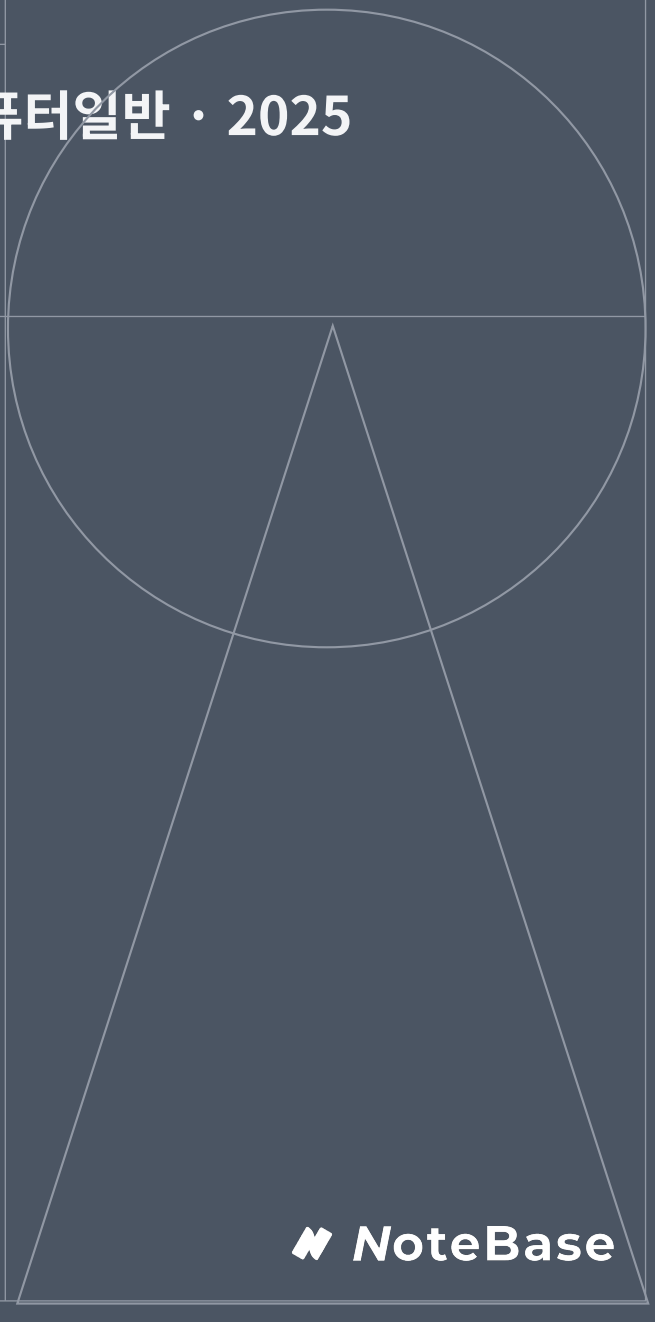
기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2025



NoteBase

기출OX 해설집

공무원/소방/경력채용 · 컴퓨터일반 · 2025

공무원/소방/경력채용 · 컴퓨터일반 · 2025

문 1. 명령어에 대한 설명으로 옳은 것을 <보기>에서 고른 것은?

<보기>
ㄱ. 연산 부분은 수행하는 기능을 의미한다.
ㄴ. 일반적으로 명령어를 구성하는 피연산자 수가 많으면 많은 수의 명령어로 원하는 작업을 수행할 수 있다.
ㄷ. 피연산자 수가 없거나 2개 또는 3개인 명령어도 있다.
ㄹ. 피연산자 A(주소 A의 자료)를 레지스터에 가져오는 명령어는 ADD이다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄷ, ㄹ

정답 2번

간단한 해설

연산 코드는 수행할 기능을 지정하며, 명령어에는 0주소·2주소·3주소 등 여러 형식이 있다. 주소 필드가 많을수록 보통 같은 작업을 적은 명령어로 표현할 수 있고, 메모리 자료를 레지스터로 읽는 동작은 ADD가 아니라 LOAD에 해당한다.

NoteBase

보기별 해설

ㄱ. (O) 명령어의 연산 부분, 즉 연산 코드(opcode)는 덧셈·전송·분기 등 **어떤 동작을 수행할지** 지정한다. 피연산자 부분은 그 동작의 대상이 되는 값이나 위치를 지정한다.

ㄴ. (X) 일반적으로 한 명령어에 명시할 수 있는 피연산자나 주소가 많으면 연산 대상과 결과 저장 위치를 더 직접 지정할 수 있어, 같은 작업을 더 적은 수의 명령어로 표현할 수 있다. 많은 수의 명령어가 필요하다는 설명은 반대이다. 다만 구체적인 명령어 수는 명령어 집합과 작업에 따라 달라진다.

ㄷ. (O) 명시적인 피연산자 주소가 없는 0주소 명령어뿐 아니라 1주소·2주소·3주소 명령어 형식이 있다. 0주소 형식은 스택의 상단 값 등을 암묵적으로 사용하므로, 명시된 피연산자가 없다는 뜻이지 실제 연산 대상이 없다는 뜻은 아니다.

ㄹ. (X) 메모리 주소 A의 데이터를 레지스터로 가져오는 것은 일반적으로 LOAD 명령의 역할이다. ADD는 **덧셈 연산**을 수행하므로 단순히 자료를 가져오는 명령과 구별된다.

옳은 보기는 ㄱ과 ㄷ이므로 ②의 조합이 맞다.

NoteBase

문 2. 파이프라인(Pipeline)에 대한 설명으로 옳은 것을 <보기>에서 고른 것은?

<보기>
ㄱ. CPU 내부에 여러 개의 명령어 파이프라인을 포함하는 구조를 슈퍼스칼라(Superscalar)라고 한다.
ㄴ. 파이프라인의 명령어 처리 단계 수를 늘리면 동시에 수행되는 명령어 수가 늘어나 속도가 감소한다.
ㄷ. 파이프라인에서는 구조적(Structural)인 위험(Hazard)과 데이터(Data) 및 제어(Control) 위험이 있다.
ㄹ. 동시에 여러 개의 명령어를 실행하는 기법이다.

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

정답 3번

간단한 해설

파이프라인은 **여러 명령어의 처리 단계를 겹쳐 실행**하여 단위 시간당 완료되는 명령어 수를 높이는 기법이다.
단계 수 증가로 동시 처리 수가 늘어나는 것을 속도 감소의 원인으로 설명한 ㄴ은 옳지 않다.

NoteBase

보기별 해설

ㄱ. O: 슈퍼스칼라는 여러 명령어를 한 클록에 발행하고 병렬로 처리할 수 있도록 복수의 실행 경로와 파이프라인을 두는 구조이다. 하나의 파이프라인에서 단계만 겹치는 방식보다 명령어 수준 병렬성을 확대한다.

ㄴ. X: 파이프라인 단계를 세분화하면 이상적인 조건에서 각 단계의 처리 시간이 짧아지고, 서로 다른 단계에서 겹쳐 처리할 수 있는 명령어 수가 늘어 처리율을 높일 수 있다. 따라서 동시 처리 명령어 수가 늘어나서 속도가 감소한다는 설명은 잘못이다. 다만 실제 성능은 단계 사이의 부가 비용과 위험에 따른 지연 때문에 단계 수에 비례해 무한히 높아지지는 않는다.

ㄷ. O: 구조적 위험은 여러 명령어가 같은 하드웨어 자원을 동시에 요구할 때, 데이터 위험은 필요한 피연산자 결과가 아직 준비되지 않았을 때 발생할 수 있다. 제어 위험은 분기 등으로 다음에 실행할 명령어가 확정되지 않을 때 발생한다.

ㄹ. O: 파이프라인은 명령어 처리를 여러 단계로 나누어, **서로 다른 명령어의 서로 다른 단계를 겹쳐 수행**한다. 한 명령어의 전체 처리를 끝낸 다음 다음 명령어를 시작하는 방식이 아니므로, 여러 명령어가 동시에 처리 중인 상태가 된다.

옳은 보기는 ㄱ, ㄷ, ㄹ이므로 정답은 ③이다.

NoteBase

문 3. 기억장소의 계층 구조에서 속도가 빠른 순으로 옳은 것은?

① 레지스터 — 캐시메모리 — 주기억장치 — 보조기억장치

② 캐시메모리 — 레지스터 — 주기억장치 — 보조기억장치

③ 캐시메모리 — 주기억장치 — 레지스터 — 보조기억장치

④ 주기억장치 — 보조기억장치 — 레지스터 — 캐시메모리

정답 1번

간단한 해설

기억장치 계층에서는 일반적으로 **CPU 내부에서 직접 사용하는 저장 공간일수록 접근 속도가 빠르다**.
고속·소용량 계층과 저속·대용량 계층을 구별하면 순서를 정할 수 있다.

올바른 순서

레지스터 → 캐시메모리 → 주기억장치 → 보조기억장치

레지스터: CPU 내부에서 명령 실행에 필요한 피연산자나 중간 결과를 직접 보관한다. CPU가 연산 과정에서 바로 사용하는 **가장 가까운 저장 공간**이므로 제시된 기억장소 중 접근 속도가 가장 빠르다.

캐시메모리: CPU가 자주 사용하는 명령어와 데이터의 사본을 저장한다. 레지스터보다는 느리지만 **주기억장치에 접근하는 지연을 줄이기 위한 고속 기억장치**이므로 주기억장치보다 빠르다.

주기억장치: 실행 중인 프로그램과 데이터를 담는 RAM 등이 해당한다. 캐시보다 큰 작업 공간을 제공하지만 접근 속도는 상대적으로 느리며, **보조기억장치보다 빠르게**CPU에 필요한 데이터를 제공한다.

보조기억장치: SSD나 하드디스크처럼 프로그램과 데이터를 장기간 보관하는 장치이다. 큰 용량과 비휘발성 저장을 제공하는 대신, 제시된 계층에서는 **주기억장치보다 접근 속도가 느리다**.

NoteBase

문 4. 명령어 주소 지정 방식에 대한 설명으로 옳지 않은 것은?

① 직접 주소 지정 방식(Direct addressing mode)은 명령어에 지정된 주소에서 데이터를 읽거나 쓰는 방식이다.

② 즉시 주소 지정 방식(Immediate addressing mode)은 명령어 데이터가 저장된 레지스터를 참조하는 방식이다.

③ 간접 주소 지정 방식(Indirect addressing mode)은 명령어에 지정된 주소가 데이터를 저장하고 있는 주소를 가리키는 방식이다.

④ 상대 주소 지정 방식(Relative addressing mode)은 프로그램 카운터를 기준으로 상대적인 주소를 계산하여 데이터를 참조하는 방식이다.

정답 2번

간단한 해설

즉시 주소 지정은 피연산자 값 자체가 명령어에 포함되는 방식이다. 데이터가 들어 있는 레지스터를 지정하여 읽는 방식은 레지스터 주소 지정이므로 둘을 구별해야 한다.

