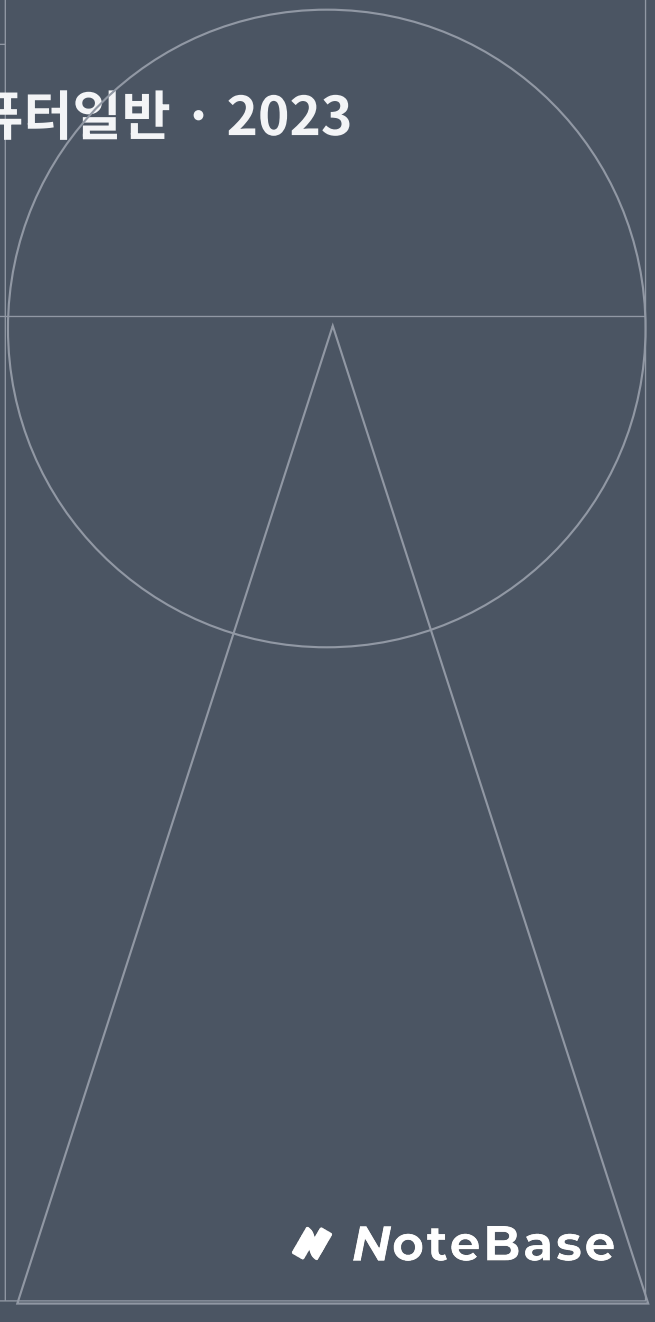
기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2023



NoteBase

기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2023

공무원/소방/경력채용 · 컴퓨터일반 · 2023

문 1. 시스템을 잠그거나 암호화해 컴퓨터나 파일을 사용할 수 없게 만들고, 이를 해소하는 대가로 금전을 요구하려고 퍼뜨리는 악성 프로그램은?

① 웜
② 피싱
③ 스미싱
④ 랜섬웨어

정답 4번

간단한 해설

시스템을 잠그거나 파일을 암호화한 뒤 **복구를 대가로 금전을 요구**하는 악성 프로그램은 '랜섬웨어'이다. 사용을 막아 놓고 몸값을 요구한다는 점이 결정적인 단서이다.

개념 구분

'랜섬웨어'는 시스템을 잠그거나 파일을 암호화하여 정상적인 이용을 막고, 잠금 해제나 복호화의 대가로 금전을 요구하는 악성 프로그램이다. 문제의 **사용 불능과 금전 요구**가 이 특징에 대응한다.

'웜'은 독립적으로 실행되며 **스스로 복제되어 네트워크로 전파**되는 프로그램이다. 자기 복제와 전파 방식에 초점을 둔 명칭이며, 복구 대가로 금전을 요구하는 것이 정의의 핵심은 아니다.

'피싱'은 신뢰할 만한 사람이나 기관을 사칭하여 이메일이나 가짜 웹사이트 등으로 개인정보 입력이나 악성 프로그램 설치를 유도하는 수법이다. 핵심은 **사칭을 통한 속임수와 유인**이다.

'스미싱'은 **문자메시지를 이용하는 피싱**이다. 메시지의 링크 등을 이용하여 개인정보를 빼내거나 악성 앱 설치를 유도한다.

제시된 내용은 피싱·스미싱이라는 유인 수법이나 웜의 자기 복제 특성보다, 사용을 막은 뒤 복구 대가를 요구하는 특징을 설명하므로 ④ '랜섬웨어'에 해당한다.

NoteBase

문 2. 다음에서 설명하는 컴퓨터 네트워크 서비스의 용어로 옳은 것은?

· 호스트의 도메인 이름에서 IP 주소를 얻는 서비스를 제공한다.
· 계층구조를 지원하는 도메인 기반의 주소 표기 방법을 위한 분산 데이터베이스 시스템이다.

① DNS
② DHCP
③ SNMP
④ SMTP

정답 1번

간단한 해설

도메인 이름으로 IP 주소를 조회하는 기능은 **DNS의 이름 해석**이다. DNS는 이름 공간을 계층적으로 구성하고 관련 정보를 여러 네임 서버에 나누어 저장하는 **분산 데이터베이스 시스템**이다.

핵심 풀이

DNS(Domain Name System)는 호스트의 도메인 이름에 대응하는 IP 주소 등의 정보를 제공하는 시스템이다. 지문의 '도메인 이름에서 IP 주소를 얻는 서비스'는 DNS의 **이름 해석 기능**을 뜻한다.

DNS의 도메인 이름 공간은 상하위 도메인이 연결된 **계층적 트리 구조**이다. 도메인 정보를 여러 관리 영역으로 나누고 각 영역을 담당하는 네임 서버들이 관리하므로, 지문의 '분산 데이터베이스 시스템'에도 해당한다.

DHCP는 호스트에 **IP 주소와 네트워크 설정 정보를 할당·제공**하는 프로토콜이다. 도메인 이름에 대응하는 IP 주소를 조회하는 DNS와 역할이 다르다.

SNMP는 네트워크 장비의 상태와 관리 정보를 조회하거나 설정하는 **네트워크 관리 프로토콜**이다.

SMTP는 **전자우편을 전송**하는 프로토콜이다.

따라서 이름 해석 기능과 계층적 분산 데이터베이스라는 두 설명에 모두 해당하는 것은 ① DNS이다.

NoteBase

문 3. C 프로그램 결과의 출력값으로 옳은 것은?

```
#include <stdio.h>
 int sum(int a, int b);
  void main( ) {
        int k = 10;
        k = sum(k, 50)/2;
        printf("%d", k);
  }
  int sum(int a, int b) {
        int tot;
        tot = b - a;
        return tot;
  }
```

① 20
② 25
③ 30
④ 35

정답 1번

간단한 해설

sum 함수는 이름과 달리 **b - a**를 반환하므로 sum(10, 50)은 40이다. 이를 2로 나눈 값 20이 k에 저장되어 출력된다.

NoteBase

계산 과정

초기값과 함수 호출: main에서 k는 10으로 초기화된다. sum(k, 50)을 호출하면 첫 번째 인수 10은 매개변수 a에, 두 번째 인수 50은 매개변수 b에 전달된다.

반환값 계산: 함수 이름이 sum이어도 실제 수행하는 연산은 **b - a**이다. 따라서 tot에는 50 - 10 = 40이 저장되고, return tot에 의해 40이 호출 위치로 반환된다.

대입과 출력: k = sum(k, 50)/2는 k = 40/2가 되어 k에 20을 저장한다. printf의 %d는 **정수 k의 값**을 십진수로 출력하므로 화면에 20이 표시된다.

따라서 출력값 20에 해당하는 선택지는 ①이다.

NoteBase

문 4. 임계 영역(critical section)에 관한 설명으로 옳지 않은 것은?

① A 프로세스가 임계 영역에서 실행하는 동안 B 프로세스는 A 프로세스의 임계 영역에 들어갈 수 없다.

② A 프로세스는 임계 영역에서 B 프로세스와 공유하는 변수를 동시에 접근하여 변경하는 등의 작업을 실행한다.

③ A 프로세스가 임계 영역에 진입하려는 요청을 한 후부터 허용될 때까지 B 프로세스가 A 프로세스의 임계 영역에 진입하도록 허용되는 횟수는 제한이 있다.

④ A 프로세스의 임계 영역에서 실행 중인 프로세스가 없고 A 프로세스의 임계 영역으로 진입하려고 하는 프로세스가 있다면, 진입할 프로세스 선택은 무기한 연기될 수 없다.

정답 2번

간단한 해설

임계 영역은 공유 변수를 다루는 구간이지만, 여러 프로세스가 **동시에 접근·변경하도록 허용하는 구간은 아니다**. 같은 공유 자원을 다룰 때 한 프로세스만 실행하도록 상호 배제를 보장해야 한다.

선지별 해설

임계 영역은 여러 프로세스가 공유하는 데이터나 자원에 접근하는 코드 구간이다. 같은 공유 자원을 다루는 임계 영역에서는 실행이 겹쳐 데이터가 잘못 갱신되지 않도록 동기화해야 한다.

① O: **상호 배제** 조건이다. A가 공유 자원을 다루는 임계 영역을 실행하는 동안 B는 같은 자원을 다루는 자신의 임계 영역에 진입할 수 없다. 선지의 'A 프로세스의 임계 영역'은 동일한 공유 자원에 대한 임계 영역을 가리키는 취지로 이해한다.

② X: 공유 변수를 읽거나 변경하는 코드가 임계 영역에 포함되는 것은 맞지만, A와 B가 그 작업을 **동시에 실행하도록 허용하는 구간은 아니다**. 한 프로세스가 실행하는 동안 다른 프로세스는 기다리도록 상호 배제를 보장해야 한다. 예를 들어 두 프로세스가 같은 값을 읽어 각각 증가시키면 한쪽 갱신이 사라질 수 있다.

③ O: **한정 대기** 조건이다. A가 진입을 요청한 뒤 허용되기 전까지 다른 프로세스가 먼저 임계 영역에 진입할 수 있는 횟수에 상한이 있어야 한다. B가 계속 새치기하여 A가 영원히 기다리는 기아 상태를 막는 조건이다.

