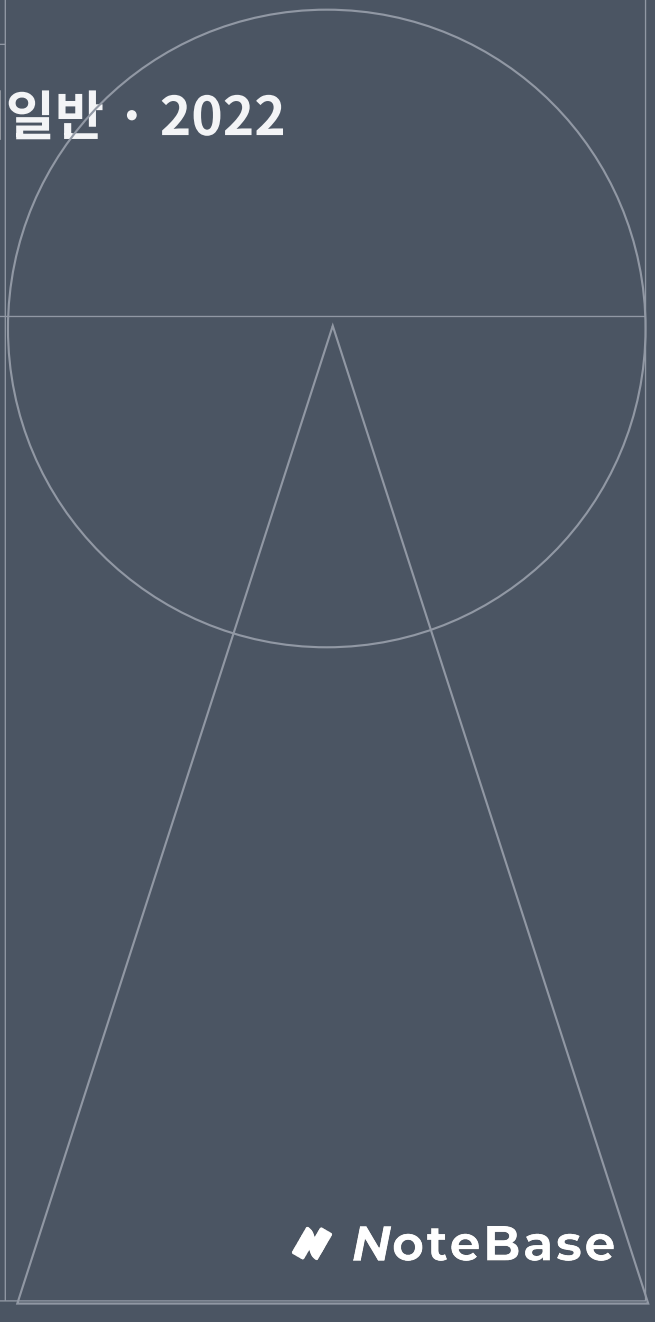
기출OX 해설집

공무원/국가직/9급 · 컴퓨터일반 · 2022



NoteBase

기출OX 해설집

공무원/국가직/9급 · 컴퓨터일반 · 2022

공무원/국가직/9급 · 컴퓨터일반 · 2022

문 1. 대표적인 반도체 메모리인 DRAM과 SRAM에 대한 설명으로 옳지 않은 것은?

① DRAM은 휘발성이지만 SRAM은 비휘발성이어서 전원이 공급되지 않아도 기억을 유지할 수 있다.

② DRAM은 축전기(Capacitor)의 충전상태로 비트를 저장한다.

③ SRAM은 주로 캐시 메모리로 사용된다.

④ 일반적으로 SRAM의 접근속도가 DRAM보다 빠르다.

정답 1번

간단한 해설

SRAM도 DRAM과 마찬가지로 **휘발성**이므로, 전원이 끊기면 저장 내용을 잃는다.

선지별 해설

① X: DRAM과 SRAM은 **모두 휘발성 메모리**이므로 전원 공급이 중단되면 저장 내용이 사라진다. SRAM은 전원이 공급되는 동안 쌍안정 회로의 상태로 비트를 유지한다. '정적(Static)'이라는 말은 DRAM과 달리 주기적인 리프레시가 필요 없다는 뜻이며, 전원 없이도 내용을 보존한다는 뜻이 아니다.

② O: DRAM은 **축전기에 저장된 전하의 유무**로 0과 1을 구분한다. 시간이 지나면 누설 전류로 전하가 줄어들기 때문에, 전원이 공급되는 동안에도 저장 내용을 유지하려면 주기적으로 전하를 보충하는 리프레시가 필요하다.

③ O: 캐시 메모리는 CPU가 자주 사용하는 데이터와 명령어를 보관하여 CPU와 주기억장치 사이의 속도 차이를 줄인다. SRAM은 **접근속도가 빠르기 때문에** 캐시 메모리에 주로 사용된다.

④ O: 일반적으로 SRAM은 DRAM보다 접근속도가 빠르다. SRAM은 쌍안정 회로의 상태를 이용하여 데이터를 빠르게 읽고 쓸 수 있다. 반면 DRAM은 **축전기의 미세한 전하를 감지하고 복원**하는 과정이 필요하다.

NoteBase

문 2. 정렬 알고리즘 중 최악의 경우를 가정할 때 시간복잡도가 다른 것은?

① 삽입 정렬(Insertion sort)

② 쉘 정렬(Shell sort)

③ 버블 정렬(Bubble sort)

④ 힙 정렬(Heap sort)

정답 4번

간단한 해설

힙 정렬은 **완전이진트리의 높이에 비례하는 작업**으로 힙을 복구하므로 최악에도 $O(n\log n)$ 시간이 걸린다. 기본 쉘 정렬을 기준으로 나머지 정렬의 최악 시간복잡도는 $O(n^2)$이다.

