

2016-계리직-컴퓨터일반-A형-해설-곽후근

1. 2진수 11110000과 10101010에 대해 XOR논리 연산을 수행한 결과 값을 16진수로 바르게 표현한 것은?

- ① 5A
- ② 6B
- ③ A5
- ④ B6

정답 체크 :

(1)

XOR 수행 : 11110000 XOR 10101010 = 01011010 // XOR은 입력에서 1의 개수가 홀수 개일 때 출력이 1이 된다. 첫 번째 비트는 두 입력 모두 1이므로 1의 개수가 짝수개가 되어 출력이 0이 된다. 두 번째 비트는 첫 번째 입력은 1이고 두 번째 입력은 0이므로 1의 개수가 홀수개가 되어 출력이 1이 된다. 나머지도 이와 마찬가지로 정리하면 된다.

16진수 변환 : 01011010₍₂₎ = 5A₍₁₆₎

2. 무선 네트워크 방식에 대한 설명으로 옳은 것은?

- ① 블루투스(Bluetooth)는 동일한 유형의 기기 간에만 통신이 가능 하다.
- ② NFC방식이 블루투스 방식보다 최대 전송 속도가 빠르다.
- ③ NFC방식은 액세스 포인트(access point) 없이 두 장치 간의 통신이 가능하다.
- ④ 최대 통신 가능거리를 가까운 것에서 먼 순서로 나열하면 Bluetooth < Wi-Fi < NFC < LTE 순이다.

정답 체크 :

(3) NFC와 Bluetooth는 AP(액세스 포인트, Wi-Fi에서 필요), BS(기지국, LTE에서 필요) 없이 두 장치 간의 통신이 가능하다.

오답 체크 :

- (1) 블루투스는 블루투스를 지원하는 기기라면 유형에 상관없이 기기 간에 통신이 가능하다.
- (2) 블루투스 방식이 NFC 방식보다 최대 전송 속도가 빠르다. (아래의 그림을 참조한다)
- (4) 최대 통신 가능거리를 가까운 것에서 먼 순서로 나열하면 NFC < Bluetooth < Wi-Fi < LTE 순이다. (아래의 그림을 참조한다)

3. <보기>의 프로세스P1, P2, P3을 시간 할당량(time quantum)이 2인 RR(Round-Robin)알고리즘으로 스케줄링할 때, 평균 응답 시간으로 옳은 것은? (단, 응답시간이란 프로세스의 도착 시간부터 처리가 종료될 때까지의 시간을 말한다. 계산 결과값을 소 수점 둘째자리에서 반올림한다)

<보기>

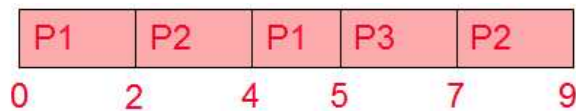
프로세스	도착시간	실행시간
P1	0	3
P2	1	4
P3	3	2

- ① 5.7
- ② 6.0
- ③ 7.0
- ④ 7.3

정답 체크 :

(1)

주어진 조건으로 간트 차트를 그리면 다음과 같다. 0초에 도착한 프로세스는 P1이므로 먼저 시간 할당량이 2만큼 수행한다. 시점 2에서 다음에 수행할 프로세스는 시점 1에 P2가 도착하여 순환 큐에 대기하고 있으므로 P2를 수행한다. 이때 순환 큐(circular queue)에 P2 다음에 스케줄링을 할 프로세스를 선택해야 하는데, 아직 P3가 도착하지 않았으므로(현재 시점은 2이고 P3는 3에 도착) P1을 다시 스케줄링 한다(순환 큐에 넣는다). P2가 끝난 시점 4에서는 P1이 실행됨과 동시에 P3가 이미 시점 3에서 도착했으므로 P3가 이미 순환 큐에 들어가 있고, 그 뒤에 P2를 순환 큐에 넣는다. P1은 전체 실행 시간이 3이므로 1만큼만 수행되고, 순환 큐에 있는 나머지 P3, P2가 차례대로 실행된다.