④ O: **진행** 조건이다. 임계 영역이 비어 있고 진입을 원하는 프로세스가 있다면, 다음 진입자를 정하는 결정을 무기한 미룰 수 없다. 누구도 들어가지 못한 채 선택만 계속 지연되는 상황을 방지한다.

NoteBase

문 5. 소프트웨어 회귀 테스트에 관한 설명으로 옳은 것만을 <보기>에서 고른 것은?

<보기>
ㄱ. 확정 테스트(confirmation test)가 끝난 후 또 한 번 테스트한다.
ㄴ. 출시 전 최종 피드백을 위해 다수의 사용자 환경에서 테스트한다.
ㄷ. 수정된 모듈뿐 아니라 관련된 모듈까지 문제가 없는지 테스트한다.
ㄹ. 테스트 케이스의 일부분을 재실행할 수 있고, 도구를 이용해 자동으로 테스트할 수 있다.

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

정답 3번

간단한 해설

회귀 테스트는 변경 후 **기존 기능에 생긴 부작용**을 확인한다. 사용자 환경에서 출시 전 피드백을 받는 ㄴ은 **베타 테스트**에 해당한다.

보기별 해설

ㄱ. O: 결함을 수정한 경우 확정 테스트로 **원래 결함이 해결되었는지** 확인한 뒤, 회귀 테스트로 수정에 따른 부작용을 확인한다. 회귀 테스트는 결함 수정 외에 기능 추가나 실행 환경 변경 후에도 필요할 수 있다.

ㄴ. X: 출시 전에 여러 사용자가 자신의 환경에서 제품을 사용하고 피드백을 제공하는 것은 **베타 테스트**이다. 회귀 테스트의 목적은 사용자 의견 수집이 아니라 변경 때문에 기존 기능에 문제가 생겼는지 확인하는 데 있다.

ㄷ. O: 수정의 영향은 모듈 사이의 호출이나 데이터 전달을 통해 퍼질 수 있다. 따라서 **수정된 모듈과 관련 모듈**의 기존 기능이 계속 정상적으로 동작하는지 검사한다.

ㄹ. O: 변경의 영향 범위를 고려하여 기존 테스트 케이스 중 필요한 일부를 선정해 재실행할 수 있다. 같은 검사를 반복하는 특성이 있으므로 **테스트 도구로 자동화**하기에도 적합하다.

옳은 보기는 ㄱ, ㄷ, ㄹ이므로 정답은 ③이다.

NoteBase

문 6. 다음에서 설명하는 NoSQL의 유형으로 옳은 것은?

· 몽고 DB, 카우치 DB 등이 대표적이다.
· 키와 문서의 쌍으로 데이터를 저장한다.
· 계층적 구조가 존재하는 JSON, XML 등과 같은 반정형 형태의 문서로 데이터를 저장한다.
· 키를 통해 문서 전체를 검색하는 것도 가능하지만 XQuery와 같은 특별한 문서 대상 질의 언어를 이용하면 문서 내의 일부를 검색하거나 질의에 활용할 수 있다.

① 키 값(key value) 데이터베이스
② 컬럼 기반(column based) 데이터베이스
③ 그래프 기반(graph based) 데이터베이스
④ 문서 기반(document based) 데이터베이스

정답 4번

간단한 해설

키에 대응하는 값을 **문서 형태로 저장**하고, 문서 내부 필드와 계층 구조까지 질의하는 것은 문서 기반 데이터베이스의 특징이다. MongoDB와 CouchDB가 대표적이다.

NoteBase

개념 구분

문서 기반 데이터베이스는 식별 키에 대응하는 **구조화된 문서 자체를 저장 단위**로 삼는다. 문서 안에는 필드·배열·중첩 구조가 들어갈 수 있고, 문서 전체뿐 아니라 내부의 특정 필드를 조건으로 검색할 수 있다. JSON·XML 같은 반정형 문서라는 설명이 이 유형을 가리킨다.

대표 제품인 MongoDB는 BSON 문서, CouchDB는 JSON 문서를 사용한다. 지문의 XQuery는 **XML 문서의 내용과 구조를 질의하는 언어의 예**이며, MongoDB와 CouchDB가 모두 XQuery를 쓴다는 뜻은 아니다.

키 값 데이터베이스는 키에 대응하는 값을 저장·조회하는 것이 기본이다. 키를 쓴다는 점만으로 이를 고르면 안 되며, 이 문항에서는 **문서 내부 구조를 대상으로 질의**한다는 점이 문서 기반 유형을 구별한다.

컬럼 기반 데이터베이스는 열을 중심으로 데이터를 조직하는 유형이다. 제시문의 핵심인 문서 단위 저장과 내부 필드 질의를 대표하는 분류는 아니다.

그래프 기반 데이터베이스는 노드와 노드 사이의 관계를 중심으로 데이터를 표현한다. 중첩된 문서 구조를 저장한다는 설명과 구별된다.

따라서 제시된 특징과 대표 제품에 해당하는 유형은 ④ 문서 기반 데이터베이스이다.

NoteBase

문 7. 선택정렬과 삽입정렬을 수행하여 다음 숫자를 오름차순으로 정렬할 경우, 각 정렬의 첫 번째 단계 결과를 옳게 나열한 것은? (단, 숫자는 1차원 배열로 구성되어 있다.)

1차원 배열 : 4, 3, 9, 1, 2, 6

① 선택정렬: 1, 3, 9, 4, 2, 6, 삽입정렬: 1, 3, 9, 4, 2, 6

② 선택정렬: 1, 3, 9, 4, 2, 6, 삽입정렬: 3, 4, 9, 1, 2, 6

③ 선택정렬: 3, 4, 9, 1, 2, 6, 삽입정렬: 3, 4, 9, 1, 2, 6

④ 선택정렬: 1, 3, 9, 4, 2, 6, 삽입정렬: 4, 3, 6, 1, 2, 9

정답 2번

간단한 해설

선택정렬의 첫 단계는 전체 최솟값 **1을 첫 값 4와 교환**한다. 삽입정렬의 첫 단계는 두 번째 원소 **3을 앞의 4보다 왼쪽에 삽입**하므로, 두 정렬의 첫 단계 결과가 서로 다르다.

계산 과정

주어진 배열은 4, 3, 9, 1, 2, 6이다. 두 정렬 모두 오름차순이지만, 첫 단계에서 처리하는 범위와 이동 방식이 다르다.

선택정렬은 아직 정렬하지 않은 구간 전체에서 최솟값을 찾아 그 구간의 맨 앞 값과 교환한다. 첫 단계에서는 전체 배열의 최솟값 1을 찾아 첫 값 4와 교환하므로 **1, 3, 9, 4, 2, 6**이 된다. 나머지 값의 상대 위치는 이 한 번의 교환에서 바뀌지 않는다.

삽입정렬은 첫 원소 4를 이미 정렬된 구간으로 보고, 두 번째 원소 3을 그 구간의 알맞은 위치에 넣는다. 4가 3보다 크므로 4를 한 칸 오른쪽으로 밀고 3을 맨 앞에 삽입한다. 첫 단계 결과는 **3, 4, 9, 1, 2, 6**이다. 뒤쪽의 9, 1, 2, 6은 아직 삽입 대상으로 처리하지 않는다.

따라서 선택정렬 결과는 1, 3, 9, 4, 2, 6이고, 삽입정렬 결과는 3, 4, 9, 1, 2, 6이므로 두 결과를 올바르게 연결한 것은 ②이다.

NoteBase

문 8. 십진수 −7의 2의 보수(2's complement)로 옳은 것은?

① $(1001)_2$

② $(1000)_2$

③ $(0111)_2$

④ $(0110)_2$

정답 1번

간단한 해설

4비트에서 7은 $(0111)_2$이다. 비트를 반전하고 1을 더하면 $(1001)_2$이므로 정답은 ①이다.

계산 과정

선택지에 맞춰 4비트로 표현한다. 먼저 양수 7을 이진수로 쓰면 $(0111)_2$이다.

모든 비트를 반전한 뒤 1을 더한다. 0111의 0과 1을 서로 바꾸면 1000이고, 여기에 1을 더하면 $1000 + 0001 = 1001$이다.

부호를 포함한 값으로 검산한다. 4비트 2의 보수에서 최상위 비트의 가중치는 -8이므로, $(1001)_2$의 값은 $-8 + 1 = -7$이다.

따라서 정답은 ① $(1001)_2$이다.

NoteBase

문 9. 클라우드 컴퓨팅 서비스에 관한 설명이다. ( ) 안에 들어갈 용어로 옳은 것은?

(ㄱ): 소프트웨어를 개발할 때 필요한 플랫폼을 제공하는 서비스이다.
(ㄴ): 네트워킹 기능, 컴퓨터 및 데이터 스토리지 공간을 제공하는 서비스이다.
(ㄷ): 클라우드 환경에서 운영되는 애플리케이션 서비스이며, 구매한 소프트웨어를 PC에 설치하지 않아도 웹에서 사용할 수 있는 서비스이다.

① ㄱ: PaaS, ㄴ: IaaS, ㄷ: SaaS
② ㄱ: PaaS, ㄴ: SaaS, ㄷ: IaaS
③ ㄱ: IaaS, ㄴ: SaaS, ㄷ: PaaS
④ ㄱ: IaaS, ㄴ: PaaS, ㄷ: SaaS

정답 1번

간단한 해설

개발·배포 환경은 **PaaS**, 컴퓨터·저장 공간·네트워크 자원은 **IaaS**, 완성된 애플리케이션 이용은 **SaaS**에 해당한다.

빈칸별 해설

ㄱ은 PaaS(Platform as a Service)이다. 개발 언어·라이브러리·실행 환경 등 **애플리케이션을 개발하고 배포하는 데 필요한 플랫폼**을 제공한다. 사용자는 그 환경 위에 자신이 만든 애플리케이션을 올려 운용한다.

ㄴ은 IaaS(Infrastructure as a Service)이다. 가상 컴퓨터의 처리 능력, 저장 공간, 네트워크 등 **기본 컴퓨팅 기반 자원**을 제공한다. 사용자는 이 자원 위에 운영체제와 필요한 소프트웨어를 설치하여 사용할 수 있다.

ㄷ은 SaaS(Software as a Service)이다. 공급자가 클라우드에서 운영하는 **완성된 애플리케이션을 서비스로 이용**한다. 제시된 웹 기반 이용 방식처럼 사용자는 자신의 PC에 해당 소프트웨어를 직접 설치하지 않고 기능을 사용할 수 있다.

따라서 ㄱ－PaaS, ㄴ－IaaS, ㄷ－SaaS의 대응이므로 ①과 일치한다.

NoteBase

문 10. 라운드 로빈(round robin) 스케줄링 방식에 관한 설명으로 옳지 않은 것만을 <보기>에서 있는 대로 고른 것은?