선지별 해설

① 삽입 정렬: 최악 시간복잡도는 $O(n^2)$이다. 오름차순으로 정렬할 때 **자료가 역순으로 놓여 있으면**, 새 원소를 삽입할 때마다 앞의 원소들을 모두 한 칸씩 이동해야 한다. 이러한 작업이 원소마다 누적되어 연산량이 자료 수의 제곱에 비례한다.

② 쉘 정렬: 떨어진 원소들끼리 삽입 정렬을 수행하고, 간격을 줄여 마지막에는 간격 1로 정렬한다. **간격을 절반씩 줄이는 기본 방식**의 최악 시간복잡도는 $O(n^2)$이다. 쉘 정렬의 복잡도는 간격 수열에 따라 달라지므로 모든 방식이 같은 복잡도를 갖는 것은 아니다. 이 문항은 기본 방식의 복잡도를 기준으로 비교한다.

③ 버블 정렬: 최악 시간복잡도는 $O(n^2)$이다. 인접한 두 원소를 비교하여 순서가 어긋나면 교환한다. 역순 자료에서는 한 차례 순회로 가장 큰 원소 하나의 위치를 확정한 뒤, **남은 구간을 다시 순회**하는 작업을 반복하므로 비교·교환 횟수가 자료 수의 제곱에 비례한다.

④ 힙 정렬: 최악 시간복잡도는 $O(n\log n)$이다. 오름차순 정렬에서는 최대 힙의 루트에 있는 최댓값을 배열 뒤쪽으로 보내고, 남은 원소들의 힙을 복구한다. 힙은 완전이진트리이므로 높이가 $O(\log n)$이고, 한 번의 복구는 **루트에서 아래로 내려가는 경로**만 따라가면 된다. 이 작업을 원소 수에 비례하는 횟수만큼 반복하므로 전체 시간복잡도는 $O(n\log n)$이다.

따라서 기본 쉘 정렬을 전제로 ①·②·③은 최악에 $O(n^2)$, ④ 힙 정렬은 $O(n\log n)$으로 서로 다르다.

NoteBase

문 3. 기계 학습에서 지도 학습과 비지도 학습에 대한 설명으로 옳은 것은?

① 지도 학습의 대표적인 기법에는 군집화가 있다.

② 비지도 학습의 기법에는 분류와 회귀분석 등이 있다.

③ 지도 학습은 학습 알고리즘이 수행한 행동에 대해 보상을 받는 학습 방식이다.

④ 비지도 학습은 정답이 없는 데이터를 보고 유용한 패턴을 추출하는 학습 방식이다.

정답 4번

간단한 해설

비지도 학습은 **목표값이나 범주 표시가 없는 데이터**에서 데이터 간 유사성이나 숨은 구조를 찾아 유용한 패턴을 추출한다.

선지별 해설

① X: 지도 학습은 입력 데이터와 그에 대응하는 목표값이나 범주 표시인 레이블을 함께 제공하여 둘 사이의 관계를 학습한다. 군집화는 미리 주어진 범주 표시 없이 **서로 비슷한 데이터끼리 묶는 비지도 학습**의 대표적인 기법이다.

② X: 분류와 회귀분석은 대표적인 **지도 학습** 기법이다. 분류는 학습 데이터의 범주 표시를 이용하여 새로운 데이터의 범주를 예측하고, 회귀분석은 학습 데이터의 수치 목표값을 이용하여 새로운 데이터의 수치 값을 예측한다.

③ X: 행동의 결과로 주어지는 보상을 바탕으로 학습하는 방식은 **강화 학습**이다. 강화 학습은 환경과 상호작용하면서 누적 보상을 크게 하는 행동 전략을 학습하며, 입력과 레이블의 관계를 배우는 지도 학습과 구별된다.

④ O: 비지도 학습은 레이블이 없는 데이터에서 **유사성이나 분포 등 데이터 자체의 구조**를 찾아낸다. 예를 들어 고객별 구매 내역만 제공하고 고객 유형은 알려 주지 않은 상태에서, 구매 성향이 비슷한 고객들을 묶는 군집화가 이에 해당한다.

NoteBase

문 4. 무선주파수를 이용하며 반도체 칩이 내장된 태그와 리더기로 구성된 인식시스템은?

① RFID

② WAN

③ Bluetooth

④ ZigBee

정답 1번

간단한 해설

반도체 칩이 내장된 **태그의 정보를 리더기가 무선주파수로 읽어** 대상을 식별하는 기술이 RFID이다.

공통 해설

RFID(Radio Frequency Identification)는 무선주파수를 이용해 태그에 저장된 정보를 읽고 대상을 식별하는 기술이다. 태그의 반도체 칩에는 식별 정보 등이 저장되며, 리더기는 태그와 무선으로 통신하여 그 정보를 읽는다. 따라서 물리적으로 접촉하지 않고도 태그가 부착된 대상을 인식할 수 있다. 이 문항에서 후보를 구별하는 기준은 단순히 무선 통신을 이용하는지가 아니라 **정보를 담은 태그와 이를 읽는 리더기로 대상을 식별하는지**이며, 이 구성은 RFID에 해당한다.

NoteBase

문 5. 클라우드 컴퓨팅에 대한 설명으로 옳지 않은 것은?

① 클라우드 컴퓨팅은 기업의 IT 요구를 매우 경제적이고, 신뢰성 있게 충족시킬 수 있는 수단이 된다.

② 클라우드 컴퓨팅 서비스 모델에는 IaaS, PaaS, SaaS가 있다.

③ 클라우드 컴퓨팅을 이용하는 방식에는 사설 클라우드, 공용 클라우드, 하이브리드 클라우드가 있다.

④ IaaS를 통해 사용자는 소프트웨어 설치 및 유지보수에 대한 비용을 절감할 수 있다.

정답 4번

간단한 해설

소프트웨어 설치·유지보수 부담을 제공자에게 맡기는 서비스 모델은 **SaaS**이다. IaaS에서는 **운영체제와 응용 소프트웨어를 사용자가 관리**한다.