각각의 응답 시간(반환 시간)을 계산하면 다음과 같다.

$P1 = 5 - 0 = 5$ // P1는 5초에 끝나고 0초에 들어왔으므로 응답시간은 5가 된다.

$P2 = 9 - 1 = 8$ // P2는 9초에 끝나고 1초에 들어왔으므로 응답시간은 8이 된다.

$P3 = 7 - 3 = 4$ // P3는 7초에 끝나고 3초에 들어왔으므로 응답시간은 4가 된다.

이들의 평균 응답 시간(반환 시간)을 구하면 다음과 같다.

평균 응답 시간 = $(5 + 8 + 4) / 3 = 17 / 3 = 5.7$

4. 다음 C프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
int main()
{
    int a=120, b=45;
    while ( a != b ) {
        if ( a > b ) a = a - b;
        else b = b - a;
    }
    printf("%d", a);
}
```

- ① 5
- ② 15
- ③ 20
- ④ 25

정답 체크 :

(2)

초기 조건 : int a=120, b=45; // a는 120에서 시작하고, b는 45에서 시작한다.

첫 번째 루프 : a = 75, b = 45 // a가 b보다 크므로 a는 75가 되고, b는 45로 변함이 없다.

두 번째 루프 : a = 30, b = 45 // a가 b보다 크므로 a는 30이 되고, b는 45로 변함이 없다.

세 번째 루프 : a = 30, b = 15 // b가 a보다 크므로 b는 15가 되고, a는 30으로 변함이 없다.

네 번째 루프 : a = 15, b = 15 // a가 b보다 크므로 a는 15가 되고, b는 15로 변함이 없다.

종료 조건 : a와 b가 15로 같으므로 while을 종료하고 a인 15로 출력한다.

5. <보기>와 같이 수행되는 정렬 알고리즘으로 옳은 것은?

<보기>						
단계 0 :	6	5	8	9	4	2
단계 1 :	6	5	8	2	4	9
단계 2 :	6	5	4	2	8	9
단계 3 :	2	5	4	6	8	9
단계 4 :	2	4	5	6	8	9
단계 5 :	2	4	5	6	8	9

① 셸 정렬 (shell sort)

② 힙 정렬(heap sort)

③ 버블 정렬(bubble sort)

④ 선택 정렬(selection sort)

정답 체크 :

(4) 선택 정렬 :

단계 1 // 정렬되지 않은 숫자 중에 가장 큰 숫자인 9를 선택해서 맨 오른쪽으로 보낸다. (6 5 8 2 4)는 정렬되지 않은 숫자이고 (9)는 정렬된 숫자이다.

단계 2 // 정렬되지 않은 숫자 중에 가장 큰 숫자인 8을 선택해서 9 이전으로 보낸다. (6 5 4 2)는 정렬되지 않은 숫자이고 (8 9)는 정렬된 숫자이다.

단계 3, 단계 4, 단계 5 // 나머지도 마찬가지로 정렬되지 않은 숫자 중에 가장 큰 숫자를 선택해서 정렬된 숫자에 순서대로 정렬한다.

오답 체크 :

(1) 셸 정렬 :