<보기>
ㄱ. 비선점 스케줄링 기법이다.
ㄴ. 시분할 시스템에 효과적인 방식이다.
ㄷ. 문맥 교환이 발생할 수 있는 방식이다.
ㄹ. 프로세스들이 돌아가면서 정해진 시간만큼 CPU를 할당받아 실행되는 방식이다.

① ㄱ
② ㄴ
③ ㄱ, ㄹ
④ ㄷ, ㄹ

정답 1번

간단한 해설

라운드 로빈은 시간 조각이 끝나면 실행 중인 프로세스의 CPU를 회수하는 **선점형 스케줄링**이다. 따라서 비선점 방식이라는 ㄱ의 설명이 잘못되었다.

보기별 해설

ㄱ. X: 라운드 로빈은 각 프로세스에 할당한 시간 조각이 끝나면 운영체제가 CPU를 회수하여 다른 준비 프로세스에 넘기는 **선점형 스케줄링**이다. 비선점 방식이라는 설명은 반대이다.

ㄴ. O: 여러 프로세스가 짧은 시간씩 CPU를 번갈아 사용하므로 한 프로세스가 장시간 CPU를 독점하는 것을 막는다. 여러 사용자에게 실행 기회를 나누고 응답성을 확보하려는 **시분할 시스템**에 적합하다.

ㄷ. O: 실행 중인 프로세스에서 다른 프로세스로 CPU 사용권을 넘길 때 **기존 실행 상태를 저장하고 다음 상태를 복원**하는 문맥 교환이 발생할 수 있다.

ㄹ. O: 준비 큐의 프로세스들이 순환하며 미리 정한 시간 조각만큼 CPU를 배정받는다. 다만 이 시간은 **한 번에 사용할 수 있는 최대 시간**으로, 프로세스가 먼저 종료하거나 입출력을 기다리면 그 전에 CPU를 내놓을 수 있다.

옳지 않은 보기는 ㄱ만이므로 ①에 해당한다.

NoteBase

문 11. 애자일(agile) 프로세스의 장점으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보기>
ㄱ. 변화 중심이 아닌, 계획 중심의 대응을 중시한다.
ㄴ. 절차와 도구 중심이 아닌, 상호 의사소통을 중시한다.
ㄷ. 계약과 협상 중심이 아닌, 고객과의 협력을 중시한다.
ㄹ. 문서 중심이 아닌, 실행 가능한 소프트웨어를 중시한다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄴ, ㄷ, ㄹ

정답 4번

간단한 해설

애자일은 계획을 고수하기보다 **변화에 유연하게 대응**하며, 사람 간 상호작용·고객과의 협력·작동하는 소프트웨어에 더 높은 가치를 둔다.

보기별 해설

ㄱ. X: 애자일은 최초 계획을 고수하기보다 **변화한 요구에 대응하는 것**을 중시한다. 따라서 변화보다 계획 중심의 대응을 우선한다는 설명은 반대이다.

ㄴ. O: 정해진 절차나 도구 자체보다 **개발 참여자 간 상호작용과 의사소통**을 중시한다. 이를 통해 요구와 문제를 공유하고 조율한다.

ㄷ. O: 계약 조건을 협상하는 데만 집중하기보다 **고객과 지속적으로 협력하는 것**을 중시한다. 고객의 피드백을 반영해 필요한 소프트웨어를 만들어 간다.

ㄹ. O: 문서 작성 자체보다 **실제로 작동하는 소프트웨어**를 중시한다. 계획·문서·절차·계약의 가치를 부정하는 것이 아니라, 변화 대응·상호작용·고객 협력·작동하는 결과물에 더 높은 가치를 두는 것이다.

옳은 보기는 ㄴ, ㄷ, ㄹ이므로 이를 모두 포함하는 ④와 일치한다.

NoteBase

문 12. 데이터베이스 스키마에 관한 설명이다. ( ) 안에 들어갈 용어로 옳은 것은?

(ㄱ) : 데이터베이스의 물리적 구조를 정의한 것이다.
(ㄴ) : 사용자나 응용 프로그래머가 각 개인의 입장에서 필요로 하는 데이터베이스의 논리적 구조를 정의한 것이다.
(ㄷ) : 개체 간의 관계와 제약 조건을 나타내고, 데이터베이스의 접근 권한, 보안정책 및 무결성 규정에 관한 명세를 정의한 것이다.

① ㄱ: 외부스키마, ㄴ: 내부스키마, ㄷ: 개념스키마
② ㄱ: 외부스키마, ㄴ: 개념스키마, ㄷ: 내부스키마
③ ㄱ: 내부스키마, ㄴ: 외부스키마, ㄷ: 개념스키마
④ ㄱ: 내부스키마, ㄴ: 개념스키마, ㄷ: 외부스키마

정답 3번

간단한 해설

데이터베이스의 **물리적 저장 구조는 내부스키마**, 사용자별 논리적 관점은 외부스키마, 전체 데이터의 논리적 구조와 제약 조건은 개념스키마가 정의한다.

빈칸별 해설

ㄱ: 내부스키마이다. 데이터를 저장장치에 어떤 형태로 배치하고 접근할지를 정의하는 **물리적 수준의 스키마**이다. 저장 레코드 형식, 저장 순서, 인덱스 등 실제 저장 방식과 관련된다.

ㄴ: 외부스키마이다. 전체 데이터베이스 가운데 특정 사용자나 응용 프로그램이 필요로 하는 부분을 **사용자별 논리적 관점**으로 정의한다. 사용자마다 필요한 데이터와 표현이 다르므로 여러 외부스키마가 존재할 수 있다.

ㄷ: 개념스키마이다. 데이터베이스 전체에 대한 **통합된 논리적 구조**를 정의한다. 개체와 속성, 개체 간 관계, 제약조건, 접근 권한과 무결성 규정 등이 여기에 포함되며, 구체적인 물리적 저장 방식과는 구분된다.

따라서 ㄱ은 내부스키마, ㄴ은 외부스키마, ㄷ은 개념스키마이며, 이 대응을 나타낸 ③과 일치한다.

NoteBase

문 13. HTML5의 장점으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보기>
ㄱ. <svg> 태그로 2차원 벡터 그래픽을 구현할 수 있다.
ㄴ. <canvas> 태그로 동적 그래픽과 애니메이션을 구현할 수 있다.
ㄷ. 시멘틱 요소를 도입하여 웹 페이지의 내용을 구성할 수 있다.
ㄹ. 다양한 멀티미디어 콘텐츠를 액티브X(ActiveX)의 제약 없이 웹 브라우저에서 실행되도록 만든 마크업 언어이다.

① ㄱ, ㄴ
② ㄴ, ㄷ
③ ㄱ, ㄴ, ㄹ
④ ㄱ, ㄴ, ㄷ, ㄹ

정답 4번

간단한 해설

HTML5는 **벡터 그래픽·스크립트 기반 그래픽·의미 있는 문서 구조·플러그인 없는 멀티미디어**를 지원한다. 이 기능들은 별도 플러그인 의존도를 낮추고 웹 페이지의 표현력과 구조를 개선한다.

보기별 해설

ㄱ. O: 'svg' 요소는 원·선·경로 등을 좌표와 도형 정보로 표현하는 **2차원 벡터 그래픽**을 문서에 포함한다. 벡터 방식이므로 확대하더라도 도형의 윤곽을 선명하게 표현할 수 있다.

ㄴ. O: 'canvas' 요소는 스크립트로 그림을 그릴 수 있는 영역을 제공한다. 자바스크립트로 화면을 지우고 다시 그리는 작업을 반복하면 **동적 그래픽과 애니메이션**을 구현할 수 있다.

ㄷ. O: 'header', 'nav', 'article', 'section', 'footer' 등의 **시멘틱 요소**는 머리말·탐색·독립적인 글·구획·꼬리말처럼 문서 영역의 의미를 나타낸다. 이를 이용해 페이지 내용을 의미에 맞게 구성할 수 있다.

ㄹ. O: 'audio'와 'video' 요소 등을 통해 브라우저가 멀티미디어를 직접 재생하도록 지원하므로 **액티브X 같은 별도 플러그인에 의존하지 않고** 콘텐츠를 제공할 수 있다. 다만 실제 재생 가능한 코덱과 형식은 브라우저 지원 범위에 따른다.

따라서 ㄱ·ㄴ·ㄷ·ㄹ이 모두 옳으므로 해당 조합은 ④이다.

NoteBase

문 14. 스택에서 A, B, C, D로 순서가 정해진 입력 자료를 Push → Push → Pop → Push → Push → Pop → Pop → Pop으로 연산했을 때 출력으로 옳은 것은?

① A, B, C, D

② A, C, B, D

③ B, A, D, C

④ B, D, C, A

정답 4번

간단한 해설

스택은 **마지막에 넣은 자료를 먼저 꺼내는 후입선출(LIFO)** 구조이다. 각 Pop 시점의 맨 위 자료를 차례로 꺼내면 B, D, C, A가 출력된다.

계산 과정

Push는 자료를 스택의 맨 위에 넣고, Pop은 맨 위 자료를 꺼내 출력하는 연산이다. 다음 대괄호는 왼쪽이 바닥, 오른쪽이 맨 위인 스택 상태를 나타낸다.

1. Push A: 빈 스택에 A를 넣어 [A]가 된다.

2. Push B: A 위에 B를 넣어 [A, B]가 된다.

3. Pop: 맨 위의 B를 출력하고 [A]가 남는다.

4. Push C: 남아 있는 A 위에 C를 넣어 [A, C]가 된다.

5. Push D: C 위에 D를 넣어 [A, C, D]가 된다.

6. Pop: 맨 위의 D를 출력하고 [A, C]가 남는다.

7. Pop: 맨 위의 C를 출력하고 [A]가 남는다.

8. Pop: 마지막 A를 출력하고 스택이 빈 상태가 된다.

따라서 출력 순서는 **B, D, C, A**이며, ④와 일치한다.

NoteBase

문 15. C 클래스의 네트워크 한 개를 8개의 서브넷으로 나누고자 할 때 필요한 서브넷마스크 값으로 옳은 것은?

① 255.255.255.128

② 255.255.255.192

③ 255.255.255.224

④ 255.255.255.248

정답 3번

간단한 해설

서브넷 8개를 구분하려면 $2^3 = 8$에 따라 **호스트 비트 3개를 빌린다**. 기본 /24에 3비트를 더한 /27의 마스크는 255.255.255.224이다.