선지별 해설

① O: 클라우드 컴퓨팅은 서버·저장 공간 등의 자원을 네트워크로 제공한다. 필요한 만큼 자원을 이용하고 수요에 따라 늘리거나 줄일 수 있어 **초기 구축 비용과 유휴 자원을 줄일 수 있다**. 자원의 분산·중복 구성 등을 활용하면 서비스의 신뢰성도 높일 수 있다.

② O: IaaS는 서버·저장 공간·네트워크 등 기반 자원을, PaaS는 응용 프로그램을 개발·배포·실행할 수 있는 플랫폼을, SaaS는 제공자가 운영하는 완성된 응용 소프트웨어를 제공한다. 세 모델은 **어디까지 서비스로 제공하는지**에 따라 구분된다.

③ O: 사설 클라우드는 **특정 조직이 전용으로 이용**하고, 공용 클라우드는 일반 이용자에게 개방하여 제공한다. 하이브리드 클라우드는 사설·공용 클라우드 등 서로 구분되는 클라우드를 연결하여 함께 활용하는 방식이다.

④ X: IaaS는 서버·저장 공간 등 인프라를 제공하므로 물리 장비의 구입·관리 부담을 줄일 수 있지만, **운영체제와 응용 소프트웨어의 설치·유지보수는 사용자가 담당**한다. 제공자가 응용 소프트웨어를 운영하여 사용자의 설치·유지보수 부담을 줄이는 특징은 SaaS에 해당한다. 예를 들어 IaaS의 가상 서버에 업무 프로그램을 설치했다면 그 프로그램의 업데이트도 사용자가 관리한다.

NoteBase

문 6. C 언어에서 함수 호출 시 매개변수 전달 방법에는 값에 의한 호출(Call by Value)과 참조에 의한 호출(Call by Reference)이 있다. C 프로그램 코드가 다음과 같을 때 설명으로 옳지 않은 것은?

```
int get_average(int score[], int n) {
      int i, sum;
      for(i = 0; i < n; i++)
            sum += score[i];
      return sum / n;
}
void main(void) {
      int score[3] = { 1, 2, 5 };
      printf("%d\n", get_average(score, 3));
}
```

① 전달할 데이터의 양이 많을 경우에는 참조에 의한 호출이 효율적이다.

② 값에 의한 호출로 전달된 데이터는 호출된 함수에서 값을 변경하더라도 함수 종료 후 해당 함수를 호출한 상위 함수에 반영되지 않는다.

③ 값에 의한 호출은 함수 호출 시 데이터 복사가 발생한다.

④ 위의 프로그램에서 함수 get_average()를 호출하는 데 사용한 매개변수 score는 값에 의한 호출로 처리된다.

정답 4번

간단한 해설

score는 배열 원소들을 복사해 전달하는 것이 아니라 **첫 원소의 주소를 전달**한다. 이 문제에서는 이처럼 원본 배열에 접근하는 방식을 '참조에 의한 호출'로 구분한다.

NoteBase

선지별 해설

① O: 참조에 의한 호출은 데이터 전체를 복사하는 대신 원본 데이터에 접근할 수 있는 주소나 참조를 전달하는 방식이다. 전달할 데이터가 많아도 **데이터 전체의 복사를 피할 수 있어** 복사에 필요한 시간과 저장 공간을 줄일 수 있다.

② O: 값에 의한 호출에서는 호출한 쪽의 값을 복사하여 별도의 매개변수에 저장한다. 호출된 함수가 **매개변수 자체에 다른 값을 대입**해도 바뀌는 것은 복사본이므로 호출한 쪽의 원래 변수는 바뀌지 않는다.

③ O: 값에 의한 호출은 **인수의 값을 매개변수로 복사**하여 전달한다. 예를 들어 get_average(score, 3)에서 두 번째 인수 3은 매개변수 n에 복사되어, 함수가 n을 별도로 사용할 수 있다.

④ X: get_average()의 매개변수 선언 int score[]는 함수 매개변수에서 int *score와 같은 뜻이다. 호출 시 main()의 배열 score는 첫 원소를 가리키는 포인터로 변환되므로, 배열의 1, 2, 5가 별도의 배열로 복사되는 것이 아니라 **원본 배열의 첫 원소 주소가 전달**된다. 따라서 함수의 score[i]는 main()의 배열 원소에 접근하며, 함수 안에서 score[0]에 다른 값을 대입하면 원본 배열의 첫 원소도 바뀐다. 이 문제의 구분에서는 이를 '참조에 의한 호출'로 고쳐야 한다. 다만 C의 엄밀한 언어 규칙에서는 모든 인수를 값으로 전달하며, 여기서도 포인터 값 자체는 복사된다. '배열 원소 전체의 복사'와 '주소 값의 복사'를 구별해야 한다.

NoteBase

문 7. 다음 C 프로그램에서 밑줄 친 코드의 실행 결과와 동일한 결과를 출력하는 코드로 옳은 것만을 모두 고르면?

```
#include <stdio.h>
int main()
{
      int ary[5] = {10, 11, 12, 13, 14};
      int *ap;
      ap = ary;
      printf("%d", ary[1]);
      return 0;
}
```

ㄱ. printf("%d", ary+1);
ㄴ. printf("%d", *ap+1);
ㄷ. printf("%d", *ary+1);
ㄹ. printf("%d", *ap++);

① ㄱ, ㄴ
② ㄴ, ㄷ
③ ㄷ, ㄹ
④ ㄴ, ㄷ, ㄹ

정답 2번

간단한 해설

ap와 ary가 가리키는 첫 원소의 값은 10이므로, **역참조한 값에 1을 더하면** 밑줄 친 코드와 같은 11이 나온다.
*ap++는 포인터를 이동시키지만 **이동 전 포인터로 읽은 값**을 사용하므로 10을 출력한다.

NoteBase

보기별 해설

배열의 첨자는 0부터 시작하므로 ary[0]은 10, ary[1]은 11이다. ap = ary를 실행하면 ap는 첫 원소 ary[0]을 가리킨다. 따라서 밑줄 친 printf의 출력은 11이다. 각 보기는 이 초기 상태에서 밑줄 친 코드를 각각 대체하는 것으로 판단한다.