삽입 정렬이 어느 정도 정렬된 리스트에서 대단히 빠른 것에 착안된 정렬 방법이다. 전체 리스트를 일정 간격(gap)의 부분 리스트로 나누고 나뉜 각각의 부분 리스트를 삽입 정렬한다. 간격을 줄이면서 간격이 1이 될 때까지 부분 리스트의 삽입 정렬 과정을 반복한다.

(2) 힙 정렬 :

힙(heap)를 이용하여 정렬하는 것이다. 먼저 정렬해야 할 n개의 요소들을 최대 힙(부모 노드의 키값이 자식 노드의 키값보다 크거나 같은 완전 이진 트리) 혹은 최소 힙(부모 노드의 키값이 자식 노드의 키값보다 작거나 같은 완전 이진 트리)에 삽입한다. 한 번에 하나씩 요소를 힙에서 삭제하여 저장하면 삭제되는 요소들은 값이 증가되는 순서(최소 힙의 경우) 혹은 감소되는 순서(최

대 히프의 경우)가 된다.

(3) 버블 정렬 :

인접한 2개의 데이터를 비교하여 순서대로 되어 있지 않으면 서로 교환한다. 이러한 비교 및 교환 과정을 주어진 숫자들의 왼쪽 끝에서 오른쪽 끝까지 반복한다.

6. 직원(사번, 이름, 입사년도, 부서) 테이블에 대한 SQL 문 중 문법적으로 옳은 것은?

- ① SELECT COUNT(부서) FROM 직원 GROUP 부서;
- ② SELECT * FROM 직원 WHERE 입사년도 IS NULL;
- ③ SELECT 이름, 입사년도 FROM 직원 WHERE 이름 = '최%';
- ④ SELECT 이름, 부서 FROM 직원 WHERE 입사년도 = (2014, 2015);

정답 체크 :

(2) 직원 테이블에서 입사년도가 NULL이 아닌 튜플을 검색하는 것으로 문법적으로 문제가 없다.

오답 체크 :

- (1) “GROUP 부서” 대신에 “GROUP BY 부서”를 사용해야 한다.
- (2) “WHERE 이름 = ‘최%’” 대신에 “WHERE 이름 LIKE ‘최%’”를 사용해야 한다.
- (4) “WHERE 입사년도 = (2014, 2015)” 대신에 “WHERE 입사년도 = 2014 OR 입사년도 = 2015”를 사용해야 한다.

7. <보기>와 같은 특성을 갖는 하드 디스크의 최대 저장 용량은?

<보기>

- 실린더 (cylinder) 개수: 32,768 개
- 면 (surface) 개수: 4개
- 트랙 (track) 당 섹터 (sector) 개수: 256 개
- 섹터크기 (sector size): 512 bytes

- ① 4GB
- ② 16GB
- ③ 64GB
- ④ 1TB

정답 체크 :

용량은 다음과 같이 계산된다.

용량 = 면 개수 x 트랙 개수 x 섹터 개수 x 섹터 크기

하드 디스크는 원형 원판으로 구성되고, 원형 원판은 면(단면 혹은 양면)을 가진다. 그러므로 용량 계산에 면 개수가 포함된다. 그리고 면은 여러 개의 원형 트랙을 가지므로 용량 계산에 트랙 개수가 들어간다. 주어진 문제의 조건을 보면 트랙 개수 대신에 실린더 개수가 주어졌는데 실린더는 면들이 가지는 같은 트랙의 모임이므로 실린더 개수는 트랙의 개수로 보아도 무방하다. 마지막으로 각 트랙은 섹터로 분할되므로 용량 계산에 섹터 개수와 섹터 크기가 포함된다.

주어진 조건을 대입하면 다음과 같다.

16GB = 면 개수(4) x 트랙 개수(32K) x 섹터 개수(256) x 섹터 크기(512B)