선지별 해설

① (O) 직접 주소 지정에서는 명령어의 주소 필드가 데이터가 있는 메모리 주소를 직접 나타낸다. 주소 필드를 A라고 하면 유효 주소는 A이며, 그 주소의 데이터를 읽거나 쓴다.

② (X) 즉시 주소 지정에서는 **사용할 데이터 값 자체를 명령어에 넣는다**. 예를 들어 상수 5가 명령어의 피연산자로 포함되어 있으면 별도의 데이터 저장 위치를 찾아갈 필요가 없다. 데이터가 저장된 레지스터를 참조한다는 설명은 레지스터 주소 지정에 해당한다.

③ (O) 간접 주소 지정에서는 명령어가 지정한 위치에 실제 데이터의 주소가 저장되어 있다. 먼저 그 위치에서 유효 주소를 얻고, 다시 그 유효 주소로 데이터를 참조한다. 직접 주소 지정에 비해 주소를 찾는 단계가 추가된다.

④ (O) 상대 주소 지정에서는 프로그램 카운터(PC) 값에 명령어의 변위 값을 더하여 유효 주소를 계산한다. 즉 **PC를 기준으로 한 상대 위치**를 사용하며, 정확히 어떤 시점의 PC를 기준으로 삼는지는 해당 명령어 집합의 정의를 따른다.

NoteBase

문 5. 데이터마이닝 기법에 대한 설명으로 옳지 않은 것은?

① 연관 규칙(Association rules)은 관련성이 강한 데이터의 조합을 통해 패턴을 발견한다.

② 군집 규칙(Clustering rules)은 상호 유사한 특성을 지닌 데이터를 그룹화하여 패턴을 분석한다.

③ 분류 규칙(Classification rules)은 이미 알려진 특정 그룹의 특징을 부여하고 기준에 맞게 분류한다.

④ 특성화 규칙(Characterization rules)은 트랜잭션 이력 데이터를 시계열로 분석하여 트랜잭션의 향후 발생 가능성을 예측한다.

정답 4번

간단한 해설

특성화는 **대상 집단의 공통적인 성질을 요약**하는 기법이다. 거래 이력을 시간 순서로 분석해 미래 발생을 예측한다는 설명과 구분해야 한다.

선지별 해설

① (O) 연관 규칙은 상품의 동시 구매처럼 **함께 나타나는 항목 사이의 관계**를 찾는다. 관련성이 높은 데이터 조합에서 반복되는 패턴을 발견한다는 설명은 옳다.

② (O) 군집화는 정답 범주가 미리 주어지지 않은 데이터들을 유사성에 따라 묶는다. 같은 군집 안의 데이터는 서로 비슷하고 다른 군집과는 구별되도록 그룹을 형성하여 패턴을 분석한다.

③ (O) 분류는 이미 범주가 알려진 데이터에서 각 범주의 특징과 판별 기준을 학습하고, 그 기준으로 새 데이터의 범주를 결정한다. 따라서 알려진 그룹의 특징을 이용해 분류한다는 설명은 옳다.

④ (X) **특성화는 특정 데이터 집단의 일반적인 특징을 요약**하는 것이다. 예를 들어 우수 고객 집단의 공통적인 구매 특성을 정리한다. 거래의 시간적 이력에서 순서나 추세를 찾아 향후 발생을 예측하는 것은 순차 패턴·시계열 분석에 관한 설명이지 특성화의 정의가 아니다.

NoteBase

문 6. 데이터베이스 시스템에 대한 설명으로 옳지 않은 것은?

① 데이터베이스 시스템에는 하나의 데이터베이스만 존재한다.

② 데이터베이스를 이용하여 데이터를 저장하고 처리하는 시스템을 말한다.

③ 데이터베이스 시스템 사용자에는 일반사용자(End user), 응용 프로그래머(Application programmer), 데이터베이스 관리자(Database administrator)가 있다.

④ 데이터베이스 관리시스템(DBMS)은 사용자에게 데이터 정의 기능, 데이터 조작 기능, 데이터 제어 기능을 제공한다.

정답 1번

간단한 해설

데이터베이스 시스템은 데이터베이스, DBMS, 응용 프로그램 및 사용자 등이 함께 데이터를 관리하는 환경이다. 하나의 시스템에서 **여러 데이터베이스를 관리할 수 있으므로** 반드시 하나만 존재한다는 ①이 옳지 않다.

선지별 해설

① X: 데이터베이스 시스템에 존재하는 데이터베이스의 수가 하나로 제한되는 것은 아니다. 업무나 목적에 따라 여러 데이터베이스를 구성하여 관리할 수 있다. 따라서 **'하나의 데이터베이스만 존재한다'**라는 제한이 잘못되었다.

② O: 데이터베이스 시스템은 데이터베이스를 이용하여 데이터를 저장·검색·처리하는 시스템이다. DBMS는 사용자나 응용 프로그램과 데이터베이스 사이에서 데이터 관리를 담당한다.

③ O: 일반사용자는 응용 프로그램이나 질의 도구로 데이터를 이용하고, 응용 프로그래머는 데이터베이스를 이용하는 프로그램을 개발한다. 데이터베이스 관리자는 데이터베이스의 운영, 접근 권한, 백업·복구 등을 관리한다.

④ O: **데이터 정의** 기능은 데이터 구조와 제약조건 등을 정의하고, **데이터 조작** 기능은 검색·삽입·수정·삭제를 수행한다. **데이터 제어** 기능은 접근 권한, 무결성 및 동시 접근 등을 관리하여 데이터를 일관되고 안전하게 사용할 수 있도록 한다.

NoteBase

문 7. SQL 명령어 중 성격이 나머지와 다른 것은?

① Alter

② Create

③ Drop

④ Update

정답 4번

간단한 해설

Alter, Create, Drop은 데이터베이스 객체의 구조를 다루는 **데이터 정의어(DDL)**이다. Update는 테이블에 저장된 값을 수정하는 **데이터 조작어(DML)**이므로 나머지와 성격이 다르다.

선지별 해설

① Alter: 데이터 정의어(DDL)이다. 테이블에 열을 추가하거나 열의 정의를 바꾸는 등 데이터베이스 객체의 구조를 변경한다.

② Create: 데이터 정의어(DDL)이다. 테이블이나 뷰 등 새로운 데이터베이스 객체를 생성한다.

③ Drop: 데이터 정의어(DDL)이다. 테이블이나 뷰 등 데이터베이스 객체 자체를 제거한다. 기존 테이블의 일부 행만 삭제하는 명령과는 구별된다.

④ Update: **데이터 조작어(DML)**이다. 이미 존재하는 테이블에서 행의 열 값을 수정한다. 나머지 명령들이 객체의 구조를 정의·변경·제거하는 데 쓰이는 것과 달리, 저장된 데이터 값을 다룬다.

NoteBase

문 8. 데이터베이스 설계 절차로 옳은 것은?

① 요구 사항 분석 → 물리적 설계 → 개념적 설계 → 논리적 설계 → 구현
② 요구 사항 분석 → 논리적 설계 → 개념적 설계 → 물리적 설계 → 구현
③ 요구 사항 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현
④ 요구 사항 분석 → 개념적 설계 → 물리적 설계 → 논리적 설계 → 구현

정답 3번

간단한 해설

먼저 필요한 데이터와 업무를 파악하고, 데이터의 의미를 논리 구조와 물리 저장 방식으로 점차 구체화한다. **설계가 확정된 뒤 실제 데이터베이스를 생성**한다.

올바른 순서

요구 사항 분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

요구 사항 분석: 사용자와 업무가 필요로 하는 데이터, 처리 작업, 제약 조건 등을 파악한다. **무엇을 저장하고 어떤 처리를 지원할지**가 정해져야 설계의 범위와 기준을 세울 수 있다.

개념적 설계: 분석한 요구를 개체·속성·관계 등의 개념으로 표현하여 업무의 데이터 구조를 만든다. 보통 ER 모델 등을 사용하며, **구체적인 저장 방식에 앞서 데이터의 의미와 관계**를 확정한다.

논리적 설계: 개념 모델을 관계형 모델의 테이블·키·제약 조건 등 논리적 구조로 변환한다. 필요한 정규화 등을 검토하여 **데이터를 논리적으로 조직할 방법**을 정한 뒤 물리적 저장을 설계한다.

물리적 설계: 논리적 구조를 실제 DBMS와 저장 환경에서 효율적으로 구현하도록 인덱스, 저장 배치, 접근 경로 등을 정한다. **확정된 논리 구조와 업무 부하**가 있어야 성능·공간을 고려한 구체적 설계가 가능하다.

구현: 설계한 스키마와 저장 구조를 DDL 등으로 생성하고 데이터를 적재하며 필요한 처리를 구현·시험한다. **설계 내용을 실제 데이터베이스로 구축**하는 단계이다.

NoteBase

문 9. 트랜잭션의 특성 중, 구성하는 작업들이 전부 수행되거나 전혀 수행되지 않아야 하는 것에 해당되는 것은?

① 일관성(Consistency)

② 격리성(Isolation)

③ 원자성(Atomicity)

④ 지속성(Durability)

정답 3번

간단한 해설

모두 반영하거나 아무것도 반영하지 않는 성질은 원자성이다. 작업 도중 실패하면 앞서 한 일부 변경도 취소하여 트랜잭션의 부분 완료가 남지 않게 한다.

개념 구분

원자성은 트랜잭션을 나눌 수 없는 하나의 작업 단위로 취급하는 성질이다. 모든 작업이 성공하면 전체 결과를 확정하고, 중간에 실패하면 일부 변경만 남지 않도록 취소한다. 따라서 '전부 수행되거나 전혀 수행되지 않는다'는 설명은 ③ 원자성에 해당한다.

일관성은 트랜잭션 전후에 데이터베이스의 무결성 제약이 유지되는 성질이고,

격리성은 동시에 수행되는 트랜잭션들이 서로의 중간 처리에 부당한 영향을 주지 않도록 하는 성질이다.

지속성은 이미 성공적으로 확정된 결과가 이후 장애가 발생해도 보존되는 성질이다. **원자성은 일부만 반영되는 것을 막고, 지속성은 확정된 결과가 사라지는 것을 막는다**는 점에서 구별된다.

NoteBase

문 10. 빅데이터에 대한 설명으로 옳지 않은 것은?

① 대량의 정형 또는 비정형 데이터의 집합이다.

② 데이터로부터 가치를 추출하고 그 결과를 분석하는 기술을 의미하기도 한다.

③ 빅데이터를 이용하여 사람들의 행동, 생각과 의견까지 분석하고 예측할 수 있다.

④ 더그 레이니(Doug Laney)가 정의한 빅데이터 3V로는 속도(Velocity), 규모(Volume), 시각화(Visualization)가 있다.