ㄱ. X: ary+1은 두 번째 원소 ary[1]의 **주소**이지 그 원소의 값이 아니다. 또한 %d는 int형 인수를 요구하므로 포인터를 그대로 전달하면 동작이 정의되지 않는다. 11을 출력하려면 주소를 역참조하여 printf("%d", *(ary+1));로 고쳐야 한다.

ㄴ. O: *ap+1은 (*ap)+1로 해석된다. *ap가 첫 원소의 값 10이므로 10 + 1 = 11을 출력한다. 이는 **포인터가 가리키는 위치를 바꾸는 연산이 아니라 값에 1을 더하는 연산**이다.

ㄷ. O: 이 식에서 ary는 첫 원소를 가리키는 포인터로 변환되고, *ary는 ary[0]의 값 10이다. *ary+1은 (*ary)+1이므로 10 + 1 = 11을 출력한다.

ㄹ. X: 후위 증가 연산자 ++가 역참조 연산자 *보다 우선하므로 *ap++는 *(ap++)로 해석된다. ap++의 식 값은 **증가 전 포인터**이므로 첫 원소의 값 10을 읽어 출력하고, ap는 다음 원소를 가리키도록 바뀐다. 11을 출력하려면 printf("%d", *(ap+1));로 고쳐 다음 원소의 값을 읽어야 한다.

밑줄 친 코드와 같은 11을 출력하는 보기는 ㄴ, ㄷ이며, 이는 ②의 조합이다.

NoteBase

문 8. 자료 흐름의 방향과 동시성 여부에 따라 분류한 통신 방식 중 다음에서 설명하는 통신 방식으로 옳은 것은? (단, DTE(Data Terminal Equipment)는 컴퓨터, 휴대폰, 단말기 등과 같이 통신망에서 네트워크의 끝에 연결된 장치들을 총칭하는 용어이다)

통신하는 두 DTE가 시간적으로 교대로 데이터를 교환하는 방식의 통신으로, 한 DTE가 명령을 전송하면 다른 DTE가 이를 처리하여 그에 대한 응답을 전송하는 트랜잭션(Transaction) 처리 시스템에서 볼 수 있다.

① 단방향 통신
② 반이중 통신
③ 전이중 통신
④ 원거리 통신

정답 2번

간단한 해설

두 DTE가 양쪽으로 데이터를 주고받되 **시간적으로 교대로** 전송하므로 반이중 통신에 해당한다. 한쪽이 명령을 보낸 뒤 다른 쪽이 응답을 보내는 동안에는 송신과 수신의 역할이 바뀐다.

공통 해설

자료 흐름의 방향과 동시성으로 통신 방식을 구분할 때, 한쪽 방향으로만 전송하면 단방향 통신이고 양방향으로 전송할 수 있으면 동시 송수신이 가능한지를 구별한다. 양방향 전송을 **번갈아 수행**하는 방식은 반이중 통신이며, 양방향 전송을 **동시에 수행**할 수 있는 방식은 전이중 통신이다. 원거리 통신이라는 명칭은 통신 거리에 관한 것으로, 송수신 방향과 동시성 여부를 나타내지 않는다.

지문에서는 처음에 한 DTE가 명령을 보내고 다른 DTE가 이를 받는다. 이어 명령을 받은 DTE가 처리 결과를 응답으로 보내면, 처음 명령을 보낸 DTE가 이를 받는다. 즉, 양쪽 모두 송신과 수신을 하지만 시간적으로 교대하므로 반이중 통신의 조건에 맞는다.

NoteBase

문 9. 다음 라우팅 테이블에 대한 설명으로 옳지 않은 것은?

목적지 네트워크	서브넷마스크	인터페이스
128.50.30.0	255.255.254.0	R1
128.50.28.0	255.255.255.0	R2
Default		R3

① 목적지 IP 주소가 128.50.30.92인 패킷과 128.50.31.92인 패킷은 서로 다른 인터페이스로 전달된다.

② 128.50.28.0 네트워크에 대한 브로드캐스트 주소는 128.50.28.255다.

③ 서브넷마스크 255.255.254.0은 CIDR 표기에 의해 /23으로 표현된다.

④ 이 라우터는 목적지 IP 주소가 128.50.28.9인 패킷을 R2로 전달한다.

정답 1번

간단한 해설

128.50.30.92와 128.50.31.92는 모두 128.50.30.0/23 네트워크에 속하므로 **같은 인터페이스 R1**로 전달된다. 서브넷마스크가 255.255.254.0이면 이 네트워크의 주소 범위는 128.50.30.0~128.50.31.255이다.

NoteBase

선지별 해설

① X: 라우팅 테이블의 각 항목에서 목적지 IP 주소와 서브넷마스크를 비트별 AND 연산한 결과가 목적지 네트워크 주소와 같으면 그 경로에 해당한다. R1 항목의 마스크 255.255.254.0에서 세 번째 옥텟 254는 이진수 11111110이므로 마지막 1비트는 네트워크 판별에 사용하지 않는다. 30은 이진수 00011110, 31은 00011111이어서 마스크를 적용하면 둘 다 30이 된다. 따라서 두 목적지 주소 모두 네트워크 주소가 128.50.30.0으로 계산되며, 서로 다른 인터페이스가 아니라 **같은 R1**로 전달된다.

② O: 표에서 128.50.28.0 네트워크의 서브넷마스크는 255.255.255.0이다. 앞의 24비트가 네트워크 부분이고 마지막 8비트가 호스트 부분이다. 브로드캐스트 주소는 **호스트 비트를 모두 1**로 설정하므로 마지막 옥텟이 255가 되어 128.50.28.255이다.

③ O: CIDR의 /뒤 숫자는 서브넷마스크에서 **앞에서부터 연속된 1의 개수**를 뜻한다. 255.255.254.0을 이진수로 쓰면 11111111.11111111.11111110.00000000이며, 1의 개수는 8 + 8 + 7 = 23개이다. 따라서 /23으로 표현한다.