여기서, 계산을 간단히 하기 위해 32,768은 32K로 표기한다.

8. <보기>의 직원 테이블에서 키(key)와 관련된 설명으로 옳지 않은 것은? (단, 사번과 주민등록번호

호는 각 유일한 값을 갖고, 부서 번호는 부서 테이블을 참조하는 속성이며, 나이가 같은 동명이 인이 존재할 수 있다)

〈보기〉

직원(사번, 이름, 주민등록번호, 주소, 나이, 성별, 부서번호)

- ① 부서번호는 외래키이다.
- ② 사번은 기본키가 될 수 있다.
- ③ (이름, 나이)는 후보키가 될 수 있다.
- ④ 주민등록번호는 대체키가 될 수 있다.

정답 체크 :

(3) 후보키가 되려면 유일성과 최소성을 만족해야 한다. (이름, 나이)로 키를 구성하면 동일한 (이름, 나이)가 존재할 수 있으므로 유일성이 만족되지 않는다. 그러므로 (이름, 나이)는 후보키가 될 수 없다.

오답 체크 :

- (1) 부서번호는 부서 테이블의 기본키를 참조하므로 외래키이다.
- (2) 사번 혹은 주민등록번호는 각 유일한 값을 갖고(유일성), 최소성(유일성을 만족하는 최소한의 키로 구성)을 만족하므로 기본키가 될 수 있다.
- (4) 사번이 기본키라면 주민등록번호는 대체키(기본키가 아닌 후보키)가 될 수 있다.

9. 소프트웨어 테스트에 대한 설명으로 옳지 않은 것은?

- ① 베타(beta)테스트는 고객 사이트에서 사용자에게 의해서 수행된다.
- ② 회귀(regression)테스트는 한 모듈의 수정이 다른 부분에 미치는 영향을 검사한다.
- ③ 화이트 박스(white box)테스트는 모듈의 내부 구현보다는 입력과 출력에 의해 기능을 검사한다.
- ④ 스트레스(stress) 테스트는 비정상적으로 과도한 분량 또는 빈도로 자원을 요청할 때의 영향을 검사한다.

정답 체크 :

(3) 화이트 박스 테스트 : 해당 설명은 블랙 박스 테스트이고, 화이트 박스 테스트는 프로그램 내부 구조의 타당성 여부를 시험하는 방식이다. 즉, 내부 구조를 해석해서 프로그램의 모든 처리 루틴에 대해 테스트한다.

오답 체크 :

- (1) 베타 테스트 : 시장 출시 전 시장의 피드백을 얻기 위한 목적으로 테스트한다. 특정 사용자가 미리 사용하고 문제점 및 오류를 발견하면 개발자에게 알려준다.
- (2) 회귀 테스트 : 한 모듈의 수정이 다른 부분에 영향을 끼칠 수도 있다고 생각하여 수정된 모듈뿐 아니라 관련된 모듈까지 문제가 없는지 테스트이다.
- (4) 스트레스 테스트 : 시스템이나 시스템 구성 요소(component) 또는 소프트웨어 프로그램에 다양한 스트레스를 가할 때에도 안정적으로 작동하는지를 확인하는 테스트이다. 즉, 짧은 시간 동안에 많은 사용자가 동시에 접속(access)하는 것과 같은 스트레스에 시스템이 안정적으로 작동하는지를 확인하는 과정이다.

10. 〈보기〉에 선언된 배열 A의 원소 A[8][7]의 주소를 행 우선(row-major) 순서와 열 우선(column-major) 순서로 각각 바르게 계산한 것은? (단, 첫 번째 원소 A[0][0]의 주소는 1,000이고, 하나의 원소는 1byte를 차지한다.)

〈보기〉