정답 4번

간단한 해설

빅데이터의 3V는 **규모·속도·다양성**이다. 시각화는 분석 결과를 표현하는 방법으로, 3V의 다양성(Variety)을 대신하지 않는다.

선지별 해설

① (O) 빅데이터는 기존의 일반적인 처리 방식으로 다루기 어려울 만큼 규모가 크거나 복잡한 데이터 집합을 말한다. 정해진 형식의 정형 데이터뿐 아니라 문서·영상 등의 비정형 데이터도 포함한다.

② (O) 빅데이터라는 말은 데이터 자체뿐 아니라 이를 수집·저장·처리·분석하여 **유용한 정보와 가치를 얻는 기술**까지 넓게 가리키기도 한다.

③ (O) 검색·구매 기록이나 온라인 게시글 등에서 행동 양상과 의견의 경향을 분석하고 앞으로의 변화를 예측할 수 있다. 이는 데이터에 근거한 추정이지, 개인의 생각을 직접 읽거나 예측의 정확성을 보장한다는 의미는 아니다.

④ (X) 빅데이터의 3V는 **규모(Volume), 속도(Velocity), 다양성(Variety)**이다. 규모는 데이터의 양, 속도는 생성·유입·처리의 빠르기, 다양성은 데이터 형태와 출처의 다채로움을 뜻한다. 시각화(Visualization)를 다양성(Variety)으로 고쳐야 한다.

NoteBase

문 11. C언어에서 배열에 대한 설명으로 옳지 않은 것은?

① 문자 배열은 다차원으로 구성할 수 없다.

② 배열 요소의 자료형이 배열의 자료형이다.

③ 같은 자료형의 데이터를 순서대로 나열한 구조이다.

④ 배열의 형식에는 배열 이름과 인덱스가 있다.

정답 1번

간단한 해설

C언어의 문자 배열도 **2차원 이상의 다차원 배열**로 선언할 수 있다. 예를 들어 char names[3][10];은 문자 10개로 이루어진 배열 3개를 갖는 2차원 문자 배열이므로 ①이 옳지 않다.

선지별 해설

① X: 문자 배열은 다차원으로 구성할 수 있다. char names[3][10];처럼 선언하면 각 행에 문자들을 저장할 수 있으며, 널 문자로 끝나는 문자열을 여러 개 저장하는 용도로도 사용할 수 있다.

② O: 이 문항에서 '배열의 자료형'은 배열에 저장하는 **요소의 기본 자료형**을 뜻한다. 예를 들어 int a[5];는 int형 요소를 저장하는 정수 배열이다. 엄밀한 C의 타입 체계에서는 a 자체의 타입은 'int 5개로 이루어진 배열'이고, a[0]의 타입은 int이므로 배열 타입과 요소 타입 자체를 동일하다고 혼동해서는 안 된다.

③ O: 배열은 같은 자료형의 요소를 연속된 메모리 공간에 순서대로 저장하는 구조이다. 각 요소는 배열 안에서의 위치를 나타내는 인덱스로 구별한다.

④ O: 배열 요소는 a[i]처럼 **배열 이름과 인덱스**를 사용하여 참조한다. C에서 첫 번째 요소의 인덱스는 0이므로, int a[5];의 유효한 인덱스는 0부터 4까지이다.

NoteBase

문 12. 연결 리스트에 대한 설명으로 옳지 않은 것은?

① 구조체 변수와 주소 값으로 구성된다.

② 원형 연결 리스트의 마지막 노드는 NULL을 가리킨다.

③ 메모리의 동적 할당을 기반으로 구현된 리스트이다.

④ 이중 연결 리스트는 왼쪽 노드가 오른쪽 노드를 가리킴과 동시에 오른쪽 노드도 왼쪽 노드를 가리킨다.

정답 2번

간단한 해설

원형 연결 리스트는 마지막 노드에서 첫 번째 노드로 연결되어 순환한다. 마지막 링크를 NULL로 두는 선형 연결 리스트와 이 점에서 구별된다.

선지별 해설

① O: 연결 리스트는 데이터를 담는 부분과 다른 노드를 가리키는 링크로 구성된 노드들을 연결한 자료구조이다. C에서는 데이터와 다음 노드의 주소를 저장하는 포인터를 하나의 구조체로 묶어 노드를 구현할 수 있다.

② X: 원형 연결 리스트에서는 **마지막 노드가 첫 번째 노드를 가리켜** 고리를 이룬다. 마지막 노드의 다음 링크가 NULL인 것은 일반적인 선형 단일 연결 리스트의 특징이다.

③ O: 일반적인 포인터 기반 연결 리스트는 노드를 필요에 따라 동적으로 할당하고 연결한다. 노드들이 메모리에서 연속된 위치에 있을 필요는 없으며, 삽입·삭제 때 연결 관계를 바꾼다. 동적 할당이 대표적인 구현 방식이라는 뜻이지, 다른 구현이 불가능하다는 뜻은 아니다.

④ O: 이중 연결 리스트의 각 노드는 다음 노드와 이전 노드를 가리키는 두 링크를 가진다. 인접한 두 노드는 각각의 다음·이전 링크로 서로를 가리키므로 양방향으로 순회할 수 있다.

NoteBase

문 13. 스택에 대한 설명으로 옳지 않은 것은?

① 데이터를 삽입할 때는 pop( ) 함수를 사용한다.

② 스택에서 자료를 삭제할 때는 공백 여부를 확인해야 한다.

③ 가장 나중에 삽입된 데이터가 가장 먼저 삭제되는 구조이다.

④ 순차 자료구조를 이용한 스택은 고정된 배열 때문에 메모리의 사용 효율이 떨어진다.

정답 1번

간단한 해설

스택에서 **삽입은 push, 삭제는 pop**으로 수행한다. 스택은 마지막에 넣은 자료를 먼저 꺼내는 후입선출 구조이므로 두 연산 모두 꼭대기인 top에서 이루어진다.

선지별 해설

① (X) 스택에 새 데이터를 넣는 연산은 **push**이다. pop은 스택의 top에 있는 데이터를 꺼내 삭제하는 연산이므로, 삽입에 pop을 사용한다는 설명은 틀리다.

② (O) 빈 스택에는 삭제할 자료가 없으므로 pop을 수행하기 전에 공백 여부를 검사하거나 함수 내부에서 빈 상태를 처리해야 한다. 빈 스택에서 무조건 삭제하려 하면 스택 언더플로가 발생한다.

③ (O) 스택은 **후입선출(LIFO)** 구조이다. 예를 들어 A, B, C 순서로 넣었다면 중간에 다른 삽입이 없을 때 C, B, A 순서로 꺼낸다.

④ (O) 크기가 고정된 배열로 스택을 구현하면 최대 크기만큼 공간을 미리 확보해야 한다. 실제 원소 수가 적으면 사용하지 않는 공간이 생기고, 정해진 용량을 넘겨 넣을 수도 없어 메모리 활용에 제약이 있다. 이는 고정 크기 배열 구현의 특성이며 모든 스택 구현에 동일하게 적용되는 것은 아니다.

NoteBase

문 14. 그래프에 대한 설명으로 옳지 않은 것은?

① 정점의 수를 차수라고 한다.

② 무방향 그래프에서 간선(v0, v1)과 간선(v1, v0)은 동일하다.

③ 그래프 G는 정점의 집합과 간선의 집합으로 표현한다.

④ 그래프의 탐색에는 깊이 우선 탐색과 너비 우선 탐색이 있다.

정답 1번

간단한 해설

그래프에서 정점의 차수는 **그 정점에 접속한 간선의 수**이다. 그래프 전체의 정점 수와 혼동하면 안 된다.

선지별 해설

① X: 정점의 차수는 그래프 전체의 정점 수가 아니라 **그 정점에 접속한 간선의 수**를 뜻한다. 무방향 그래프에서 루프는 같은 정점에 양 끝이 닿으므로 차수에 2를 더하며, 방향 그래프에서는 들어오는 간선과 나가는 간선을 진입 차수와 진출 차수로 구별한다.

② O: 무방향 그래프의 간선은 두 정점 사이의 연결에 **방향을 부여하지 않는다**. 따라서 두 끝점의 나열 순서를 바꾼 (v0, v1)과 (v1, v0)은 같은 간선을 나타낸다.

③ O: 그래프는 보통 G=(V, E)로 표현한다. **V는 정점 집합, E는 간선 집합**이며, 간선은 정점들 사이의 연결 관계를 나타낸다.

④ O: 깊이 우선 탐색(DFS)은 한 경로를 가능한 깊게 진행한 뒤 되돌아가고, 너비 우선 탐색(BFS)은 **시작 정점에 가까운 층부터** 확장한다. DFS는 스택이나 재귀를, BFS는 큐를 이용하여 구현할 수 있다.

따라서 옳지 않은 설명은 ①이다.

NoteBase

문 15. 트리에 대한 설명으로 옳지 않은 것은?

① 1:n 관계의 비선형 자료구조이다.

② 자식 노드(Child node)는 같은 부모 노드를 가지는 노드를 의미한다.

③ 부모 노드(Parent node)는 임의의 노드 바로 위에 있는 노드를 의미한다.

④ 레벨(Level)은 트리 각층의 번호를 의미한다.

정답 2번

간단한 해설

같은 부모를 가지는 노드들은 형제 노드이다. 자식 노드는 한 부모의 바로 아래에 직접 연결된 노드를 뜻하므로 두 관계를 구분해야 한다.

선지별 해설

① (O) 트리는 노드들을 계층적으로 연결하는 비선형 자료구조이다. 한 부모 노드가 여러 자식 노드를 가질 수 있으므로 **부모와 자식 사이의 1:n 관계**를 표현한다.

② (X) 같은 부모를 가지는 노드들은 **형제 노드**이다. 자식 노드는 특정 부모 노드의 바로 아래에 직접 연결된 노드를 말한다. 예를 들어 A의 자식이 B와 C라면 B와 C는 서로 형제 관계이다.

③ (O) 부모 노드는 해당 노드의 바로 위에서 직접 연결되는 노드이다. 여러 단계 위에 있는 노드는 조상 노드이며, 루트 노드는 부모가 없다는 점을 구분한다.

④ (O) 레벨은 루트에서부터 각 노드가 위치한 층을 나타내는 번호이다. 루트의 레벨을 0 또는 1에서 시작하는 관례가 있지만, 같은 층에 있는 노드들의 레벨이 같다는 의미는 동일하다.

NoteBase

문 16. 정적 테스트에서 비정형 방법에 해당하지 않는 것은?

① 감사

② 개별 검토

③ 모형 검증 기법

④ 소프트웨어 검사

정답 3번

간단한 해설

모형 검증은 수학적으로 표현한 상태와 성질을 검증하는 정형 기법이다. 이 문제의 비정형 방법은 사람의 검토에 기반한 감사·개별 검토·검사와 대비되며, 절차가 엄격한지 여부와는 다른 구분이다.

