X) K-means 군집화는 비지도학습이다. 차원 축소는 데이터의 변수 수를 줄이는 작업으로, 대표적인 비지도 차원 축소 기법에는 주성분 분석(PCA)이 있다. 차원 축소라는 말만으로 모두 지도학습이라고 할 수도 없으므로 지도학습만의 조합이 아니다.

④ (X) 선형 회귀는 지도학습이지만 K-means 군집화는 비지도학습이다. 두 항목이 모두 지도학습이어야 하므로 이 조합은 틀리며, K-means를 KNN으로 바꾸면 지도학습의 조합이 된다.

NoteBase