④ O: 목적지 주소 128.50.28.9에 서브넷마스크 255.255.255.0을 적용하면 마지막 옥텟이 0이 되어 네트워크 주소 128.50.28.0을 얻는다. 이는 표의 두 번째 항목과 일치하므로 **R2로 전달**한다. 일치하는 구체적인 경로가 있으므로 기본 경로인 R3는 사용하지 않는다.

NoteBase

문 10. 3단계 데이터베이스 구조에서 개념 스키마에 대한 설명으로 옳은 것만을 모두 고르면?

ㄱ. 데이터베이스를 운영하는 기관에 소속되어 있는 모든 응용시스템 또는 사용자들이 필요로 하는 데이터를 통합하여 정의한 조직 전체 데이터베이스의 논리 구조를 말한다.
ㄴ. 개념 스키마와 외부 스키마 사이에는 논리적 데이터 독립성이 있어야 한다.
ㄷ. 데이터베이스 내에는 하나의 개념 스키마만 존재한다.
ㄹ. 데이터에 대한 접근권한, 제약조건 등에 대한 정의도 포함한다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ, ㄹ
④ ㄱ, ㄴ, ㄷ, ㄹ

정답 4번

간단한 해설

개념 스키마는 조직 전체의 데이터를 통합한 **논리적 구조**로, 데이터 항목과 관계뿐 아니라 **접근권한과 제약조건**도 정의한다.

보기별 해설

ㄱ. O: 개념 스키마는 모든 응용시스템과 사용자의 데이터 요구를 통합하여 **조직 전체 데이터베이스의 논리 구조**를 정의한다. 데이터의 종류, 속성, 데이터 사이의 관계 등을 기술하며, 개별 사용자가 보는 일부 데이터의 구조를 정의하는 외부 스키마와 구별된다.

ㄴ. O: **논리적 데이터 독립성**은 개념 스키마가 변경되어도 외부 스키마와 이를 사용하는 응용프로그램을 변경하지 않을 수 있는 성질이다. 개념 스키마의 변경은 외부 스키마와 개념 스키마 사이의 사상을 조정하여 반영하므로, 두 스키마 사이에서 보장되어야 한다.

ㄷ. O: 개념 스키마는 조직 전체의 데이터를 통합한 정의이므로 하나의 데이터베이스에 **하나만 존재한다**. 사용자나 응용시스템마다 필요한 데이터가 다르므로 외부 스키마는 여러 개 존재할 수 있다.

ㄹ. O: 개념 스키마에는 데이터의 구조뿐 아니라 **접근권한과 무결성 제약조건**도 포함된다. 접근권한은 누가 어떤 데이터에 접근할 수 있는지를 정하고, 무결성 제약조건은 학번이 중복되어서는 안 된다는 규칙처럼 데이터가 지켜야 할 조건을 정한다.

ㄱ, ㄴ, ㄷ, ㄹ이 모두 옳으므로, 네 보기를 모두 포함한 조합은 ④이다.

NoteBase

문 11. TCP(Transmission Control Protocol) 기반 응용 프로토콜에 해당하지 않는 것은?

① Telnet

② FTP

③ SMTP

④ SNMP

정답 4번

간단한 해설

SNMP는 네트워크 장비의 상태를 감시하고 관리하는 응용 프로토콜로, 기본적으로 **UDP를 사용**한다. TCP 기반으로 분류되는 Telnet·FTP·SMTP와 구별된다.

개념 구분

TCP 기반: TCP는 연결을 설정한 뒤 데이터가 순서대로 신뢰성 있게 전달되도록 하는 전송 계층 프로토콜이다. 원격 접속에 쓰이는 Telnet, 파일 전송에 쓰이는 FTP, 전자우편 전송에 쓰이는 SMTP는 **TCP를 이용하는 응용 프로토콜**이다. 응용 프로토콜의 용도는 서로 달라도 그 아래에서 사용하는 전송 계층 프로토콜은 같다.

SNMP: 라우터나 스위치 등 네트워크 장비의 상태 정보를 조회하고 관리하는 프로토콜이다. 기본적인 SNMP 통신은 **UDP 기반**으로 이루어지며, UDP는 TCP처럼 연결을 먼저 설정하지 않고 데이터그램을 전달한다. 따라서 이 문항에서 TCP 기반 응용 프로토콜에 해당하지 않는 것은 SNMP이다.

NoteBase

문 12. 운영체제에서 프로세스의 정보를 관리하는 프로세스 제어블록(Process Control Block)의 포함 요소로 옳지 않은 것은?

① 프로세스 식별자

② 인터럽트 정보

③ 프로세스의 우선순위

④ 프로세스의 상태

정답 2번

간단한 해설

②의 '인터럽트 정보'는 **통상적인 PCB의 기본 구성 항목으로 분류하지 않는다**. 다만 인터럽트로 프로세스 실행이 중단될 때 저장하는 프로그램 카운터와 레지스터 값은 실행 재개에 필요한 정보이므로 구별해야 한다.

개념 구분

프로세스 제어블록(PCB)은 운영체제가 개별 프로세스를 식별하고 실행을 관리하기 위한 자료구조이다. 프로세스 식별자는 어느 프로세스인지를 구별하고, 우선순위는 CPU를 배정할 때의 판단 기준이 되며, 상태는 준비·실행·대기 등 현재 상황을 나타낸다. 이들은 모두 **프로세스별 관리 정보**이므로 PCB의 대표적인 포함 요소이다.

'인터럽트 정보'는 이러한 통상적인 PCB 구성 분류에서 별도의 기본 항목으로 보지 않는다. 그러나 이를 인터럽트와 관련된 값이 PCB에 전혀 저장되지 않는다는 뜻으로 이해해서는 안 된다. 인터럽트로 실행이 중단되어 다른 프로세스로 전환할 때는, 나중에 이어 실행할 수 있도록 프로그램 카운터와 CPU 레지스터 값 등 **중단 시점의 실행 문맥**을 저장한다. 구체적인 구현에 따라 인터럽트 허용 상태 같은 제어·상태 레지스터의 값도 이 문맥에 포함될 수 있다.