계산 과정

C 클래스 네트워크의 기본 서브넷 마스크는 255.255.255.0으로, 네트워크 부분은 24비트이고 호스트 부분은 8비트이다. 서브넷을 나누려면 호스트 부분의 앞쪽 비트를 서브넷 식별에 사용한다.

서브넷 비트를 n개 빌리면 서브넷 수는 2^n개이다. $2^3 = 8$이므로 **호스트 부분에서 3비트를 빌려야 한다**. 따라서 마스크에서 1인 비트 수는 24 + 3 = 27비트가 된다.

마지막 옥텟은 11100000이므로 십진수로 128 + 64 + 32 = 224이다. 앞의 세 옥텟은 그대로 255이므로 필요한 마스크는 **255.255.255.224(/27)**이며, ③과 일치한다.

NoteBase

문 16. CISC에 관한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보기>
ㄱ. 명령어의 형태와 크기가 다양한 '가변 길이 명령어'를 활용한다.
ㄴ. 수행 속도를 높이기 위하여 처리할 수 있는 명령어의 수를 줄인 것을 의미한다.
ㄷ. 중앙처리장치의 주요 설계방식 중 하나이며, 단순하고 적은 명령어와 고정 길이 명령어를 가진다.
ㄹ. 복잡한 명령어를 가지고 있어 명령어 하나를 실행하는 데 여러 클록(clock) 주기를 필요로 한다.

① ㄱ, ㄹ
② ㄴ, ㄷ
③ ㄷ, ㄹ
④ ㄱ, ㄴ, ㄷ

정답 1번

간단한 해설

CISC는 다양한 기능의 **복잡한 가변 길이 명령어**를 제공하며, 복잡한 명령을 여러 클록 주기에 걸쳐 실행하는 전통적 특성이 있다. 단순한 명령어와 고정 길이 형식을 중심으로 한 설명은 RISC에 해당한다.

보기별 해설

ㄱ. O: CISC는 복잡하고 다양한 기능의 명령어를 제공하는 설계 방식이다. 전형적으로 명령어에 필요한 연산 코드·피연산자 등의 구성에 따라 크기가 달라지는 **가변 길이 명령어**를 사용한다.

ㄴ. X: 명령어 집합을 단순화하여 빠른 처리가 가능하도록 하는 방식은 **RISC**에 대한 설명이다. CISC는 다양한 기능을 수행하는 복잡한 명령어를 제공한다.

ㄷ. X: 단순한 명령어와 규칙적인 고정 길이 형식을 중심으로 설계하는 것은 전통적인 **RISC의 특징**이다. CISC는 명령어 종류와 형식이 다양하여 해독·제어가 상대적으로 복잡하다.

ㄹ. O: 전통적인 CISC에서는 복잡한 명령어가 여러 세부 동작을 수행하므로 **하나의 명령 실행에 여러 클록 주기**가 필요할 수 있다. 이는 전형적인 설계 특성에 관한 설명이며 모든 구현에서 모든 명령의 실행 시간이 같다는 뜻은 아니다.

옳은 보기는 ㄱ과 ㄹ이므로 ①과 일치한다.

NoteBase

문 17. 다음에서 설명하는 디지털 포렌식의 기본 원칙으로 옳은 것은?

증거를 획득한 뒤에는 이송, 분석, 보관, 법정 제출이라는 일련의 과정이 명확해야 하며 이러한 과정을 추적할 수 있어야 한다.

① 재현의 원칙
② 정당성의 원칙
③ 무결성의 원칙
④ 연계 보관성의 원칙

정답 4번

간단한 해설

연계 보관성의 원칙은 증거의 수집부터 법정 제출까지 **취급·이송·보관 이력을 끊김 없이 기록하여 추적할 수 있어야 한다**는 원칙이다. 제시문은 증거 내용의 변화 여부보다 증거가 거쳐 온 과정의 명확성을 강조한다.

개념 구분

연계 보관성의 원칙은 증거를 누가, 언제, 어디에서 취급하고 누구에게 인계했는지 기록하여 증거의 관리 과정을 추적할 수 있도록 하는 원칙이다. 제시문의 '이송, 분석, 보관, 법정 제출'이라는 일련의 과정과 '추적할 수 있어야 한다'는 표현이 핵심 단서이다.

무결성의 원칙은 수집한 디지털 증거가 이후 처리 과정에서도 **임의로 변경되거나 훼손되지 않았음**을 보장하는 데 초점을 둔다. 연계 보관성이 증거의 취급 이력을 확인하는 것이라면, 무결성은 증거 내용이 동일하게 유지되었는지를 확인하는 것이다.

재현의 원칙은 같은 조건과 절차로 분석했을 때 같은 결과를 얻을 수 있어야 한다는 원칙이다.

정당성의 원칙은 증거의 수집과 처리 등이 적법한 권한과 절차에 따라 이루어져야 한다는 원칙이다.

따라서 증거가 거친 전 과정을 명확히 하고 추적 가능하게 해야 한다는 설명은 **연계 보관성의 원칙**에 해당하며, ④와 일치한다.

NoteBase

문 18. 다음에서 설명하는 머신러닝에서 사용하는 학습 용어로 옳은 것은?

주어진 현재 상황에서 보상을 최대화하도록 다음 행동을 학습시키는 방법이다. 학습 주체인 대리인에게 지시한 명령에 대해 대리인을 관찰하여 다음 행동을 하면 잘못한 행동에는 벌을, 잘한 행동에는 보상을 줌으로써 대리인은 가장 큰 보상을 얻기 위한 정책을 통해 스스로 학습하는 방법이다.

① 지도학습

② 강화학습

③ 반지도학습

④ 비지도학습

정답 2번

간단한 해설

강화학습은 대리인이 환경과 상호작용하면서 **보상에 따라 행동 정책을 개선**하는 방법이다. 정답이 붙은 예제를 그대로 맞히는 것이 아니라, 행동의 결과로 얻는 누적 보상을 크게 만드는 방향으로 학습한다.

개념 구분

지문의 핵심은 대리인이 현재 상태에서 행동을 선택하고, 그 결과로 받은 보상이나 벌을 이용해 다음 행동을 결정하는 **정책**을 학습한다는 점이다. 강화학습의 정책은 상태가 주어졌을 때 어떤 행동을 할지 정하는 규칙이며, 당장의 보상뿐 아니라 장기적으로 얻을 보상도 고려한다.

지도학습은 입력과 정답이 짝지어진 데이터로 예측 관계를 학습한다.

반지도학습은 정답이 있는 데이터와 정답이 없는 데이터를 함께 사용한다.

비지도학습은 정답 표지 없이 데이터의 군집이나 잠재 구조를 찾는다. 이들은 지문처럼 **행동의 결과로 받은 보상으로 정책을 학습**한다는 설명과 구별된다.

따라서 지문이 설명하는 학습 방법은 ② 강화학습이다.

NoteBase

문 19. 이더넷(ethernet)에 관한 설명으로 옳지 않은 것은?

① IEEE 802.3 국제표준이다.

② 매체 접근 제어 방식은 CSMA/CD이다.

③ 호스트에서 전송할 데이터가 많아질수록 충돌 발생 빈도가 증가한다.

④ 호스트가 채널의 상태를 감지해 충돌을 피하는 토큰 패싱(token passing) 매체 접근 제어 방식을 사용한다.

정답 4번

간단한 해설

이더넷의 전통적인 공유매체 접근 방식은 **토큰 패싱이 아니라 CSMA/CD**이다. 토큰 패싱은 전송 권한을 나타내는 토큰을 가진 호스트만 전송하는 방식이다.

선지별 해설

① O: 이더넷은 **IEEE 802.3**으로 표준화된 유선 LAN 기술이다. 표준은 프레임과 매체 접근 제어, 물리 계층 등을 규정한다.

② O: 공유매체를 사용하는 반이중 이더넷은 **CSMA/CD**로 매체 접근을 제어한다. 전송 전에 매체가 사용 중인지 감지하고, 전송 중 충돌이 검출되면 전송을 중단한 뒤 대기 후 재전송한다. 전이중 스위치 이더넷에서는 이러한 충돌 제어가 필요하지 않다.

③ O: 같은 충돌 도메인의 여러 호스트가 공유매체를 두고 경쟁하는 경우, 전송하려는 데이터와 전송 시도가 많아지면 **동시에 전송을 시작할 가능성**이 커져 충돌 발생 빈도가 증가한다. 충돌이 없는 전이중 링크의 특성을 설명한 것은 아니다.

④ X: 이더넷은 **토큰 패싱을 사용하지 않는다**. CSMA/CD는 매체 상태를 감지하고 발생한 충돌을 검출하는 경쟁 방식인 반면, 토큰 패싱은 토큰을 순서대로 전달하여 토큰을 가진 호스트에게 전송 권한을 주는 방식이다.

NoteBase

문 21. 운영체제의 기능 중 자원관리로 옳지 않은 것은?

① 메모리 관리

② 프로세스 관리

③ 사용자 권한 관리

④ 파일(데이터) 관리

정답 3번

간단한 해설

사용자 권한 관리는 누가 어떤 자원에 접근할 수 있는지를 제한하는 **보호·보안 기능**이다.
메모리·프로세스·파일을 배분하고 운용하는 자원관리와 구분한다.

선지별 해설

① O: 메모리 관리는 실행 중인 프로그램에 필요한 메모리 공간을 **할당하고 회수**하며, 사용 중인 영역과 사용 가능한 영역을 관리한다. 한정된 메모리를 여러 프로세스가 효율적으로 사용하도록 하는 자원관리이다.

② O: 프로세스 관리는 프로세스를 생성·종료하고 실행 상태를 관리하며, 스케줄링을 통해 **CPU 사용 기회**를 배분한다. 프로세스 간 동기화와 통신을 지원하는 일도 포함한다.

③ X: 사용자 권한 관리는 사용자별 접근 허용 범위와 수행 가능한 작업을 정하는 **보호·보안 기능**으로 분류한다. 운영체제가 수행하지 않는다는 뜻이 아니라, 이 문항의 기능 분류에서 자원의 할당·운용을 다루는 자원관리와 구분된다는 뜻이다.

④ O: 파일 관리는 파일과 디렉터리의 생성·삭제, 저장 위치와 공간, 읽기·쓰기 등의 작업을 관리한다.
보조기억장치에 저장된 데이터를 체계적으로 이용하게 하는 자원관리이다.

NoteBase

문 22. 객체지향 기법에 관한 설명이다. ( ) 안에 들어갈 용어로 옳은 것은?

(ㄱ): 상위 클래스의 속성을 하위 클래스가 물려받는 것이다.
(ㄴ): 정보처리에 필요한 변수, 함수 등을 묶어 필요한 인터페이스만 밖으로 드러내는 과정이다.
(ㄷ): 객체의 세부 사항을 없애고 중요한 부분만 추출하여 간결하고 이해하기 쉽게 하는 과정이다.