선지별 해설

① (O) 감사는 소프트웨어 산출물이나 개발 과정이 정해진 기준·절차·계약 등을 따랐는지 검토하는 활동이다. 수학적 모델의 성질을 증명하는 방식이 아니라 기준에 따라 증거를 검토하므로, 문제에서 말하는 비정형 방법에 해당한다.

② (O) 개별 검토는 검토자가 요구사항·설계·코드 등의 산출물을 읽고 결함이나 부적합 사항을 찾는 활동이다. 프로그램을 실행하지 않고 사람의 지식과 판단을 이용한다는 점에서 비정형 정적 검토로 분류한다.

③ (X) **모형 검증(model checking)**은 시스템을 상태와 상태 전이 등의 수학적 모델로 표현하고, 명세한 성질이 그 모델에서 성립하는지 체계적으로 확인하는 정형 검증 기법이다. 따라서 비정형 방법이 아닌 것을 묻는 이 문제의 정답이다.

④ (O) 소프트웨어 검사는 산출물을 여러 검토자가 점검하여 결함을 찾는 검사 활동으로 이해한다. 역할·절차·기록을 엄격히 정한 공식적인 검사일 수 있지만, 그것이 곧 수학적 정형 검증이라는 뜻은 아니다. 이 문제에서는 사람 중심의 검토 방법으로 분류한다.

NoteBase

문 17. 나선형(Spiral) 모델의 단계에 대한 설명으로 옳지 않은 것은?

① 개발 단계에서는 프로토타입을 만든다.

② 계획 및 요구분석 단계에서는 여러 제약 조건의 대안을 고려한 계획을 수립한다.

③ 위험분석 단계에서 요구사항 확정 이후에 계속되는 변경 요구는 위험 요소가 아니다.

④ 사용자 평가 단계에서는 개발된 내용을 사용자가 확인하고, 추가 및 수정 요구사항을 담당자가 반영한다.

정답 3번

간단한 해설

나선형 모델은 **위험을 분석하고 줄이는 반복 개발**을 중시한다. 요구사항 변경은 재작업·일정·비용에 영향을 주므로 위험 요소에서 제외할 수 없다.

선지별 해설

나선형 모델은 계획, 위험 분석, 개발, 평가를 반복하면서 시스템을 구체화하는 위험 중심의 개발 모델이다. 반복을 통해 요구를 보완하지만, 변경이 일으킬 위험까지 없어지는 것은 아니다.

① O: 개발 단계에서는 이번 반복에서 확인할 기능이나 대안을 구현하며 **프로토타입을 만들어 평가에 활용**할 수 있다. 이를 통해 요구를 구체화하고 다음 반복의 개선 사항을 찾는다. 모든 반복의 산출물이 반드시 폐기용 시제품으로 한정되는 것은 아니다.

② O: 계획 및 요구분석에서는 목표와 요구를 파악하고 비용·일정·기술 등의 **제약 조건과 가능한 대안**을 고려한다. 이를 바탕으로 해당 반복에서 수행할 작업과 계획을 정한다.

③ X: 요구사항이 정해진 뒤에도 변경이 계속되면 설계·구현의 재작업, 일정 지연, 비용 증가, 범위 확대 등이 발생할 수 있으므로 **요구 변경은 위험 분석 대상**이다. 변경을 수용할 수 있는 모델이라는 점과 변경이 위험하지 않다는 주장은 다르다.

④ O: 사용자는 개발된 결과를 확인하여 요구와의 차이를 평가하고 추가·수정 요구를 제시한다. 담당자는 이를 검토하여 **다음 계획과 개발 반복에 반영**함으로써 결과를 점진적으로 개선한다.

따라서 옳지 않은 설명은 ③이다.

NoteBase

문 18. 디자인 패턴 중, 구조 패턴에 해당하지 않는 것은?

① Adapter

② Bridge

③ Composite

④ Template method

정답 4번

간단한 해설

Template method는 **알고리즘의 뼈대를 유지하면서 일부 단계의 구현을 바꾸는 행위 패턴**이다. Adapter, Bridge, Composite처럼 클래스나 객체의 구성·연결을 다루는 구조 패턴과 구분된다.

선지별 해설

① (O) Adapter는 **서로 맞지 않는 인터페이스를 변환**하여 기존 클래스나 객체를 함께 사용할 수 있게 한다. 구성 요소 사이의 연결 방식을 다루므로 구조 패턴에 해당한다.

② (O) Bridge는 추상화와 구현을 분리하여 각각 독립적으로 확장할 수 있게 한다. 두 부분을 연결하는 객체 구조를 설계하므로 구조 패턴이다.

③ (O) Composite는 객체들을 트리 구조로 구성해 부분과 전체의 관계를 나타낸다. 개별 객체와 복합 객체를 동일한 인터페이스로 다룰 수 있게 하므로 구조 패턴이다.

④ (X) Template method는 **알고리즘의 전체 흐름을 상위 클래스에 정하고 일부 단계를 하위 클래스에서 구현·재정의**하게 하는 행위 패턴이다. 객체를 결합하는 구조보다 처리 절차와 동작의 변화를 다루므로 구조 패턴이 아니다.

NoteBase

문 19. 기능 점수 산정 방법에 대한 설명으로 옳지 않은 것은?

① 실제 구현 방법과 무관하다.

② 객관적인 요구사항만으로 측정한다.

③ 계획 단계뿐 아니라 분석, 설계, 구현 단계에서도 사용이 가능하다.

④ 측정의 일관성 유지를 위하여 개발 기술, 개발 방법, 품질 수준 등을 고려한다.

정답 4번

간단한 해설

기능 점수는 사용자가 요구하고 제공받는 **기능의 규모**를 측정한다. 구현 기술이나 개발 방법과 독립적으로 측정하므로, 일관성을 위해 개발 기술·개발 방법·품질 수준을 고려한다는 ④가 옳지 않다.

선지별 해설

① O: 기능 점수는 어떤 언어나 기술로 구현하는지보다 **사용자에게 어떤 기능을 제공하는지**를 기준으로 한다. 동일한 기능 요구사항이라면 구현 방법이 달라도 기능 규모를 같은 기준으로 비교할 수 있다.

② O: 사용자 관점에서 확인할 수 있는 기능 요구사항을 근거로 정해진 규칙에 따라 측정한다. 개발자의 주관적인 난이도나 작성한 코드의 길이를 기능 규모의 기준으로 삼는 방식이 아니다.

③ O: 요구사항을 바탕으로 하므로 계획 단계에서 추정에 활용할 수 있고, 분석·설계·구현 단계에서도 기능이 구체화되거나 변경될 때 측정할 수 있다. 즉, 코드 작성이 끝난 뒤에만 사용할 수 있는 방법이 아니다.

④ X: 기능 규모 측정의 일관성을 위해서는 **개발 기술·개발 방법과 독립된 기능적 기준**을 적용한다. 기술이나 품질 요구는 개발 노력·비용에 영향을 줄 수 있지만, 이를 기능 점수로 측정하는 기능 규모 자체와 혼동해서는 안 된다.

NoteBase

문 20. CMMI(Capability Maturity Model Integration)의 소프트웨어 프로세스 성숙도 단계에 대한 설명으로 옳지 않은 것은?

① 관리(Managed) 단계는 기본적인 프로젝트 관리 체계를 수립하는 단계이다.

② 확립(Established) 단계는 소프트웨어공학 원칙을 기반으로 정의된 프로세스를 수행하는 단계이다.

③ 정량적 관리(Quantitative managed) 단계는 정량적으로 프로세스를 측정하고 통제하는 단계이다.

④ 최적화(Optimizing) 단계는 프로세스를 개선하는 활동 단계이다.

정답 2번

간단한 해설

CMMI에서 조직의 표준 프로세스를 정의하여 사용하는 성숙도 3단계의 명칭은 **정의(Defined)**이다.
'확립(Established)'을 CMMI 성숙도 단계 이름으로 제시한 부분이 잘못되었다.

선지별 해설

① (O) 성숙도 2단계인 관리(Managed)는 프로젝트의 계획·수행·점검 등을 관리하는 체계를 갖추는 단계이다. 개별 프로젝트 수준에서 작업을 계획하고 추적·통제한다는 점이 핵심이다.

② (X) 조직 차원의 표준 프로세스를 정하고 각 프로젝트가 이에 맞춘 프로세스를 사용하는 단계는 **성숙도 3단계인 정의(Defined)**이다. 선지는 이 단계의 취지와 관련된 설명을 하면서 명칭을 확립(Established)으로 제시했으므로 틀리다.

③ (O) 성숙도 4단계인 정량적 관리(Quantitatively Managed)는 품질과 프로세스 성과에 대한 정량적 목표를 세우고, 통계적·정량적 기법으로 성과를 측정하고 통제하는 단계이다. 단순히 지표를 수집하는 것을 넘어 변동을 관리한다.

④ (O) 성숙도 5단계인 최적화(Optimizing)는 조직의 목표에 맞추어 프로세스 성과를 지속적으로 개선하는 단계이다. 정량적으로 파악한 성과와 문제 원인을 바탕으로 점진적 개선과 혁신을 추진한다.

NoteBase

문 21. 다음 C프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
main( ){
     int a, b, sum;

     a = 10; b = 20;
     sum = a++ + ++b;
     printf("a = %d\n", a);
     printf("b = %d\n", b);
     printf("sum = %d", sum);
}
```

① a = 10
b = 21
sum = 31
② a = 11
b = 21
sum = 31
③ a = 10
b = 21
sum = 32
④ a = 11
b = 21
sum = 32

정답 2번

간단한 해설

후위 증가 a++는 식에 10을, 전위 증가 ++b는 식에 21을 제공한다. **sum은 31이고, 문장 실행 후 a는 11, b는 21**이다.

NoteBase

계산 과정

초기값: a = 10, b = 20으로 설정한 뒤 sum = a++ + ++b를 계산한다. 두 증가 연산은 서로 다른 변수에 적용되므로 각 항이 식에 제공하는 값과 문장 실행 후 변수 값을 나누어 추적한다.

a++: 후위 증가 연산은 **증가 전의 값 10을 식의 값으로 제공**하고 a 자체는 1 증가시킨다. 이 문장이 끝난 뒤 a는 11이다.

++b: 전위 증가 연산은 **b를 먼저 21로 증가시킨 값을 식에 제공**한다. 따라서 덧셈에는 20이 아니라 21이 사용되고, 문장이 끝난 뒤 b도 21이다.

합과 출력: sum에는 각 항이 제공한 값의 합인 10 + 21 = 31이 저장된다. 이후 출력문에서는 증가가 반영된 a와 b를 읽으므로 다음과 같이 출력한다.
a = 11
b = 21
sum = 31

따라서 실행 결과는 ②와 일치한다.

NoteBase

문 22. 다음 C프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
#include <string.h>
main( ){
     char temp, name[10] = “Alibaba”;
     int   i, n;

     n = strlen(name);
     for(i = 0; i < n/2; i++){
         temp = name[i];
         name[i] = name[n-i-1];
         name[n-i-1] = temp;
     }
     printf(“%s\n”, name);
}
```

① AlibilA

② abababa

③ ababilA

④ aLIBABA

정답 3번

간단한 해설

반복문은 **문자열 양 끝의 문자를 차례로 교환하여 순서를 뒤집는다.** Alibaba를 뒤집으면 대소문자를 그대로 유지한 ababilA가 된다.