NoteBase

문 13. SSD(Solid-State Drive)에 대한 설명으로 옳지 않은 것은?

① 반도체 기억장치 칩들을 이용하여 구성된 저장장치이다.

② 하드디스크에 비해 저장용량 대비 가격이 비싸다.

③ 기계적 장치를 사용하여 하드디스크보다 데이터 입출력 속도가 빠르다.

④ 하드디스크를 대체하려고 개발한 저장장치로서 플래시 메모리로 구성된다.

정답 3번

간단한 해설

③의 '기계적 장치를 사용하여'가 잘못된 설명이다. SSD는 **움직이는 기계 부품 없이** 반도체 메모리에 전자적으로 접근하므로, 디스크 회전과 헤드 이동이 필요한 하드디스크보다 데이터 입출력이 빠르다.

선지별 해설

① O: SSD는 **반도체 기억장치 칩**에 데이터를 저장하는 장치이다. 하드디스크처럼 회전하는 자기 디스크에 데이터를 기록하는 방식과 구별된다.

② O: 일반적으로 SSD는 하드디스크보다 **단위 저장용량당 가격**이 높다. 이는 제품 하나의 총가격이 아니라 같은 용량을 확보하는 데 드는 비용을 비교한 것이다.

③ X: SSD에는 하드디스크의 회전 디스크나 이동하는 헤드 같은 기계적 구동부가 없다. 반도체 메모리에 전자적으로 접근하므로 **헤드 이동과 디스크 회전에 따른 대기 시간이 없다**. 따라서 입출력이 빠른 이유는 기계적 장치를 사용하기 때문이 아니라, 이러한 기계적 동작이 필요 없기 때문이다.

④ O: 일반적인 SSD는 하드디스크를 대체하는 저장장치로, **전원이 꺼져도 데이터가 유지되는 플래시 메모리**를 사용한다. 따라서 운영체제나 파일을 지속적으로 보관하는 보조기억장치로 사용할 수 있다.

NoteBase

문 14. 다음 후위 표기 식을 전위 표기 식으로 변환하였을 때 옳은 것은?

3 1 4 1 - * +

① 3 + 1 * 4 - 1

② 4 - 1 * 1 + 3

③ + 3 * 1 - 4 1

④ + 3 - 4 1 * 1

정답 3번

간단한 해설

후위 표기 식에서 '4 1 -'가 먼저 하나의 부분식을 이루고, 여기에 1을 곱한 뒤 3을 더한다. 전위 표기에서는 각 **연산자를 두 피연산자 앞에** 놓으므로 '+ 3 * 1 - 4 1'이 된다.

계산 과정

후위 표기는 '왼쪽 피연산자 오른쪽 피연산자 연산자', 전위 표기는 '연산자 왼쪽 피연산자 오른쪽 피연산자' 순서이다. 주어진 식을 왼쪽부터 읽으며 피연산자를 스택에 넣고, 연산자를 만나면 두 항목을 꺼내 전위 부분식으로 묶어 다시 넣는다. 이때 **먼저 꺼낸 항목이 오른쪽 피연산자**이고, 다음에 꺼낸 항목이 왼쪽 피연산자이다.

– 처리: '3 1 4 1'을 읽은 스택은 아래에서부터 3, 1, 4, 1이다. 첫 연산자 '-'를 만나면 맨 위의 1과 그 아래의 4를 꺼낸다. 왼쪽이 4, 오른쪽이 1이므로 전위 부분식은 '- 4 1'이다. 이를 다시 넣으면 스택에는 3, 1, '- 4 1'이 남는다.

* 처리: 맨 위의 부분식 '- 4 1'과 그 아래의 1을 꺼낸다. **부분식 전체가 하나의 피연산자**이므로, 1을 왼쪽에 두고 '- 4 1'을 오른쪽에 둔 전위 부분식은 '* 1 - 4 1'이다. 스택에는 3과 '* 1 - 4 1'이 남는다.

+ 처리: 오른쪽 피연산자는 '* 1 - 4 1', 왼쪽 피연산자는 3이다. 두 피연산자 앞에 '+'를 붙이면 전체 전위 표기 식은 '+ 3 * 1 - 4 1'이 된다. 이는 ③의 식과 일치한다.

NoteBase

문 15. 운영체제의 세마포어(Semaphore)에 대한 설명으로 옳지 않은 것은?

① 프로세스 간 상호배제(Mutual Exclusion)의 원리를 보장하는 데 사용된다.

② 여러 개의 프로세스가 동시에 그 값을 수정하지 못한다.

③ 세마포어에 대한 연산은 수행 중에 인터럽트 될 수 있다.

④ 세마포어는 플래그 변수와 그 변수를 검사하거나 증감시키는 연산들로 정의된다.

정답 3번

간단한 해설

세마포어 값을 검사하고 변경하는 연산은 **원자적으로 수행**되어야 한다. 연산의 중간 상태에 다른 프로세스가 끼어들어 같은 값을 변경할 수 있으면 상호배제를 보장할 수 없다.

선지별 해설

① O: 세마포어는 여러 프로세스가 공유 자원에 접근하는 순서를 조정한다. 상호배제를 위해 초기값을 1로 설정하면, 한 프로세스가 진입 허가를 얻은 동안 다른 프로세스는 기다리므로 **임계구역에 하나의 프로세스만 진입**하게 할 수 있다. 임계구역은 공유 자원에 접근하는 코드 부분이다.

② O: 세마포어 값의 검사와 갱신은 **원자적 연산**으로 보호된다. 따라서 여러 프로세스가 같은 세마포어를 사용하더라도 그 값을 동시에 수정하여 갱신 결과가 뒤섞이지 않는다.