① ㄱ: 상속, ㄴ: 캡슐화, ㄷ: 추상화
② ㄱ: 캡슐화, ㄴ: 상속, ㄷ: 추상화
③ ㄱ: 추상화, ㄴ: 상속, ㄷ: 캡슐화
④ ㄱ: 상속, ㄴ: 추상화, ㄷ: 캡슐화

정답 1번

간단한 해설

상위 클래스의 특성을 물려받는 것은 **상속**, 변수와 함수를 묶고 필요한 인터페이스만 드러내는 것은 **캡슐화**이다. 불필요한 세부 사항을 생략하고 핵심 특성만 표현하는 것은 **추상화**이다.

빈칸별 해설

ㄱ. 상속: 상위 클래스에 정의된 속성과 기능을 하위 클래스가 물려받아 사용하는 것이다. 공통 부분을 재사용하고, 하위 클래스에서 필요한 특성을 추가하거나 기능을 재정의할 수 있다. 따라서 '상위 클래스의 속성을 하위 클래스가 물려받는다'는 설명에 해당한다.

ㄴ. 캡슐화: 관련된 데이터와 이를 처리하는 함수를 하나의 단위로 묶고, 외부에는 필요한 인터페이스를 제공하는 것이다. **내부 구현을 감추고 정해진 통로로 접근하게 한다**는 점이 핵심이다. 변수와 함수의 결합 및 외부 공개 범위의 통제를 설명하므로 추상화와 구별된다.

ㄷ. 추상화: 대상의 복잡한 세부 사항은 생략하고 목적에 필요한 중요한 특성만 추려 표현하는 것이다. **무엇이 핵심 특성인지 선택하여 단순화**한다는 점에서, 내부 구현을 묶고 감추는 캡슐화와 구별된다.

따라서 ㄱ은 상속, ㄴ은 캡슐화, ㄷ은 추상화이며, 이 대응을 제시한 ①과 일치한다.

NoteBase

문 23. 판매 데이터베이스의 제품과 주문 테이블에서 홍길동 고객이 2회 이상 주문한 제품의 제품명과 제조업체를 검색하는 SQL 명령을 작성할 때, ( ) 안에 들어갈 내용으로 옳은 것은?

<판매 데이터베이스의 테이블 스키마>
제품(제품번호, 제품명, 재고량, 단가, 제조업체)
주문(주문번호, 주문고객, 주문제품, 수량, 배송지, 주문일자)

<SQL 명령>

```
SELECT 제품명, 제조업체
FROM 제품
WHERE 제품번호 ( ㄱ ) ( SELECT 주문제품
                         FROM 주문
                         WHERE 주문고객='홍길동'
                         GROUP BY ( ㄴ )
                         HAVING COUNT(*) >= 2 );
```

① ㄱ: IN, ㄴ: 주문제품
② ㄱ: IN, ㄴ: 주문고객
③ ㄱ: =, ㄴ: 주문고객
④ ㄱ: =, ㄴ: 주문제품

정답 1번

간단한 해설

홍길동의 주문을 **제품별로 묶어 주문 횟수가 2회 이상인 제품**을 찾는다. 결과가 여러 제품일 수 있으므로 바깥 질의에서는 **IN**으로 포함 여부를 검사한다.

NoteBase

빈칸별 해설

ㄴ에는 **주문제품**이 들어간다. 내부 질의의 WHERE절로 홍길동의 주문만 남긴 다음, GROUP BY 주문제품으로 같은 제품의 주문들을 한 그룹으로 묶는다. HAVING COUNT(*) >= 2는 각 제품의 주문 행 수가 2개 이상인 그룹만 남기므로, 수량 합계가 아니라 해당 제품을 주문한 횟수를 검사한다. 주문고객으로 묶으면 이미 홍길동만 남은 행 전체가 한 그룹이 되어 제품별 주문 횟수를 구할 수 없다.

ㄱ에는 **IN**이 들어간다. 내부 질의는 조건을 만족하는 주문제품 번호를 여러 개 반환할 수 있다. 바깥 질의는 제품번호가 이 결과 집합에 포함되는 제품을 찾아 제품명과 제조업체를 출력해야 하므로 IN을 사용한다. 단순 등호는 단일 값과 비교하는 경우에 쓰므로 여러 제품이 나올 수 있는 내부 질의에 적합하지 않다.

따라서 ㄱ은 IN, ㄴ은 주문제품이며, 이 조합은 ①이다.

NoteBase

문 24. C++ 프로그램 결과의 출력값으로 옳은 것은?

```
#include <iostream>

using spacename std;

int func(int& p) {
  p = 20;
  return 50;
}
int main(void) {
  int num = 0;
  int& x = num;
  func(x);
  cout << x << endl;
  return 0;
}
```

① 15
② 20
③ 30
④ 50

정답 2번

출제 포인트

C++의 참조가 원래 변수에 붙인 별명이라는 점과 함수의 반환값이 참조를 통해 변경한 변수의 값과 다르다는 점을 묻는다.

핵심 개념

참조는 기존 변수와 같은 객체를 가리키는 별명이다. int& x = num;에서 x는 num의 별명이고, int& p로 선언된 매개변수도 전달된 변수를 참조한다. 따라서 p에 대입하면 원래 변수의 값도 바뀐다. 함수의 반환값은 return으로 호출한 곳에 돌려주는 값이며, 참조 변수에 자동으로 대입되는 값이 아니다.
제시문의 using spacename std;는 그대로는 컴파일되지 않는다. 이 해설은 spacename을 namespace의 명백한 인쇄 오기로 해석하여, using namespace std;를 의도한 코드의 참조 동작을 설명한다.

NoteBase

실행 과정

1. int num = 0;으로 num의 초기값은 0이다.
2. int& x = num;으로 x는 num을 참조한다.
3. func(x)를 호출하면 참조 매개변수 p도 x가 가리키는 num을 참조한다. p = 20;을 실행하면 num과 x의 값은 모두 20이 된다.
4. return 50;은 50을 반환하지만, 호출문 func(x);는 반환값을 변수에 대입하거나 출력하지 않는다. 따라서 x는 20을 유지한다.
5. cout << x << endl;은 x의 값인 20을 출력하고 줄을 바꾼다.

선지별 해설

① X — 15: 코드에는 참조 대상의 값을 15로 만드는 대입이나 연산이 없다. p = 20;이 num과 x의 값을 바꾸므로 출력값은 15가 아니라 20이다.
② O — 20: p가 num을 참조하므로 p = 20;을 실행하면 x로 읽는 값도 20이 된다. 출력문은 x를 출력한다.
③ X — 30: 코드에는 출력값을 30으로 만드는 연산이 없다. 참조 대상에는 20이 대입되므로 출력값은 30이 아니라 20이다.
④ X — 50: 50은 func의 반환값이다. func(x);에서는 이 반환값을 사용하지 않고, 출력문도 함수의 반환값이 아닌 x를 출력하므로 출력값은 50이 아니라 20이다.

오답 포인트

④의 50은 함수의 반환값과 참조를 통해 바뀐 변수의 값을 혼동하게 하는 선지이다. func(x);는 x = func(x);가 아니므로 반환값 50이 x에 대입되지 않는다. x와 p를 별개의 값 복사본으로 보아도 참조에 의한 변경을 놓치게 된다.

다시 나오면 이렇게 판단

참조 선언의 &를 확인하여 각 이름이 어떤 원래 변수를 가리키는지 연결한다. 함수 내부의 대입이 그 변수에 미치는 영향을 추적한 뒤, 반환값을 사용하는 호출인지 확인하고 마지막 출력문이 실제로 출력하는 대상을 판단한다.

암기 포인트

참조에 대입하면 원래 변수가 바뀐다. 반환값은 호출한 곳에서 사용해야 대입이나 출력에 반영된다.

NoteBase

문 25. 연산식의 후위 표기법에 관한 설명으로 옳지 않은 것은?

① 괄호가 필요 없다.

② 연산을 위해 스택을 이용한다.

③ 연산자의 우선순위를 고려할 필요가 없다.

④ 산술식을 표현할 때 연산자 뒤에 피연산자를 붙여 표기한다.

정답 4번

간단한 해설

후위 표기법은 **피연산자 뒤에 연산자**를 놓는 방식이다. 연산자를 먼저 쓰고 뒤에 피연산자를 놓는 것은 전위 표기법이다.

선지별 해설

① O: 후위 표기법은 연산자의 위치 자체가 계산 순서를 나타내므로 **계산 순서를 지정하는 괄호가 필요 없다**. 예를 들어 중위식 (A + B) × C는 후위식 'A B + C ×'로 나타낸다.

② O: 후위식을 왼쪽부터 읽으면서 피연산자는 스택에 넣고, 연산자를 만나면 필요한 피연산자를 스택에서 꺼내 계산한 뒤 결과를 다시 넣는다. 나중에 넣은 피연산자를 먼저 꺼내는 **후입선출 구조**가 계산에 적합하다.

③ O: 이미 후위식으로 표현된 식을 계산할 때는 곱셈 우선과 같은 **연산자 우선순위를 별도로 적용할 필요가 없다**. 다만 중위식을 후위식으로 변환할 때는 원래 식의 우선순위와 괄호를 반영해야 한다.

④ X: 후위 표기법은 **피연산자 뒤에 연산자**를 놓는다. 중위식 A + B의 후위식은 'A B +'이다. 연산자를 피연산자 앞에 놓는 '+ A B'는 전위 표기법이다.

NoteBase

문 26. OSI 7계층 중 다음에서 설명하는 계층으로 옳은 것은?

· 제한된 네트워크의 전송 효율을 높이기 위해 데이터를 압축한다.
· 송수신 데이터를 보호하기 위해 암호화 알고리즘으로 데이터를 변조한다.

① 전송 계층
② 표현 계층
③ 네트워크 계층
④ 데이터 링크 계층

정답 2번

간단한 해설

표현 계층은 응용 계층이 주고받는 데이터의 표현 형식을 다루며, 압축과 암호화·복호화 기능을 수행한다.

개념 구분

표현 계층은 OSI 7계층의 제6계층으로, 서로 다른 시스템에서도 데이터를 해석할 수 있도록 표현 형식을 변환한다. 제시된 **데이터 압축과 암호화**는 이 계층의 대표 기능이다.

압축은 데이터의 양을 줄여 전송 자원을 효율적으로 사용하게 한다. 암호화는 데이터를 알아보기 어려운 형태로 변환하여 기밀성을 보호하며, 수신 측에서는 복호화하여 원래 내용을 얻는다.