NoteBase

계산 과정

문자열 Alibaba의 길이는 **7**이므로 n은 7이다. 종료 문자까지 세는 것이 아니며, 배열의 크기 10도 문자열 길이와 다르다.

n과 2가 정수이므로 n/2는 3이다. 반복문은 i가 0, 1, 2일 때 실행되며, 매번 **name[i]와 name[n−i−1]**을 임시 변수 temp를 이용해 교환한다.

i = 0: 인덱스 0의 A와 6의 a를 교환하여 alibabA가 된다.

i = 1: 인덱스 1의 l과 5의 b를 교환하여 abibalA가 된다.

i = 2: 인덱스 2의 i와 4의 a를 교환하여 ababilA가 된다.

가운데인 인덱스 3의 b는 그대로 두고 양 끝에서 안쪽으로 교환하므로 문자열이 뒤집힌다. 대소문자는 바꾸지 않으며, 최종 출력은 **ababilA**이므로 정답은 ③이다.

NoteBase

문 23. 다음 C프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>

int fib(int k){
        if(k == 1)
              return 0;
        else if(k == 2)
              return 1;
        else
              return fib(k-1)+fib(k-2);
}
main( ){
        int n = 6;
        printf("%d", fib(n));
}
```

① 2

② 3

③ 4

④ 5

정답 4번

간단한 해설

fib는 **첫 두 값 0, 1에서 시작하여 앞의 두 값을 더하는** 재귀 함수이다. fib(1)부터 fib(6)까지의 값은 0, 1, 1, 2, 3, 5이므로 5가 출력된다.

NoteBase

계산 과정

① 초기값 확인: 함수의 종료 조건에 따라 fib(1)은 0이고 fib(2)는 1이다. 첫 번째 값을 1로 놓지 않도록 주의한다.

② 재귀식 적용: k가 3 이상이면 fib(k)는 앞의 두 함수값을 더한다.
fib(3) = fib(2) + fib(1) = 1 + 0 = 1
fib(4) = fib(3) + fib(2) = 1 + 1 = 2
fib(5) = fib(4) + fib(3) = 2 + 1 = 3
fib(6) = fib(5) + fib(4) = 3 + 2 = 5

③ 출력값 확인: main에서 n은 6이므로 printf는 fib(6)의 정수값인 5를 출력한다. 따라서 정답은 ④이다.

NoteBase

문 24. 프로그래밍 언어에 대한 설명으로 옳은 것만을 <보기>에서 모두 고른 것은?

<보기>
ㄱ. C, JAVA, PASCAL은 컴파일러 언어이다.
ㄴ. 어셈블리어는 번역 과정 없이 직접 실행이 가능한 언어이다.
ㄷ. 소스 파일을 실행하기 위해서는 번역-로딩-링킹 순으로 처리해야 한다.
ㄹ. 고급언어는 사람이 이해하기 쉬운 언어이며 저급언어는 컴퓨터가 이해하기 쉬운 언어이다.

① ㄱ, ㄴ
② ㄱ, ㄹ
③ ㄱ, ㄷ, ㄹ
④ ㄴ, ㄷ, ㄹ

정답 2번

간단한 해설

옳은 보기는 **ㄱ, ㄹ**이다. 어셈블리어는 어셈블러의 번역이 필요하고, 일반적인 정적 실행 파일의 처리 순서는 **번역 → 링킹 → 로딩**이므로 ㄴ, ㄷ은 틀리다. 따라서 정답은 ②이다.

보기별 해설

ㄱ. O: C, Java, Pascal은 일반적인 처리 방식에서 컴파일러를 이용하는 언어로 분류한다. C와 Pascal은 소스 코드를 목적 코드 등으로 번역하며, Java는 소스 코드를 바이트코드로 컴파일한 뒤 JVM에서 실행한다. Java가 인터프리터나 JIT 컴파일러도 활용한다는 사실은 소스 코드의 컴파일 과정이 없다는 뜻이 아니다.

ㄴ. X: 어셈블리어는 기계 명령을 사람이 다루기 쉬운 기호로 나타낸 저급언어이므로, **어셈블러를 통해 기계어로 번역**해야 한다. CPU가 직접 실행하는 것은 기계어 명령이다.

ㄷ. X: 일반적인 정적 실행 파일의 처리 순서는 **번역 → 링킹 → 로딩**이다. 번역으로 목적 코드를 만들고, 링킹으로 관련 목적 모듈과 라이브러리를 연결한 뒤, 로딩으로 실행할 프로그램을 메모리에 적재한다. 동적 연결처럼 세부 방식이 다른 경우는 있지만, 제시된 번역 → 로딩 → 링킹을 일반적인 필수 순서로 볼 수는 없다.

ㄹ. O: 고급언어는 사람이 문제 해결 절차를 표현하기 쉽게 추상화되어 있고, 저급언어는 기계의 명령 체계와 구조에 가깝다. 다만 저급언어인 어셈블리어도 번역이 필요하다는 점은 구별해야 한다.

따라서 옳은 보기 ㄱ, ㄹ의 조합은 ②이다.

NoteBase

문 25. 컴퓨터의 구성요소 중, 다음 파이썬 프로그램의 변수가 생성되는 장치로 옳은 것은?