③ X: 세마포어 연산은 **중간에 다른 프로세스의 접근이 끼어들지 않는 불가분의 연산**이어야 한다. 선지의 '수행 중에 인터럽트 될 수 있다'는 이러한 원자성에 어긋나는 설명이다. 예를 들어 값이 1인지 검사하는 과정과 이를 0으로 바꾸는 과정 사이에 다른 프로세스가 끼어들면, 두 프로세스가 모두 사용 가능하다고 판단하여 임계구역에 진입할 수 있다. 따라서 값의 검사와 변경을 하나의 원자적 동작으로 보장해야 한다.

④ O: 세마포어는 자원의 사용 가능 상태를 나타내는 **정수 변수와 그 값을 다루는 연산**으로 정의된다. 대표적으로 P(wait)는 사용 가능 여부를 확인하여 자원을 확보하고, 자원이 없으면 기다리게 한다. V(signal)는 자원을 반납하고 필요하면 대기 중인 프로세스를 깨운다. 선지의 플래그 변수는 이러한 제어용 변수를 뜻하며, 세마포어가 항상 참·거짓 두 값만 갖는다는 뜻은 아니다.

NoteBase

문 16. 소프트웨어에 대한 ISO/IEC 품질 표준 중에서 프로세스 품질 표준으로 옳은 것은?

① ISO/IEC 12119

② ISO/IEC 12207

③ ISO/IEC 14598

④ ISO/IEC 25010

정답 2번

간단한 해설

ISO/IEC 12207은 소프트웨어의 개발·운영·유지보수 등 **생명주기 프로세스**를 규정한다. 소프트웨어 제품의 특성보다 그 제품을 만들고 관리하는 활동을 다루므로 프로세스 품질 표준에 해당한다.

개념 구분

ISO/IEC 12207: 소프트웨어의 개발·운영·유지보수 등을 수행하는 데 필요한 **프로세스·활동·작업**의 공통 체계를 제시한다. 프로세스는 일을 수행하는 과정이므로, 문항의 프로세스 품질 표준에 해당한다.

ISO/IEC 12119: 제공·납품되는 **소프트웨어 패키지의 품질 요구사항과 시험**을 다룬다. 패키지가 요구사항을 충족하는지 확인하는 표준이며, 패키지를 생산하는 과정 자체는 적용 범위에 포함하지 않는다.

ISO/IEC 14598: **소프트웨어 제품의 품질 평가**를 다루는 표준이다. 평가 절차를 다룬다는 점 때문에 혼동하기 쉽지만, 평가 대상은 소프트웨어 제품의 품질이다.

ISO/IEC 25010: 제품의 품질을 명세·측정·평가할 때 기준이 되는 **제품 품질모델**을 정의한다. 개발·운영 활동의 체계보다 제품이 갖추어야 할 품질 특성을 다룬다는 점에서 12207과 구별된다.

NoteBase

문 17. 블록체인(Block Chain)에 대한 설명으로 옳지 않은 것은?

① 블록에는 트랜잭션(Transaction)이 저장되어 있다.

② 스마트 컨트랙트(Smart Contract)는 실세계의 계약이 블록체인에서 이루어질 수 있도록 하는 기술이다.

③ 중앙 서버를 통해 전파된 블록은 네트워크에 참가한 개별 노드에서 유효성을 검증받은 후, 중앙 서버로 다시 전송된다.

④ 블록체인은 공개범위에 따라 Public 블록체인과 Private 블록체인으로 나눌 수 있다.

정답 3번

간단한 해설

③의 '중앙 서버로 다시 전송된다'는 설명이 잘못되었다. 블록체인은 **중앙 서버를 거치지 않고 노드 간에 블록을 전파**하며, 노드들이 유효성을 검증하고 합의 규칙에 따라 원장을 갱신하는 분산 구조이다.

선지별 해설

① O: 트랜잭션은 자산의 이전이나 데이터 변경 등의 거래 기록이다. 블록에는 이러한 **트랜잭션 데이터**가 담기며, 블록들이 이전 블록의 해시값을 통해 연결되어 거래 이력을 구성한다.

② O: 스마트 컨트랙트는 **계약 조건과 이행 절차를 프로그램으로 구현**하여 블록체인에서 실행하는 기술이다. 정해진 조건이 충족되면 약속된 동작을 수행하므로, 실세계 계약의 일부 절차를 자동화할 수 있다.

③ X: 블록체인은 참여 노드들이 원장을 분산하여 유지하는 구조이다. 새 블록은 **중앙 서버를 거치지 않고 노드 간에 전파**되며, 이를 받은 노드는 거래와 블록이 정해진 규칙에 맞는지 검증한다. 원장은 해당 블록체인의 합의 규칙에 따라 갱신되므로, 검증 결과를 중앙 서버로 다시 보내 확정받는 방식이라는 설명은 잘못되었다.

④ O: 퍼블릭 블록체인은 일반에 개방되어 누구나 참여할 수 있는 방식이고, 프라이빗 블록체인은 특정 조직 등이 **허가한 참여자로 접근을 제한**하는 방식이다. 이처럼 공개 및 참여 허용 범위에 따라 구분할 수 있다.

NoteBase

문 18. 아래의 고객 릴레이션에서 등급이 gold이고 나이가 25 이상인 고객들을 검색하기 위해 기술한 관계대수 표현으로 옳은 것은?

<고객 릴레이션>

고객				
고객아이디	이름	나이	등급	직업
hohoho	이순신	29	gold	교사
grace	홍길동	24	gold	학생
mango	삼돌이	27	silver	학생
juce	갑순이	31	gold	공무원
orange	강감찬	23	silver	군인

<검색결과>

고객아이디	이름	나이	등급	직업
hohoho	이순신	29	gold	교사
juce	갑순이	31	gold	공무원

① _고객(등급 = ‘gold’ 나이 ≥ 25)

② _등급 = ‘gold’ 나이 ≥ 25(고객)

③ _고객(등급 = ‘gold’ 나이 ≥ 25)

④ _등급 = ‘gold’ 나이 ≥ 25(고객)

정답 2번

간단한 해설

조건을 만족하는 고객의 **행을 고르는 셀렉션(σ)**을 사용한다. σ의 아래첨자에는 '등급이 gold이고 나이가 25 이상'이라는 조건을, 괄호 안에는 대상 릴레이션인 고객을 쓴다.