전송 계층은 종단 간 데이터 전달,

네트워크 계층은 논리 주소와 경로 선택,

데이터 링크 계층은 인접 노드 사이의 프레임 전달과 오류 검출 등을 담당한다.

따라서 제시된 기능에 해당하는 것은 ② **표현 계층**이다.

NoteBase

문 27. 전송 계층의 TCP(Transmission Control Protocol)와 UDP(User Datagram Protocol)에 관한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보기>

ㄱ. TCP는 패킷 손실로 서비스 지연이 발생할 수 있다.

ㄴ. TCP는 실시간 인터넷 영화 감상에 적합한 프로토콜이다.

ㄷ. TCP는 신뢰성 있는 데이터 전송을 보장하는 연결형 프로토콜이다.

ㄹ. UDP는 송신측에서 데이터를 패킷 단위로 분해하고 수신측에서 조립한다.

① ㄱ, ㄴ

② ㄱ, ㄷ

③ ㄱ, ㄷ, ㄹ

④ ㄴ, ㄷ, ㄹ

정답 2번

간단한 해설

TCP는 **연결을 설정하고 순서 제어와 재전송으로 신뢰성을 확보**하므로, 손실된 데이터를 기다리는 동안 지연이 생길 수 있다. UDP는 자체적으로 데이터 스트림의 순서 복원이나 재전송을 보장하지 않는다.

NoteBase

보기별 해설

ㄱ. O: TCP는 손실된 데이터를 재전송하고 수신 데이터를 순서에 맞게 전달한다. 앞부분의 데이터가 손실되면 이를 복구할 때까지 뒤에 도착한 데이터의 전달이 지연될 수 있으므로 서비스 지연이 발생할 수 있다.

ㄴ. X: 이 문항은 정확한 재전송보다 **지연을 줄이는 실시간성**에 초점을 두어 UDP와 비교한다. TCP는 손실 복구와 순서 보장을 위해 기다리는 시간이 발생하므로, 실시간 전송의 대표적인 프로토콜은 자체 재전송을 하지 않는 UDP이다. 다만 TCP로 동영상을 전송할 수 없다는 뜻은 아니며, 버퍼링을 사용하는 인터넷 동영상 서비스에는 TCP도 사용된다.

ㄷ. O: TCP는 통신 전에 연결을 설정하는 연결형 프로토콜이다. 순서 번호, 확인 응답, 오류 검출 및 재전송 등을 이용하여 **신뢰성 있는 순서대로의 바이트 스트림 전송**을 제공한다.

ㄹ. X: UDP는 응용 프로그램이 보낸 메시지를 데이터그램 단위로 전송하며, 바이트 스트림을 나누어 보내고 수신 측에서 순서대로 복원하는 기능을 제공하지 않는다. 이러한 스트림 분할과 순서 복원은 TCP의 특징이다. UDP 데이터그램이 IP 계층에서 단편화·재조립될 수는 있지만, 이를 UDP 자체의 스트림 조립 기능과 혼동해서는 안 된다.

따라서 옳은 보기는 ㄱ, ㄷ이며, ②와 일치한다.

NoteBase

문 28. 다음에서 설명하는 용어로 옳은 것은?

현실 세계의 물리적 시스템과 기능 및 동작을 그대로 가상공간에 만들어 컴퓨터상에 디지털 데이터 모델로 표현하여 똑같이 복제하고 실시간으로 서로 반응하도록 한 것이다.

① 가상현실
② 증강현실
③ 디지털 트윈
④ 유비쿼터스 컴퓨팅

정답 3번

간단한 해설

현실 시스템을 디지털 모델로 복제하고 **실제 대상과 데이터를 주고받으며 상태를 반영**하는 것은 디지털 트윈이다. 단순한 가상공간 표현보다 현실과 모델의 연결이 핵심이다.

핵심 풀이

디지털 트윈은 **현실의 물리적 대상과 대응하는 디지털 모델**을 만들고 실제 대상의 데이터를 연결하여 상태와 동작을 반영하는 기술이다. 단순히 모양을 그리는 데 그치지 않고 실제 시스템과 연계된 모델로 분석이나 시뮬레이션을 수행한다.

지문의 '물리적 시스템과 기능 및 동작을 가상공간에 복제'한다는 부분은 현실 대상에 대응하는 모델을 뜻하고, '실시간으로 서로 반응'한다는 부분은 **현실과 디지털 모델의 데이터 연계**를 뜻한다. 이 두 단서가 디지털 트윈을 가리킨다.

가상현실은 컴퓨터로 만든 환경에 몰입하게 하는 기술이고, 증강현실은 현실 장면 위에 디지털 정보를 겹쳐 보여주는 기술이다. 유비쿼터스 컴퓨팅은 여러 사물과 환경에 컴퓨팅 기능이 스며들어 언제 어디서나 서비스를 이용하는 개념이다. 이들은 현실 시스템의 대응 모델을 데이터로 연결한다는 디지털 트윈의 핵심과 구별되므로 정답은 ③이다.

NoteBase

문 29. 빅데이터 분석 파이프라인의 단계를 순서대로 옳게 나열한 것은?

ㄱ. 데이터 탐색
ㄴ. 데이터 보기
ㄷ. 데이터 준비
ㄹ. 데이터 시각화
ㅁ. 데이터 클린징

① ㄱ — ㄷ — ㄴ — ㄹ — ㅁ
② ㄱ — ㄷ — ㄴ — ㅁ — ㄹ
③ ㄷ — ㄴ — ㅁ — ㄱ — ㄹ
④ ㄷ — ㄴ — ㄱ — ㄹ — ㅁ

정답 3번

간단한 해설

데이터를 준비해 상태를 확인한 뒤 **오류를 정제하고 패턴을 탐색**한다. 이후 탐색 결과를 시각적으로 정리해야 자료의 오류로 인한 해석 왜곡을 줄일 수 있다.

올바른 순서

ㄷ → ㄴ → ㅁ → ㄱ → ㄹ

ㄷ. 데이터 준비: 분석에 필요한 데이터를 확보하고 불러와 사용할 수 있는 형태로 준비한다. 먼저 분석 대상이 마련되어야 내용을 확인할 수 있다.

ㄴ. 데이터 보기: 데이터의 일부와 항목, 자료형, 기본적인 분포 등을 살펴 구조와 상태를 파악한다. 이 과정에서 결측치·중복·형식 오류처럼 **정제가 필요한 부분**을 확인한다.

ㅁ. 데이터 클린징: 확인한 결측치, 중복, 오류와 형식 불일치 등을 처리한다. 데이터의 품질을 개선한 뒤 분석해야 **오류에 따른 왜곡**을 줄일 수 있다.

ㄱ. 데이터 탐색: 정제된 자료의 분포, 변수 간 관계, 특징이나 패턴을 살펴 분석에 필요한 단서를 찾는다. 앞의 데이터 보기가 자료의 구조와 상태를 확인하는 단계라면, 여기서는 분석 목적에 맞게 내용을 더 깊이 조사한다.

ㄹ. 데이터 시각화: 탐색한 특성과 관계를 그래프·도표 등으로 표현하여 결과를 이해하기 쉽게 전달한다. 제시된 파이프라인에서는 탐색 결과를 시각적으로 정리하는 단계에 해당한다.

NoteBase

문 30. 신경망 학습에서 각 뉴런의 출력신호를 결정하기 위하여 입력신호와 가중치를 곱한 후 다음 층으로 값을 전달하기 전에 거치는 함수로 옳은 것은?

① 오차 함수

② 목적 함수

③ 손실 함수

④ 활성화 함수

정답 4번

간단한 해설

활성화 함수는 뉴런이 계산한 가중합을 입력받아 다음 층으로 보낼 출력값을 결정한다. 손실 함수는 뉴런의 출력 변환이 아니라 예측과 목표 사이의 차이를 평가한다.

개념 구분

뉴런은 입력값에 각각의 가중치를 곱해 합하고 편향을 더한 뒤, 그 결과에 **활성화 함수**를 적용한다. 입력을 x□, 가중치를 w□, 편향을 b라고 하면 출력은 $y = f(\sum_i w_i x_i + b)$로 나타낼 수 있으며, 여기서 f가 활성화 함수이다.

시그모이드, tanh, ReLU 등은 대표적인 활성화 함수이다. 비선형 활성화 함수는 층을 여러 개 쌓은 신경망이 단순한 선형 변환을 넘어 복잡한 관계를 표현하도록 한다.

오차 함수나 손실 함수는 예측 결과가 목표와 얼마나 다른지를 수치화하며, 목적 함수는 학습에서 최소화하거나 최대화할 기준을 나타낸다. 이들은 개별 뉴런에서 다음 층으로 전달할 값을 변환하는 기능과 구별되므로, 해당 함수는 ④ **활성화 함수**이다.

NoteBase

문 31. C 프로그램에서 배열 변수 a와 포인터 변수 p가 다음과 같이 선언되었을 때, *(p++)의 결괏값으로 옳은 것은?

```
int a[] = { 1, 2, 3, 4, 5, 6 };
int *p = &a[2];
```

① 2

② 3

③ 4

④ 5

정답 2번

간단한 해설

후위 증가식 'p++'은 **증가 전 주소**를 식의 값으로 사용한다. 따라서 '*(p++)'은 처음 p가 가리킨 a[2]의 값인 3을 얻는다.

계산 과정

C 배열의 첨자는 0부터 시작하므로 a[0]은 1, a[1]은 2, a[2]는 3이다. p에는 &a[2]가 저장되어 있으므로 **처음 p가 가리키는 값은 3**이다.

후위 증가식 'p++'의 식 값은 **증가 전의 포인터 값**이다. 동시에 포인터 p 자체는 다음 int 원소를 가리키도록 증가한다. 따라서 '*(p++)'의 역참조에는 증가 전 주소인 &a[2]가 사용된다.

결국 '*(p++)'에서 얻는 값은 a[2]의 값인 **3**으로, ②와 일치한다. 이 식의 평가가 완료된 뒤 p는 a[3]을 가리키지만, 그 위치의 값 4가 이번 식의 결과가 되는 것은 아니다.

NoteBase

문 32. 인터프리터 방식에 관한 설명으로 옳은 것만을 <보기>에서 고른 것은?

<보기>
ㄱ. 실행될 행 단위마다 번역한다.
ㄴ. 프로그램 전체를 한꺼번에 번역한다.
ㄷ. 큰 기억장치가 필요하지 않으며 실행 결과를 위해 목적 언어로 번역한다.
ㄹ. 반복문이나 계속 호출되는 서브 프로그램이 많은 경우, 프로그램 실행 시간이 길어진다.