```
name=input('당신의 이름은?')
age=input('당신의 나이는?')
print('성명:', name, '나이:', age)
```

① 레지스터

② 연산 장치

③ 제어 장치

④ 주기억 장치

정답 4번

간단한 해설

실행 중인 프로그램이 입력값을 보관하고 다시 참조하는 기본 저장 공간은 **주기억 장치**이다. name과 age가 가리키는 입력 데이터도 프로그램의 메모리에 유지된다.

개념 구분

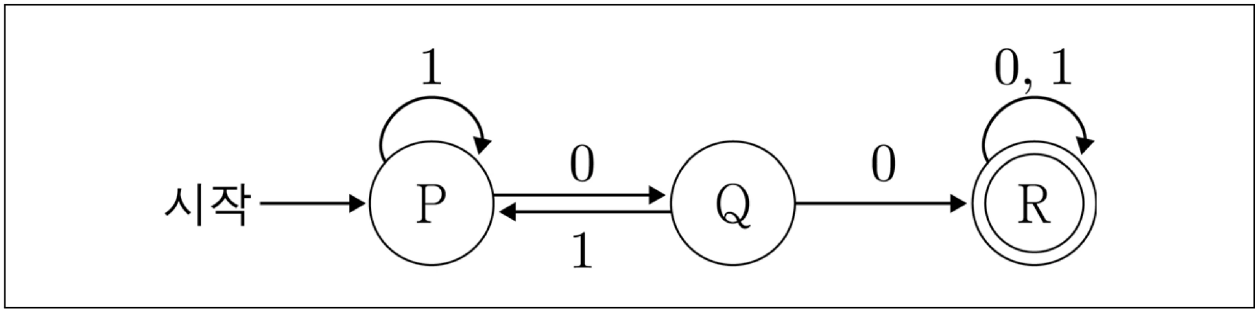
입력문이 실행되면 입력받은 값을 이후 출력문에서 다시 사용할 수 있도록 실행 중인 프로그램의 메모리에 보관한다. 이와 같은 프로그램의 데이터가 저장되는 컴퓨터 구성요소는 **주기억 장치**이므로 ④가 맞다.

파이썬에서는 name과 age라는 이름이 입력값을 담은 객체를 가리키도록 연결된다. Python 3의 input은 문자열을 반환하므로, 제시 코드의 age도 별도의 숫자 변환이 없다면 문자열을 가리킨다. 변수 이름과 객체의 연결을 단순히 CPU 레지스터 하나의 생성으로 해석하면 안 된다.

레지스터는 CPU 내부에서 처리 중인 값이나 주소를 일시적으로 보관하는 고속 저장 장소이고, 연산 장치는 계산을 수행하며, 제어 장치는 명령의 실행 흐름을 제어한다. 실제 처리 과정에서 레지스터가 사용될 수는 있지만, 이 문제에서 묻는 프로그램 변수의 일반적인 저장 장소는 주기억 장치이다.

NoteBase

문 26. 다음 오토마타에서 인식할 수 있는 문자열로 옳지 않은 것은?



① 0001

② 0100

③ 0110

④ 1000

정답 3번

간단한 해설

이 오토마타는 **00을 연속해서 포함하는 문자열**을 인식한다. 0110에는 연속된 00이 없어 최종 상태가 Q가 된다.

NoteBase

선지별 해설

시작 상태는 P이고, 이중 원으로 표시된 R이 수용 상태이다. P에서 0을 읽으면 Q로 가고, Q에서 0을 읽으면 R로 가므로 **연속된 00이 한 번이라도 나타나면**R에 도달한다. P에서 1은 P에 머물고 Q에서 1은 P로 돌아가며, R은 이후 0과 1을 읽어도 계속 R에 머문다.

① O: 0001을 읽으면 P → Q → R → R → R로 이동한다. 처음 두 문자가 00이어서 R에 도달하고 마지막까지 **수용 상태 R**에 있으므로 인식한다.

② O: 0100을 읽으면 P → Q → P → Q → R로 이동한다. 두 번째 문자 1에서 P로 돌아가지만 마지막 두 문자 **00으로 R에 도달**하므로 인식한다.

③ X: 0110을 읽으면 P → Q → P → P → Q로 이동한다. 연속된 00이 없고 입력을 모두 읽었을 때 **수용 상태가 아닌 Q**에 있으므로 인식하지 못한다.

④ O: 1000을 읽으면 P → P → Q → R → R로 이동한다. 첫 1은 P에 머물게 하고 이어지는 00으로 R에 도달하므로 인식한다.

따라서 인식할 수 없는 문자열은 ③ 0110이다.

NoteBase

문 27. <보기>에서 동일하거나 유사한 종류의 프로그램으로 분류된 항목 중 옳지 않은 것은?

<보기>
ㄱ. 워드 프로세서, 스프레드시트, 프레젠테이션
ㄴ. 알파고(AlphaGo), 시리(Siri), ChatGPT
ㄷ. 파이썬, C, JAVA, 스크래치(Scratch)
ㄹ. 유닉스, 리눅스, MS오피스, 안드로이드

① ㄱ
② ㄴ
③ ㄷ
④ ㄹ

정답 4번

간단한 해설

유닉스·리눅스·안드로이드는 운영체제이지만 **MS오피스는 사무용 응용 소프트웨어**이다. 시스템의 실행 환경을 관리하는 소프트웨어와 그 위에서 사용자의 작업을 수행하는 프로그램을 구분한다.

보기별 해설

ㄱ. (O) 워드 프로세서, 스프레드시트, 프레젠테이션 프로그램은 문서 작성, 표 계산, 발표 자료 작성 등 사무 작업을 지원하는 **사무용 응용 소프트웨어**로 묶을 수 있다.

ㄴ. (O) 알파고는 바둑을 두는 인공지능 시스템이고, 시리는 음성 기반의 지능형 비서이며, ChatGPT는 대화형 인공지능 서비스이다. 세부 목적은 다르지만 인공지능을 활용한다는 공통점으로 분류할 수 있다.

ㄷ. (O) 파이썬, C, Java, 스크래치는 모두 프로그램을 작성하는 데 사용하는 **프로그래밍 언어**이다. 스크래치는 블록 기반의 시각적 프로그래밍 방식이라는 차이가 있지만 같은 큰 범주에 속한다.

ㄹ. (X) 유닉스, 리눅스, 안드로이드는 운영체제 범주에 속하지만, **MS오피스는 사무용 응용 소프트웨어 모음**이다. 운영체제는 하드웨어 자원을 관리하고 프로그램 실행 환경을 제공하며, MS오피스는 그 환경 위에서 문서 등의 작업을 수행한다.

같은 종류로 묶기 부적절한 항목은 **ㄹ**이므로 정답은 ④이다.

NoteBase

문 28. 프로세스에 대한 설명으로 옳은 것만을 <보기>에서 모두 고른 것은?

<보기>
ㄱ. 프로세스는 메모리에 올라간 프로그램을 의미한다.
ㄴ. 실행 상태의 프로세스는 CPU에 할당되어 처리되고 있는 프로세스를 의미한다.
ㄷ. 대기 상태의 프로세스는 CPU에 입출력 자원을 요청한 후 기다리는 상태의 프로세스를 의미한다.
ㄹ. 준비 상태의 프로세스는 사용자의 선택 명령을 기다리는 보조기억장치에 있는 프로그램을 의미한다.

① ㄱ, ㄴ
② ㄴ, ㄷ
③ ㄱ, ㄴ, ㄷ
④ ㄱ, ㄷ, ㄹ

정답 3번

간단한 해설

프로세스의 **준비 상태는 CPU 할당 대기**, 실행 상태는 CPU 사용, 대기 상태는 입출력 완료 등 사건 발생 대기이다. 준비 상태를 보조기억장치의 프로그램이 사용자 명령을 기다리는 상태로 설명한 ㄹ은 옳지 않다.

보기별 해설

ㄱ. O: 프로세스는 실행을 위해 메모리에 적재되어 운영체제의 관리 대상이 된 프로그램의 실행 단위이다. 단순히 보조기억장치에 저장된 프로그램 파일과 구별되며, 코드뿐 아니라 현재 실행 상태와 관련 자원도 포함한다.

ㄴ. O: 실행 상태는 프로세스가 CPU를 할당받아 실제로 명령어를 수행하는 상태이다. CPU를 받을 준비만 마친 준비 상태와 구별된다.

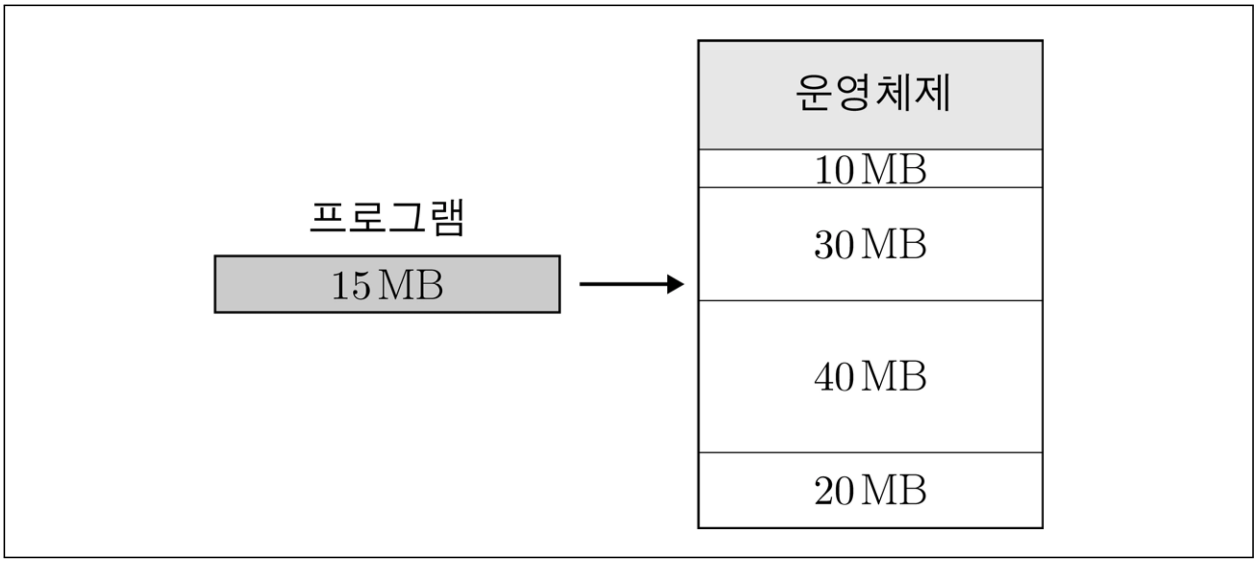
ㄷ. O: 대기 상태는 입출력 요청 후 그 완료 등 특정 사건을 기다려, 당장 CPU를 받아도 다음 작업을 진행할 수 없는 상태이다. 지문의 입출력 자원 요청은 운영체제를 통한 입출력 요청으로 이해하며, 핵심은 **입출력 완료를 기다린다는 점**이다.

ㄹ. X: 준비 상태는 실행에 필요한 준비를 마친 프로세스가 CPU 할당을 기다리는 상태이다. 보조기억장치에 저장된 프로그램이 사용자의 실행 명령을 기다리는 상태를 뜻하지 않는다.

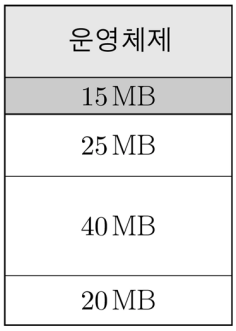
옳은 보기는 ㄱ, ㄴ, ㄷ이므로 정답은 ③이다.

NoteBase

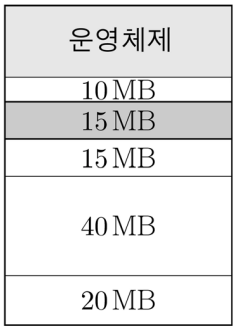
문 29. 다음과 같이 분할된 빈 메모리에 프로그램을 할당할 때 최악 적합(Worst Fit)에 해당하는 것은?



①



②

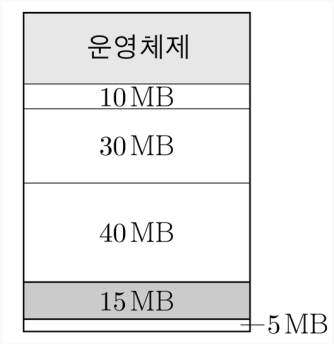


③



NoteBase

④



정답 3번

간단한 해설

최악 적합은 15 MB를 넣을 수 있는 영역 중 **가장 큰 40 MB 영역**을 선택한다. 그 안에 15 MB를 할당하고 25 MB를 남긴 배치가 해당한다.

계산 과정

최악 적합은 프로그램을 넣을 수 있는 빈 영역 중 **가장 큰 영역**을 선택한다. 프로그램은 15 MB이므로 10 MB 영역에는 들어갈 수 없고, 30 MB·40 MB·20 MB 영역이 후보가 된다.

후보 중 가장 큰 영역은 40 MB이다. 여기에 15 MB를 할당하면 남는 공간은 $40 - 15 = 25$MB이다. 따라서 위에서부터 10 MB, 30 MB, 할당된 15 MB와 남은 25 MB, 20 MB가 배치되는 ③이 맞다.

②처럼 30 MB를 선택하면 위에서부터 탐색했을 때 처음 들어갈 수 있는 영역을 고르는 최초 적합에 해당한다.

④처럼 20 MB를 선택하면 들어갈 수 있는 영역 중 가장 작은 것을 골라 5 MB를 남기는 최적 적합에 해당한다.

①은 제시된 분할 영역을 그대로 두고 가장 큰 40 MB 영역을 선택한 배치가 아니다.

NoteBase

문 30. 운영체제에 대한 설명으로 옳은 것만을 <보기>에서 모두 고른 것은?

<보기>
ㄱ. 일괄처리(Batch Processing) 시스템은 여러 작업들을 모아 순서대로 자동 처리할 수 있는 시스템을 의미한다.
ㄴ. 시분할(Time Sharing) 시스템은 CPU 운영 시간을 잘게 쪼개어 여러 사용자가 동시에 사용할 수 있도록 지원하는 시스템을 의미한다.
ㄷ. 실시간(Real Time) 시스템은 정해진 짧은 시간 내에 사용자 서비스를 지원하는 시스템을 의미하며 제한 시간은 CPU 성능을 고려하여 관리자가 정한다.

① ㄱ
② ㄱ, ㄴ
③ ㄴ, ㄷ
④ ㄱ, ㄴ, ㄷ

정답 2번

간단한 해설

일괄처리는 모은 작업을 자동으로 처리하고 시분할은 CPU 시간을 나누어 사용자에게 배분한다. 실시간의 기준은 **응용이 요구하는 시간 제약**이므로 관리자 편의의 제한 시간과 구별해야 한다.

보기별 해설

ㄱ. O: 일괄처리 시스템은 처리할 작업을 모아 정해진 순서나 스케줄에 따라 **사용자의 지속적인 개입 없이 자동 처리**한다. 급여 계산이나 정기 집계처럼 즉각적인 대화형 응답이 필요하지 않은 작업에 적합하다.