NoteBase

핵심 풀이

연산의 구분: 셀렉션(σ)은 릴레이션에서 **조건을 만족하는 행(튜플)**을 골라내고, 프로젝션(π)은 지정한 열(속성)을 골라낸다. 검색결과는 고객아이디·이름·나이·등급·직업이라는 열 구성을 유지하면서 일부 고객의 행만 남기므로 셀렉션에 해당한다. π를 사용한 ③·④는 이 검색 연산을 나타내지 못한다.

표현의 구성: 셀렉션은 _조건(릴레이션)으로 쓴다. 두 조건을 모두 만족해야 하므로 논리곱 ∧로 연결하여 _등급 = 'gold' 나이 ≥ 25(고객)으로 표현한다. 이는 ②의 표현이다. ①은 **조건과 릴레이션의 위치를 바꾼** 것으로, 아래첨자의 고객과 괄호 안의 조건을 서로 바꾸어야 한다.

자료에 적용: 이순신은 gold·29세, 갑순이는 gold·31세이므로 두 조건을 모두 만족한다. 홍길동은 gold이지만 24세이고, 삼돌이는 27세이지만 silver이므로 제외된다. 강감찬은 silver·23세로 두 조건을 모두 만족하지 못한다. 따라서 검색결과에는 이순신과 갑순이의 행이 남으며, 제시된 검색결과와 일치한다.

NoteBase

문 19. (가)에 들어갈 어드레싱 모드로 옳은 것은?

(가)는 명령어가 피연산자의 주소를 가지고 있는 레지스터를 지정한다. 즉, 선택된 레지스터는 피연산자 그 자체가 아니라 피연산자의 주소이다. 일반적으로 이 모드를 사용할 때에 프로그래머는 이전의 명령어에서 레지스터가 피연산자의 주소를 가졌는지를 확인해 보아야 한다.

① 레지스터 간접 모드(Register Indirect mode)

② 레지스터 모드(Register mode)

③ 간접 주소 모드(Indirect Addressing mode)

④ 인덱스 어드레싱 모드(Indexed Addressing mode)

정답 1번

간단한 해설

명령어가 지정한 레지스터에 피연산자 자체가 아니라 **피연산자의 메모리 주소**가 들어 있으므로 레지스터 간접 모드이다.

공통 해설

레지스터 간접 모드는 **명령어가 지정한 레지스터의 내용을 피연산자의 메모리 주소로 사용하는 방식**이다.
명령어는 레지스터를 지정하고, 프로세서는 그 레지스터에 들어 있는 주소로 메모리에 접근하여 실제 피연산자를 얻는다. 판단 기준은 레지스터를 사용한다는 사실만이 아니라, 그 안의 값을 데이터 자체가 아닌 주소로 사용한다는 점이다.

예를 들어 R1에 100이 들어 있고 메모리 주소 100에 25가 저장되어 있다면, R1을 지정한 레지스터 간접 모드에서 피연산자는 25이다. 100은 데이터를 찾기 위한 주소이고, 25가 실제 연산에 사용되는 데이터이다.

이 방식으로 원하는 피연산자에 접근하려면 실행 시 해당 레지스터에 올바른 주소가 들어 있어야 한다. 지문의 이전 명령어에 관한 설명은 레지스터에 필요한 주소가 미리 준비되어 있어야 한다는 뜻이다.

따라서 주소를 담은 레지스터를 지정한다는 지문의 설명에 해당하는 모드는 레지스터 간접 모드이다.

NoteBase

문 20. 디스크 큐에 다음과 같이 I/O 요청이 들어와 있다. 최소탐색시간 우선(SSTF) 스케줄링 적용 시 발생하는 총 헤드 이동 거리는? (단, 추가 I/O 요청은 없다고 가정한다. 디스크 헤드는 0부터 150까지 이동 가능하며, 현재 위치는 50이다)

큐: 80, 20, 100, 30, 70, 130, 40

① 100

② 140

③ 180

④ 430

정답 2번

간단한 해설

SSTF는 **현재 헤드 위치에서 가장 가까운 미처리 요청**을 매번 선택한다. 이 기준으로 이동한 각 구간의 거리를 합하면 총 헤드 이동 거리는 140이다.

계산 과정

SSTF(최소탐색시간 우선)는 현재 헤드 위치와 각 미처리 요청 위치의 차이를 비교하여, **이동 거리가 가장 짧은 요청**부터 처리한다. 요청을 처리할 때마다 바뀐 헤드 위치를 기준으로 다시 선택한다. 현재 위치는 50이며, 추가 요청 없이 주어진 7개 요청만 처리한다.

먼저 50에서 가장 가까운 요청은 40으로, 이동 거리는 10이다. 이어서 40에서는 30이, 30에서는 20이 가장 가까우므로 각각 10씩 이동한다. 따라서 처음 세 요청의 처리 경로는 50 → 40 → 30 → 20이고, 누적 이동 거리는 10 + 10 + 10 = 30이다.

20에 도착하면 남은 요청은 70, 80, 100, 130이다. 이 중 가장 가까운 70까지의 거리는 70 − 20 = 50이므로, **이동 방향을 바꾸어 70으로 이동**한다. 이때 누적 이동 거리는 30 + 50 = 80이다.

70에서는 80까지 10, 80에서는 100까지 20, 100에서는 130까지 30을 차례로 이동한다. 전체 경로는 50 → 40 → 30 → 20 → 70 → 80 → 100 → 130이다. 0과 150은 이동 가능 범위의 끝점일 뿐이며, 그 위치에 요청이 없으므로 방문할 필요가 없다.

총 헤드 이동 거리는 각 구간의 위치 차이를 절댓값으로 구해 더한다. 따라서 10 + 10 + 10 + 50 + 10 + 20 + 30 = 140으로, ②의 값과 일치한다.

NoteBase