① ㄱ, ㄴ
② ㄱ, ㄹ
③ ㄴ, ㄷ
④ ㄷ, ㄹ

정답 2번

간단한 해설

인터프리터는 **실행할 문장을 해석하면서 실행**하는 방식이다. 전체를 미리 목적 코드로 번역하지 않으며, 반복 실행에서도 해석 부담이 생기는 것이 전통적인 특징이다.

보기별 해설

ㄱ. O: 전통적인 인터프리터 방식은 실행할 문장을 만날 때마다 **문장 단위로 해석하여 실행**한다. 문제의 행 단위 번역은 이 실행 방식을 가리킨다.

ㄴ. X: 프로그램 전체를 미리 번역하여 목적 프로그램을 만드는 것은 **컴파일러 방식**이다. 인터프리터는 실행 과정에서 문장을 해석한다.

ㄷ. X: 인터프리터를 프로그램 전체의 **목적 코드를 생성하는 방식**으로 설명한 부분이 틀렸다. 전통적인 인터프리터는 별도의 목적 프로그램을 만들지 않고 해석과 실행을 진행한다. 기억장치 사용량은 구현과 프로그램에 따라 달라지므로 메모리 절약만으로 방식 전체를 판정하지 않는다.

ㄹ. O: 전통적인 인터프리터는 같은 문장이 다시 실행될 때에도 해석 과정이 반복되므로 **반복문이나 자주 호출되는 서브 프로그램의 처리 부담**이 커질 수 있다. 이는 미리 번역한 목적 코드를 실행하는 방식과 비교한 설명이다.

옳은 보기는 ㄱ과 ㄹ이므로 ②와 일치한다.

NoteBase

문 33. IPv4와 IPv6 주소 체계에 관한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보기>

ㄱ. IPv6는 48bit의 주소 공간을 갖는다.

ㄴ. IPv4와 IPv6의 주소를 하나의 주소 내에 함께 표현할 수 있다.

ㄷ. IPv6에서는 주소 생략법을 통해 긴 주소를 간단히 표현할 수 있다.

ㄹ. IPv4는 프로토콜 내에 보안 관련 기능을 탑재할 수 있도록 설계하였다.

ㅁ. IPv6는 트래픽을 효과적으로 분류하기 위해 IPv6 헤더 내에 플로 레이블(flow label) 필드가 정의되어 있다.

① ㄱ, ㄴ

② ㄴ, ㄷ, ㅁ

③ ㄷ, ㄹ, ㅁ

④ ㄴ, ㄷ, ㄹ, ㅁ

정답 2번

간단한 해설

IPv6는 **128bit 주소**를 사용하며 IPv4 혼합 표기와 주소 생략, 플로 레이블을 지원한다. 설계상 보안 확장을 포함한 IPv6의 특징과 **IPv4에도 IPsec을 추가 적용할 수 있다는 사실**은 구별한다.

NoteBase

보기별 해설

ㄱ. X: IPv6 주소의 길이는 48bit가 아니라 **128bit**이다. 16bit씩 8개 부분으로 나누어 16진수로 표시한다. IPv4 주소의 길이는 32bit이다.

ㄴ. O: IPv4 주소를 포함하는 IPv6 주소는 앞부분을 IPv6의 16진수 형식으로, 하위 32bit를 IPv4의 점으로 구분한 10진수 형식으로 함께 표시할 수 있다. 예를 들어 ::ffff:192.0.2.1은 이러한 **혼합 표기**이다. 서로 독립적인 주소 두 개를 단순히 이어 붙이는 것이 아니라, 하나의 128bit 주소 안에 IPv4 주소를 표현하는 방식이다.

ㄷ. O: 각 16bit 부분의 선행 0을 생략할 수 있고, 연속하여 0으로 채워진 부분은 '::'로 압축할 수 있다. 생략된 0의 위치와 개수를 복원할 수 있어야 하므로 **'::'는 한 주소에서 한 번만** 사용한다.

ㄹ. X: IPv4와 IPv6의 설계상 특징을 비교하는 이 문항에서, 인증·무결성·기밀성을 위한 **보안 확장을 명시적으로 포함한 것**은 IPv6이다. IPv4의 원래 Security 옵션은 보안등급과 취급 제한 등을 전달하는 표지로서, 데이터 인증이나 암호화 자체를 제공하는 기능과는 다르다. 다만 IPv4에도 이후 정의된 IPsec을 적용할 수 있으므로, IPv4에서는 보안 기능을 사용할 수 없다는 뜻은 아니다. 또한 IPv6를 사용한다는 이유만으로 모든 통신이 자동으로 암호화되는 것도 아니다.

ㅁ. O: IPv6 기본 헤더에는 **20bit의 플로 레이블** 필드가 있다. 같은 흐름에 속하는 패킷을 식별하여 흐름별 처리가 가능하도록 하며, 송신지·수신지 주소 등과 함께 트래픽을 분류하는 데 이용한다. 레이블이 있다는 것만으로 특정 서비스 품질이 자동 보장되는 것은 아니다.

따라서 옳은 보기는 ㄴ, ㄷ, ㅁ이며, 이 조합은 ②에 해당한다.

NoteBase

문 34. 파이선(python) 프로그램 결과의 출력값으로 옳은 것은?

```
fruits = ['apple', 'banana', 'orange', 'apple',
'apple', 'banana']
unique_fruits = set(fruits)
print(len(unique_fruits))
```

① 3
② 4
③ 5
④ 6

정답 1번

간단한 해설

set()은 **중복된 값을 하나로 합친 집합**을 만든다. 과일 이름은 'apple', 'banana', 'orange'의 세 종류이므로 집합의 길이는 3이다.

계산 과정

원래 fruits 리스트에는 'apple'이 3개, 'banana'가 2개, 'orange'가 1개로 총 6개의 원소가 들어 있다.
리스트는 같은 값을 여러 번 저장할 수 있다.

set(fruits)는 리스트를 **중복을 허용하지 않는 집합**으로 변환한다. 따라서 unique_fruits에 남는 서로 다른 값은 'apple', 'banana', 'orange'의 세 가지이다. 집합의 표시 순서는 이 문제의 결과에 영향을 주지 않는다.

len(unique_fruits)는 변환된 집합의 원소 수를 구한다. 따라서 print가 출력하는 값은 **3**이며, ①과 일치한다.

NoteBase

문 35. 컴퓨터 네트워크 관리에서 장애관리(fault management)에 관한 설명으로 옳지 않은 것은?

① 사전 장애관리와 사후 장애관리가 있다.

② 장애 유무가 발견되면 문제가 발생한 위치와 정보를 파악한 후 원인을 분석한다.

③ 네트워크 자원과 정보를 보호하며, 패스워드와 인증, 접근 제어 정보를 관리하고 분배한다.

④ 모니터링 관리 프로토콜로 CMIP와 SNMP를 이용하는 방식, ICMP의 ping을 이용하는 방식 등을 사용한다.

정답 3번

간단한 해설

장애관리는 네트워크의 **장애를 예방·탐지하고 원인을 찾아 복구**하는 기능이다. 패스워드, 인증 및 접근 제어를 통해 자원과 정보를 보호하는 것은 **보안관리**이므로 옳지 않은 설명은 ③이다.

선지별 해설

① O: 사전 장애관리는 상태 감시와 점검으로 장애를 예방하거나 발생 징후를 조기에 발견하는 것이고, 사후 장애관리는 발생한 장애의 원인을 분석하고 정상 상태로 복구하는 것이다.

② O: 장애가 발견되면 영향을 받는 장비·회선 등 발생 위치와 관련 정보를 수집하여 원인을 분석한다. 원인을 파악한 뒤 적절한 복구 조치를 수행한다.

③ X: 네트워크 자원과 정보를 보호하고 패스워드·인증·접근 제어 정보를 관리하는 것은 **보안관리**이다. 장애관리는 장애의 발견, 위치 파악, 원인 분석 및 복구를 담당한다.

④ O: CMIP와 SNMP는 관리 대상 장비의 상태나 이벤트 정보를 수집하는 데 사용할 수 있는 네트워크 관리 프로토콜이다. ping은 ICMP 에코 요청과 응답을 이용하여 대상 장비의 응답 여부와 도달 가능성을 점검하므로 장애 진단에 활용된다. 다만 응답이 없다는 사실만으로 장비 고장을 단정할 수는 없다.

NoteBase

문 36. 자바 프로그래밍에서 ACore 클래스는 다중 상속을 지원하는 클래스이며, BCore 클래스에게 상속한다. ( ) 안에 들어갈 용어로 옳은 것은?

```
( ㄱ ) ACore {
                String guest;
    // 이하 생략
}
```

```
public class BCore ( ㄴ ) ACore {
            public void Data(){
        System.out.println("Test!");
     // 이하 생략
   }
}
```

① ㄱ: interface, ㄴ: extends

② ㄱ: interface, ㄴ: implements

③ ㄱ: abstract, ㄴ: extends

④ ㄱ: abstract, ㄴ: implements

정답 2번

간단한 해설

자바의 다중 인터페이스 관계에서 **인터페이스 선언은 interface, 클래스의 인터페이스 구현은 implements**를 사용한다. 클래스 상속을 나타내는 extends와 인터페이스 구현을 구별한다.

NoteBase

빈칸별 해설

ㄱ: **interface**. 자바에서 클래스는 여러 클래스를 동시에 상속할 수 없지만, 인터페이스는 여러 인터페이스를 상속할 수 있고 클래스도 여러 인터페이스를 구현할 수 있다. 따라서 다중 인터페이스 관계를 표현하는 ACore의 선언 키워드는 interface이다. abstract는 추상 클래스 선언에 쓰이는 제한자이며, 그것만으로 클래스의 다중 상속이 가능해지지 않는다. 다만 인터페이스 필드는 선언 시 초기화식이 필요하므로, 제시된 String guest;는 컴파일 오류를 일으킨다. 이는 빈칸의 의도된 답과 별개의 원본 코드 결함이다.

ㄴ: **implements**. BCore는 class로 선언되므로 ACore 인터페이스를 구현할 때 implements를 사용한다. extends는 클래스가 다른 클래스를 상속하거나 인터페이스가 다른 인터페이스를 상속할 때 사용한다.

따라서 ㄱ에는 interface, ㄴ에는 implements가 들어가는 ②의 조합이 맞다.

NoteBase

문 37. 다음 연산식에서 이진수 Y의 값으로 옳은 것은?

$(10110101)_2 + Y = (11101111)_2$

① $(01110101)_2$
② $(11001001)_2$
③ $(11001110)_2$
④ $(00111010)_2$

정답 4번

간단한 해설

알려진 합에서 가수를 빼면 $Y = 239 - 181 = 58$이다. 이를 이진수로 나타내면 $(00111010)_2$이다.