ㄴ. O: 시분할 시스템은 CPU 시간을 짧은 시간 조각으로 나누어 여러 작업에 번갈아 배분한다. 빠른 전환으로 **여러 사용자가 동시에 이용하는 것처럼**지원하며, 각 사용자에게 대화형 응답을 제공한다.

ㄷ. X: 실시간 시스템의 핵심은 단순히 빠른 실행이 아니라 **응용이나 제어 대상이 요구하는 마감 시간의 충족**이다. 제한 시간을 CPU 성능만을 기준으로 관리자가 임의로 정하는 것이 아니라, 외부 사건과 서비스의 시간 요구에 맞추어 정하고 시스템이 이를 만족하도록 설계해야 한다.

옳은 보기는 ㄱ과 ㄴ이므로 ②와 일치한다.

NoteBase

문 31. VPN(Virtual Private Network)에 관한 내용으로 옳은 것을 <보기>에서 고른 것은?

<보기>
ㄱ. 물리적 사설 네트워크 보안 기술
ㄴ. 터널링으로 제공하는 보안 서비스
ㄷ. TCP/IP 세션 계층에서 제공하는 보안 서비스
ㄹ. 외부에서 제한된 내부 서비스에 접근하기 위해 네트워크 트래픽을 암호화하여 제공하는 보안 서비스

① ㄱ, ㄷ
② ㄱ, ㄹ
③ ㄴ, ㄷ
④ ㄴ, ㄹ

정답 4번

간단한 해설

VPN은 공용망 위에 **가상의 사설 통신 경로**를 구성한다. 보안 VPN은 터널링에 암호화·인증을 결합하여 외부와 내부망 사이의 통신을 보호하며, 물리적 사설망이나 특정 세션 계층의 서비스로 한정되지 않는다.

보기별 해설

ㄱ. (X) VPN은 **가상 사설망**이다. 공용 네트워크 위에 논리적인 사설 통신 경로를 구성하므로, 별도의 물리적 전용 사설망을 구축하는 기술이라는 설명은 틀리다.

ㄴ. (O) VPN은 **터널링**으로 원래의 데이터나 패킷을 다른 프로토콜 안에 담아 지정된 끝점 사이로 전달한다. 보안 VPN은 여기에 암호화와 인증 등을 결합하여 통신을 보호한다. 터널링 자체와 암호화는 구별되는 기능이다.

ㄷ. (X) 일반적인 TCP/IP 계층 모델에는 독립된 세션 계층이 없다. 또한 VPN의 구현 계층은 방식에 따라 다르다. 예를 들어 IPsec VPN은 인터넷 계층의 IP 통신을 보호하므로 VPN을 TCP/IP 세션 계층 서비스라고 규정할 수 없다.

ㄹ. (O) 원격 접속용 보안 VPN은 외부 사용자가 내부 서비스에 접근할 때 인증과 암호화된 통신 경로를 제공한다. 이를 통해 공용망을 지나는 네트워크 트래픽의 내용을 보호할 수 있다.

옳은 보기는 **ㄴ, ㄹ**이므로 정답은 ④이다.

NoteBase

문 32. 사물 인터넷(IoT, Internet of Things)의 보안 위협에 대한 설명으로 옳지 않은 것은?

① 저사양 장치는 구조가 단순하여 보안이 안전하다.

② 이기종 간 무선 네트워크 연결 서비스는 일정한 수준의 보안 유지가 어렵다.

③ 공개 플랫폼을 통한 기기-서비스 간 허위 데이터 전송 및 오작동 등의 공격이 가능하다.

④ 클라우드 서비스를 이용하여 감염 PC를 생산하고, 냉장고 및 청소 로봇 등에서 악성 코드 공격을 받을 수 있다.

정답 1번

간단한 해설

IoT 기기는 자원과 관리 기능이 제한되어 보안 대책 적용에 제약이 생길 수 있다. 따라서 **저사양·단순 구조만으로 보안이 안전하다고 단정할 수 없다.**

선지별 해설

① X: 저사양 장치는 연산 성능·메모리·전력의 제약 때문에 강한 보안 기능이나 지속적인 업데이트를 제공하기 어려울 수 있다. **구조가 단순하다는 이유만으로 안전성이 보장되지는 않는다.** 인증, 접근 제어, 안전한 업데이트 등의 보호 대책이 필요하다.

② O: 제조사, 운영체제, 통신 방식과 보안 기능이 서로 다른 기기들이 연결되면 공통된 인증·접근 제어·업데이트 수준을 유지하기 어려워진다. 보안이 약한 기기가 다른 시스템으로 공격이 확산되는 통로가 될 수도 있다.

③ O: 기기와 서비스 사이의 인증이나 데이터 무결성 검증이 부족하면 허위 데이터나 변조된 명령이 전달되어 오작동을 유발할 수 있다. 공개 플랫폼이라는 사실 자체가 취약성을 뜻하는 것은 아니지만, 외부 연동 지점에 대한 보호가 미흡하면 공격 경로가 될 수 있다.

④ O: 공격자는 클라우드의 저장·전달 기능 등을 악성코드 유포에 악용하여 PC나 연결된 기기를 감염시킬 수 있다. 냉장고·청소 로봇 같은 IoT 기기도 소프트웨어와 네트워크 기능을 가지므로 악성코드의 공격 대상이나 공격에 동원되는 기기가 될 수 있다.

NoteBase

문 33. 정보보안의 3대 요구사항에 해당하지 않는 것은?

① 안전성(Safety)

② 무결성(Integrity)

③ 가용성(Availability)

④ 기밀성(Confidentiality)

정답 1번

간단한 해설

정보보안의 3대 요구사항은 **기밀성·무결성·가용성(CIA)**이다. 안전성(Safety)은 이 세 항목의 명칭에 포함되지 않는다.

선지별 해설

① X: 안전성(Safety)은 사고나 위해를 방지하는 관점의 성질이며, 정보보안의 기본 **CIA 3요소에 포함되는 명칭은 아니다**. 안전이 중요하지 않다는 뜻이 아니라 기밀성·무결성·가용성이라는 분류와 구별해야 한다.

② O: 무결성(Integrity)은 정보가 **허가 없이 변경되거나 훼손되지 않도록** 보호하는 성질이다. 데이터의 정확성과 완전성을 유지하고 부적절한 변조를 막는 요구사항이다.

③ O: 가용성(Availability)은 권한 있는 사용자가 **필요한 때에 정보와 서비스를 이용**할 수 있도록 하는 성질이다. 장애나 서비스 거부 등으로 정상 이용이 막히지 않도록 보호한다.

④ O: 기밀성(Confidentiality)은 정보를 **권한 없는 대상에게 공개하지 않는** 성질이다. 접근 통제나 암호화 등으로 정보의 무단 열람·유출을 방지한다.

따라서 정보보안의 3대 요구사항에 해당하지 않는 것은 ① 안전성이다.

NoteBase

문 34. 해시 함수의 특징에 대한 설명으로 옳은 것은?

① 양방향성을 가지고 있다.

② 평문의 길이에 따라 해시값의 길이도 달라진다.

③ 평문의 내용이 조금만 달라져도 해시값이 달라진다.

④ 해시는 정보 위변조를 막기 위해 사용되는 암호와 달리 정보를 숨기기 위해 사용된다.

정답 3번

간단한 해설

암호학적 해시 함수는 입력이 조금만 바뀌어도 출력이 크게 달라지는 **눈사태 효과**를 갖도록 설계된다. 따라서 입력 변화에 대한 민감성을 설명한 ③이 옳다.

선지별 해설

① X: 암호학적 해시 함수는 **일방향성**을 갖는다. 입력으로 해시값을 계산하는 것은 쉽지만, 해시값만으로 그에 해당하는 원문을 찾는 것은 계산상 어렵도록 설계된다. 암호문을 복호화하듯 원문으로 되돌리는 양방향 변환이 아니다.

② X: 이 문항에서 다루는 고정 길이 해시 함수는 입력 길이에 관계없이 정해진 길이의 해시값을 출력한다. 예를 들어 SHA-256의 출력 길이는 256비트이다.

③ O: 입력의 작은 변화가 출력의 여러 비트에 영향을 미치는 **눈사태 효과**를 설명한다. 따라서 내용이 조금 달라진 입력은 보통 크게 다른 해시값을 갖는다. 다만 출력의 길이가 유한하므로 서로 다른 입력이 같은 값을 갖는 충돌 가능성 자체가 없어지는 것은 아니다.

④ X: 역할을 뒤바꾸어 설명했다. 암호화는 주로 원문을 숨겨 **기밀성**을 보호하고, 해시는 데이터의 변경 여부를 확인하는 **무결성 검증** 등에 활용된다. 해시값 자체도 공격자가 바꿀 수 있다면 단순 비교만으로 안전하지 않으므로, 신뢰할 수 있는 기준 해시나 인증 수단과 함께 사용해야 한다.

NoteBase

문 35. 무선랜 공격에 대한 대응 기법으로 옳지 않은 것은?

① MAC 주소 필터링 인증

② 동적 WEP 인증 및 EAP 인증

③ 키로거(Key logger)를 통한 접속 제한

④ SSID(Service Set Identifier) 설정을 통한 접속 제한

정답 3번

간단한 해설

키로거는 키보드 입력을 기록하는 도구이지 무선랜 접속을 제한하는 인증 기법이 아니다. 나머지 항목의 보안상 한계와 별개로, 키로거는 무선랜 접속 통제라는 기능 자체에 해당하지 않는다.

선지별 해설

① (O) MAC 주소 필터링은 접속을 허용할 단말의 MAC 주소를 등록하여 다른 주소의 접속을 제한하는 방식이다. 다만 **MAC 주소는 위조될 수 있으므로 단독으로 강한 인증을 제공하지는 못한다**.

② (O) EAP는 무선랜의 사용자·단말 인증에 활용하는 프레임워크이며, 동적 WEP는 인증 과정과 연계해 WEP 키를 동적으로 배포하는 과거의 보안 방식이다. 따라서 무선랜 보안 기술의 종류에는 해당하지만, **WEP는 취약한 구형 방식이므로 현재의 안전한 구성으로 권장되는 것은 아니다**.

③ (X) 키로거는 키보드 입력을 기록하는 프로그램 또는 장치이다. 악용되면 비밀번호 등의 입력 정보를 빼낼 수 있지만, **무선랜 접속 권한을 인증하거나 제한하는 보안 기법은 아니다**.

④ (O) 이 문항에서는 SSID 설정을 접속할 무선망을 지정·관리하는 항목으로 분류한다. 다만 SSID는 네트워크 식별자일 뿐 비밀번호가 아니며, 이름을 바꾸거나 숨기는 것만으로 공격자의 접속을 막을 수는 없다. 실질적인 접속 보호에는 별도의 인증과 암호화가 필요하다.

NoteBase

문 36. 시스템에서 사용하는 보안 기법에 대한 설명으로 옳지 않은 것은?

① 침입 방지 시스템은 방화벽 앞에 설치하여 효율적으로 패킷을 걸러주도록 한다.

② DRM(Digital Right Management)은 문서 보안에 초점을 둔 기술이다.

③ SSO(Single Sign On)는 한 시스템의 인증으로 여러 시스템에 인증이 가능하다.