계산 과정

주어진 식은 $(10110101)_2 + Y = (11101111)_2$이므로, **합에서 알려진 가수를 빼면** 미지의 가수 Y를 구할 수 있다. 따라서 $Y = (11101111)_2 - (10110101)_2$이다.

이진수의 각 자릿값은 오른쪽부터 $2^0, 2^1, \ldots, 2^7$이다. 1인 자리의 값을 더하여 십진수로 바꾸면 다음과 같다.
$(11101111)_2 = 128 + 64 + 32 + 8 + 4 + 2 + 1 = 239$
$(10110101)_2 = 128 + 32 + 16 + 4 + 1 = 181$
따라서 두 수의 차는 $239 - 181 = 58$이다.

$58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1$이므로 **해당 자릿값의 비트만 1**로 두면 $Y = (00111010)_2$이다. 앞의 0은 값을 바꾸지 않으며, 주어진 선택지와 같은 8자리 형식으로 나타낸 것이다.

검산하면 $181 + 58 = 239$이므로 원래 식을 만족한다. 따라서 $Y = (00111010)_2$인 ④와 일치한다.

NoteBase

문 38. 네트워크 토폴로지(topology)에 관한 설명이다. ( ) 안에 들어갈 용어로 옳은 것은?

(ㄱ): 중앙에 있는 전송제어장치에 장애가 발생하면 전체 네트워크가 작동할 수 없고, 허브가 통신망의 처리 능력과 신뢰성을 좌우한다.
(ㄴ): 케이블의 전송 거리가 멀어지면 신호가 점점 약해지기 때문에 증폭기를 사용해야 하며, 연결된 노드 수가 많으면 충돌이 잦아져서 네트워크의 성능이 떨어진다.

① ㄱ: 버스형, ㄴ: 링형
② ㄱ: 트리형, ㄴ: 링형
③ ㄱ: 트리형, ㄴ: 버스형
④ ㄱ: 버스형, ㄴ: 트리형

정답 3번

간단한 해설

트리형은 **중심 장치를 뿌리로 하는 계층 구조**여서 중심 허브에 대한 의존성이 있다. 버스형은 **공통 전송 매체를 공유**하므로 노드가 늘면 충돌과 성능 저하가 발생하기 쉽다.

빈칸별 해설

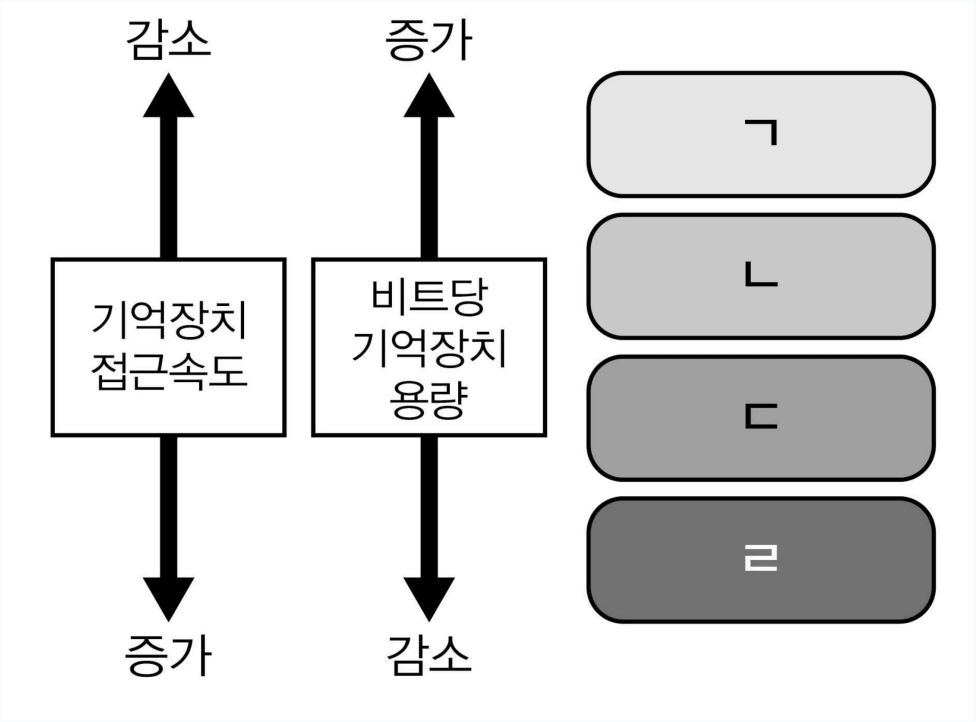
ㄱ은 **트리형**이다. 트리형은 상위의 중심 장치를 뿌리로 하여 하위 장치나 망을 계층적으로 연결한다. 중심 허브에 의존하는 구조에서는 그 장치의 성능이 통신 처리 능력을 좌우하고, 중심 장치의 장애가 망 전체의 통신에 영향을 줄 수 있다. 중앙 장치 의존성은 성형에서도 나타나지만, 제시된 선택지의 트리형 역시 이러한 특성을 가진다.

ㄴ은 **버스형**이다. 여러 노드가 하나의 공통 전송 매체를 공유하므로, 노드가 많아져 동시에 전송하려는 시도가 늘면 충돌 가능성이 커진다. 또한 전송 거리가 길어질수록 신호가 감쇠하므로 허용 거리를 늘리려면 신호를 보강·재생하는 장치가 필요하다. 이는 노드들을 고리 모양으로 연결하는 링형과 구별된다.

따라서 ㄱ은 트리형, ㄴ은 버스형이며, 이 조합은 ③이다.

NoteBase

문 39. 메모리 계층구조에 관한 그림의 ㄱ~ㄹ에 해당하는 설명을 옳게 연결한 것은?



a. 보조기억장치로서 주기억장치의 제한된 용량을 지원한다.

b. 캐시 메모리로 실행 중인 프로그램의 명령어와 데이터를 저장한다.

c. 중앙처리장치와 정보교환을 위해 프로그램과 데이터를 기억하는 장치이다.

d. 중앙처리장치 내부에 존재하며, 프로그램을 실행하는 데 필요한 중간값을 임시로 저장할 때 쓰이는 장치이다.

① ㄱ: a, ㄴ: b, ㄷ: c, ㄹ: d

② ㄱ: a, ㄴ: c, ㄷ: b, ㄹ: d

③ ㄱ: d, ㄴ: c, ㄷ: a, ㄹ: b

④ ㄱ: d, ㄴ: b, ㄷ: c, ㄹ: a

정답 2번

간단한 해설

그림에서 **접근 속도가 증가하는 아래쪽 끝은 레지스터**이다. 위에서부터 보조기억장치·주기억장치·캐시·레지스터로 구분하면 각 설명을 연결할 수 있다.

NoteBase

도식 해설

그림에서는 **아래쪽으로 갈수록 접근 속도가 증가**한다. 따라서 위에서부터 보조기억장치, 주기억장치, 캐시 메모리, 레지스터를 배치한다. 보통 빠른 계층일수록 저장 용량은 작으며, 이 그림의 용량 증가 방향은 위쪽이다.

ㄱ → a: 가장 위의 ㄱ은 네 계층 중 접근 속도가 가장 느리고 대용량 저장을 담당하는 **보조기억장치**이다. 디스크나 SSD 등이 이에 해당하며, 주기억장치에 모두 담지 못하는 프로그램과 데이터를 보관한다.

ㄴ → c: ㄴ은 보조기억장치와 캐시 사이의 **주기억장치**이다. CPU가 실행할 프로그램과 데이터를 저장하고 CPU와 정보를 주고받는 작업 공간이다.

ㄷ → b: ㄷ은 주기억장치보다 빠르고 레지스터보다 느린 **캐시 메모리**이다. 실행 중 자주 이용하는 명령어와 데이터의 사본을 가까이 두어 주기억장치 접근 부담을 줄인다.

ㄹ → d: 가장 아래의 ㄹ은 CPU 내부의 **레지스터**이다. 연산에 필요한 피연산자나 중간 결과 등을 임시 저장하며, 제시된 계층 중 가장 빠르게 접근한다.

따라서 ㄱ-a, ㄴ-c, ㄷ-b, ㄹ-d의 연결이므로 ②와 일치한다.

NoteBase

문 40. 교착상태(dead lock)에 관한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보기>
ㄱ. 상호배제 기법을 통해 교착상태에서 벗어날 수 있다.
ㄴ. 회피하기 위해서는 임계 구역 개념을 사용해야 한다.
ㄷ. 두 개 이상의 연관된 프로세스가 동시에 실행되는 것을 말한다.
ㄹ. 다중 프로그래밍 시스템에서 하나 이상의 프로세스가 아무리 기다려도 발생하지 않을 특정 사건을 기다리고 있는 상태이다.

① ㄱ
② ㄹ
③ ㄴ, ㄷ
④ ㄷ, ㄹ

정답 2번

간단한 해설

교착상태는 프로세스들이 서로의 자원이나 진행을 기다려 **더 이상 진행하지 못하는 상태**이다. 상호배제는 발생 조건이고, 회피는 안전 상태를 유지하도록 자원 할당을 판단하는 방식이다.

NoteBase

보기별 해설

ㄱ. (X) **상호배제는 교착상태의 발생 조건** 중 하나이지, 이미 발생한 교착상태를 해소하는 기법이 아니다. 교착상태에서 회복하려면 관련 프로세스를 종료하거나 자원을 선점하는 등의 조치가 필요하다.

ㄴ. (X) **교착상태 회피는 자원 할당 후에도 안전 상태가 유지되는지 판단**하는 방식이다. 대표적으로 은행원 알고리즘이 있다. 임계 구역은 공유 자원에 접근하는 코드 구간으로, 그 접근을 조정하는 것은 상호배제와 동기화에 관한 문제이다.

ㄷ. (X) 여러 프로세스가 동시에 실행되거나 실행을 번갈아 진행하는 것은 병렬성·동시성에 관한 설명이다. 교착상태는 프로세스들이 **서로가 보유한 자원이나 서로의 진행에 의존하는 사건을 기다려 더 이상 진행하지 못하는 상태**이다.

ㄹ. (O) 교착상태에 빠진 프로세스는 다른 대기 프로세스가 자원을 해제하는 등의 사건을 기다리지만, 그 프로세스도 진행하지 못하므로 기다리는 사건이 발생하지 않는다. 따라서 기다리는 것만으로는 상태가 해소되지 않는다는 설명이다.

옳은 보기는 **ㄹ**이므로 이를 고른 ②가 정답이다.

NoteBase