④ 침입 탐지 시스템은 설치 위치에 따라 호스트 기반 침입 탐지 시스템과 네트워크 기반 침입 탐지 시스템으로 구분된다.

정답 1번

간단한 해설

IPS의 검사 부담을 줄이는 통상적인 구성에서는 방화벽이 불필요한 트래픽을 먼저 차단하고, **그 뒤의 IPS가 통과한 트래픽을 정밀 검사**한다. 따라서 방화벽 앞을 일반적인 설치 위치로 설명한 부분이 부적절하다.

선지별 해설

① X: 독립된 네트워크형 침입 방지 시스템(IPS)의 검사 부담을 줄이는 통상적인 구성은 외부망에서 들어오는 트래픽이 **방화벽을 먼저 통과한 뒤 IPS에서 정밀 검사**를 받도록 하는 것이다. 방화벽이 접근정책에 맞지 않는 트래픽을 먼저 차단하면 IPS가 분석할 양을 줄일 수 있으므로, 이 기본 배치에서는 방화벽의 뒤쪽인 내부망 측에 설치한다. 다만 방화벽 자체를 보호하거나 그 부하를 줄이기 위한 앞쪽 배치도 가능하다. 따라서 앞쪽 설치가 기술적으로 불가능한 것은 아니며, 이 문항은 IPS 검사 효율을 고려한 일반적인 배치를 기준으로 판단한다.

② O: 기업의 문서 보안에서 DRM은 문서를 암호화하고 사용자의 열람·편집·복사·인쇄 등 **사용 권한을 통제**하여 무단 이용과 유출을 줄이는 기술이다. 파일이 전달된 뒤에도 허용된 사용만 하도록 보호하는 데 초점을 둔다. DRM의 적용 대상이 문서에만 한정되는 것은 아니며, 음악·영상 등 디지털 콘텐츠의 권리 보호에도 사용된다. 통상적인 영문 명칭은 Digital Rights Management이다.

③ O: SSO는 사용자가 한 번 인증한 결과를 연계된 여러 시스템이 신뢰하여 **서비스마다 다시 로그인하는 절차를 줄이는** 통합인증 방식이다. 인증 연계가 구성된 시스템에 적용되며, 한 번 인증했다고 모든 시스템의 모든 자원에 대한 권한까지 자동으로 생기는 것은 아니다.

④ O: 호스트 기반 IDS는 개별 서버나 단말에 설치하여 로그, 파일 변경, 프로세스 등 해당 호스트의 활동을 감시한다. 네트워크 기반 IDS는 네트워크 구간에 배치하여 패킷과 통신 흐름을 분석한다. 따라서 **감시 대상과 배치 지점**에 따라 호스트 기반과 네트워크 기반을 구분할 수 있다.

NoteBase

문 37. 가상화(Virtualization)에 대한 설명으로 옳지 않은 것은?

① 애플리케이션 가상화에서 사용자 PC는 입출력 장치로 사용된다.

② 가상화를 통해 전산 자원들의 이용률(Utilization)을 높일 수 있다.

③ 클라이언트 가상화는 PC 내에 사용자별로 구분되는 물리 공간을 제공한다.

④ 서버 가상화는 한 대의 물리적 고성능 서버를 여러 대의 가상 서버로 분리하는 것이다.

정답 3번

간단한 해설

가상화는 물리 자원을 추상화하여 **독립적인 논리적 실행 환경**을 제공한다. 사용자별로 물리 공간을 만든다는 설명과 구별해야 한다.

선지별 해설

① O: 애플리케이션을 서버에서 실행하여 제공하는 방식에서는 사용자 PC가 키보드·마우스 입력을 보내고 화면 결과를 표시하는 **입출력 단말 역할**을 한다. 이 설명은 원격 실행형 애플리케이션 가상화에 해당하며, 모든 애플리케이션 가상화 방식이 서버에서만 실행된다는 뜻은 아니다.

② O: 가상화는 여러 실행 환경이 물리적 CPU·메모리 등의 자원을 나누어 쓰도록 한다. 개별 서버에 흩어진 작업을 통합하면 **유휴 자원을 활용하여 이용률**을 높일 수 있다.

③ X: 클라이언트 가상화가 제공하는 구분은 사용자별로 독립된 **논리적·가상 실행 환경**이다. 사용자마다 별개의 물리 공간을 새로 만드는 것이 아니며, 공유되는 물리 자원 위에서 환경을 분리·관리한다.

④ O: 서버 가상화는 하나의 물리 서버에서 **여러 가상 서버를 독립적으로 실행**할 수 있게 한다. 가상화 계층이 CPU·메모리·입출력 자원 등을 배분하므로 각 가상 서버는 별도의 서버처럼 동작할 수 있다.

따라서 옳지 않은 설명은 ③이다.

NoteBase

문 38. 네트워크 트래픽 유형은 탄력적 트래픽과 비탄력적 트래픽으로 구분할 수 있다. 트래픽의 유형이 나머지와 다른 서비스는?

① 화상 회의

② 전자 우편

③ 인터넷 전화

④ 인터넷 뱅킹

정답 2번

출제 포인트

네트워크 지연과 전송률 변화에 적응할 수 있는지를 기준으로 탄력적 트래픽과 비탄력적 트래픽을 구분한다.

핵심 개념

탄력적 트래픽은 전송이 늦어지거나 전송률이 낮아져도 기다리면서 서비스 목적을 달성할 수 있는 트래픽이다. 전자 우편과 파일 전송이 대표적이다.
비탄력적 트래픽은 지연이나 전송률 변화에 쉽게 적응하지 못하는 트래픽이다. 실시간 음성·영상처럼 제때 전달되는 것이 중요하거나 신속한 응답이 필요한 서비스가 해당한다.

선지별 해설

O/X는 '나머지와 다른 유형의 서비스인가'에 대한 판단이다.
① X — 화상 회의는 실시간으로 영상과 음성을 주고받으므로 지연과 지연 변동이 커지면 대화가 어려워진다. 비탄력적 트래픽이므로 다른 유형이라는 판단은 틀리다. 올바른 분류는 ③·④와 같은 비탄력적 트래픽이다.
② O — 전자 우편은 보내는 사람과 받는 사람이 동시에 접속하여 즉시 응답할 필요가 없다. 전송이 늦어져도 메일을 전달할 수 있으므로 탄력적 트래픽이며, 나머지 서비스와 유형이 다르다.
③ X — 인터넷 전화는 실시간 음성 대화이므로 지연이 커지면 서로의 말이 엇갈리고 대화 품질이 낮아진다. 다른 유형이 아니라 ①·④와 같은 비탄력적 트래픽으로 분류한다.
④ X — 이 문항의 인터넷 뱅킹은 거래 요청에 대한 신속한 응답이 필요한 대화형 금융 서비스로 보아 비탄력적 트래픽으로 분류한다. 따라서 다른 유형이라는 판단은 틀리고, ①·③과 같은 유형이다.

오답 포인트

④ 인터넷 뱅킹을 영상이나 음성이 없다는 이유만으로 탄력적 트래픽으로 고르기 쉽다. 구분 기준은 멀티미디어 사용 여부가 아니라 서비스의 시간 요구사항이다. 이 문항에서는 인터넷 뱅킹의 대화형 거래 응답을 강조한다. 또한 데이터의 정확성이 중요하다는 사실만으로 비탄력적 트래픽이라고 판단하지 않는다.

NoteBase

다시 나오면 이렇게 판단

전송이 늦어져도 기다린 뒤 목적을 달성할 수 있으면 탄력적 트래픽으로 판단한다. 실시간 대화나 지연에 민감한 거래 응답처럼 제때 전달되는 것이 핵심이면 비탄력적 트래픽으로 판단한다. 서비스 이름만으로 일반화하지 않고 문제에서 전제한 사용 방식과 시간 요구사항을 함께 본다.

암기 포인트

전자 우편은 기다릴 수 있는 탄력적 트래픽이다. 화상 회의·인터넷 전화는 실시간 전달이 필요한 비탄력적 트래픽이며, 이 문항의 인터넷 뱅킹도 신속한 거래 응답을 기준으로 비탄력적 트래픽에 포함한다.

NoteBase

문 39. 전기통신 분야의 국제 표준을 담당하는 유엔 산하 기구는?

① ISO

② ITU

③ ANSI

④ IEEE

정답 2번

간단한 해설

ITU(국제전기통신연합)는 정보통신 분야의 유엔 전문기구이다. 전기통신 국제표준화는 ITU-T 부문이 담당한다는 점을 연결하면 구분할 수 있다.

핵심 풀이

ITU는 **국제전기통신연합**으로, 정보통신 분야를 담당하는 유엔 전문기구이다. 특히 ITU-T 부문이 전기통신 관련 국제 표준화 권고를 개발한다.

문제의 핵심 조건은 단순히 표준을 만드는 기관이라는 점이 아니라 **전기통신 분야이면서 유엔 산하 기구**라는 점이다. 두 조건을 함께 충족하는 것은 ITU이므로 정답은 ②이다.

ISO는 여러 산업 분야의 국제표준을 다루는 국제표준화기구이고, ANSI는 미국의 국가표준 체계를 조정하는 기관이다. IEEE는 전기·전자·컴퓨터 분야의 전문단체로 IEEE 802 계열 등의 표준을 개발하지만 유엔 전문기구는 아니다.

NoteBase

문 40. VoIP QoS에 영향을 주는 파라미터에 대한 설명으로 옳지 않은 것은?

① 지연(Delay)은 데이터를 전송하는 데 지연되는 시간을 말한다.

② 지터(Jitter)는 패킷이 도착하는 시간 간격이 규칙적인 경우를 말한다.

③ 패킷 손실(Packet lost)이 발생한 경우 서비스에 따라 패킷 손실 은폐 방법을 사용하기도 한다.

④ 무음 억제(Silence suppression)는 패킷의 수를 줄이기 위해 무음이 지속되는 기간 동안 음성 패킷 생성을 억제한다.

정답 2번

간단한 해설

지터는 패킷별 전송 지연이 달라져 **도착 간격이 불규칙해지는 변동**을 뜻한다. 규칙적인 도착 간격을 뜻한다는 ②가 옳지 않다.

선지별 해설

① O: 지연은 송신 측에서 보낸 데이터가 수신 측에 전달되기까지 걸리는 시간이다. VoIP에서는 음성의 부호화·패킷화, 네트워크 전송, 버퍼링 등에서 발생하는 지연이 커지면 자연스러운 대화가 어려워진다.

② X: 지터는 **패킷 전송 지연의 변동**이다. 일정한 간격으로 보낸 음성 패킷도 네트워크 상황에 따라 불규칙한 간격으로 도착할 수 있다. 수신 측의 지터 버퍼는 도착 시각의 차이를 흡수하여 일정한 속도로 재생하도록 돕지만, 버퍼링에 따른 지연이 추가될 수 있다.

③ O: 패킷 손실 은폐는 손실된 음성 구간을 앞뒤 음성 정보 등으로 추정하거나 대체하여 끊김을 덜 느끼도록 하는 방법이다. 이는 손실 패킷을 반드시 원래대로 복구하는 것이 아니라 **손실이 들리는 영향을 완화**하는 것이다.

④ O: 무음 억제는 음성이 없는 구간에서 불필요한 음성 패킷 생성을 줄여 대역폭을 절약한다. 구현에 따라 수신 측에서 자연스러운 배경 잡음을 재생하도록 별도의 잡음 관련 정보를 보낼 수도 있다.

NoteBase