```
char A[20][30];
```

	행 우선 주소	열 우선 주소
①	1,167	1,148
②	1,167	1,218
③	1,247	1,148
④	1,247	1,218

정답 체크 :

(3)

int A[row][col]; // 2차원 배열이 선언되었다고 가정한다.

A[0][0]; // 시작 주소는 A[0][0] 혹은 A이다.

요소크기 = int = 4 bytes; // 요소크기는 int이기 때문에 4바이트이다. 만약, 요소가 int가 아니라 char(1바이트) 혹은 double(8바이트)이면 요소크기에 주의해야 한다.

A[i][j]의 주소 = 시작주소 + (i*col + j)*요소크기; // 행 우선 순서로 저장할 때는 주어진 수식으로 개별 요소의 주소를 계산할 수 있다. (행을 중심으로 배열의 요소를 배치하므로 i*col로 계산된다.)

A[i][j]의 주소 = 시작주소 + (j*row + i)*요소크기; // 열 우선 순서로 저장할 때는 주어진 수식으로 개별 요소의 주소를 계산할 수 있다. (열을 중심으로 배열의 요소를 배치하므로 j*row로 계산된다. 이는 행 우선 순서와 역의 관계를 가진다.)

주어진 조건으로 문제를 풀면 다음과 같다.

char A[20][30]; // 주어진 2차원 배열이다.

시작주소 = 1,000 // 시작 주소이다. (주어진 조건이 없으므로 10진수로 계산한다)

요소크기 = char = 1바이트 // 요소크기는 char이다.

array[8][7]의 주소 = 1,000 + (8*30 + 7)*1 = 1,247; // 행 우선 순위로 계산한 주소값이다.

array[8][7]의 주소 = 1,000 + (7*20 + 8)*1 = 1,148; // 열 우선 순위로 계산한 주소값이다.

Tip! : 행 우선 순서와 열 우선 순서는 역의 관계를 가지므로 행 우선 순서 하나만 이해하고 외우면 된다.

11. 인터넷 주소 체계인 IPv4와 IPv6의 주소 길이와 주소표시 방법을 각각 바르게 나열한 것은?

IPv4	IPv6
① (32비트, 8비트씩 4부분)	(128비트, 16비트씩 8부분)
② (32비트, 8비트씩 4부분)	(128비트, 8비트씩 16부분)
③ (64비트, 16비트씩 4부분)	(256비트, 32비트씩 8부분)
④ (64비트, 16비트씩 4부분)	(256비트, 16비트씩 16부분)

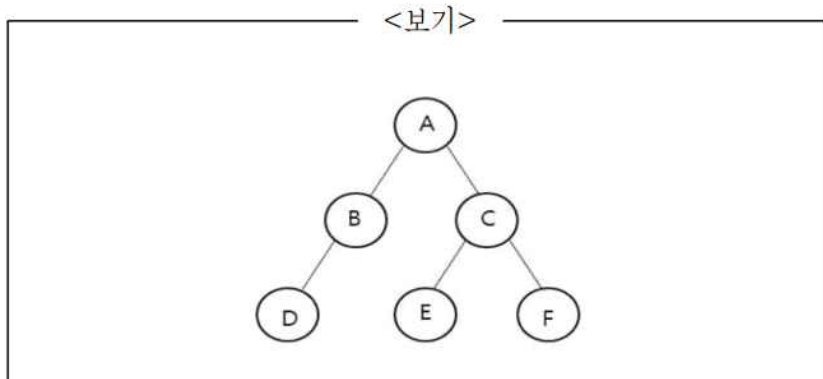
정답 체크 :

(1)

IPv4 : 32비트(4바이트), 8비트씩 4부분, 10진수로 표현

IPv6 : 128비트(16바이트), 16비트씩 8부분, 16진수로 표현

12. 〈보기〉의 이진 트리에 대해 지정된 방법으로 순회한 결과가 옳지 않은 것은?



- ① 중위순회: $D \rightarrow B \rightarrow A \rightarrow E \rightarrow C \rightarrow F$
- ② 레벨순회: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$
- ③ 전위순회: $A \rightarrow B \rightarrow D \rightarrow C \rightarrow E \rightarrow F$
- ④ 후위순회: $D \rightarrow B \rightarrow A \rightarrow E \rightarrow F \rightarrow C$

정답 체크 :

(4)

후위순회 : $D \rightarrow B \rightarrow E \rightarrow F \rightarrow C \rightarrow A$

13. 컴퓨터 시스템의 인터럽트(interrupt)에 대한 설명으로 옳지 않은 것은?

- ① 인터럽트는 입출력 연산, 하드웨어 실패, 프로그램 오류 등에 의해서 발생한다.
- ② 인터럽트 처리 우선순위 결정 방식에는 폴링(polling) 방식과 데이지 체인(daisy-chain) 방식이 있다.
- ③ 인터럽트가 추가된 명령어 사이클은 인출 사이클, 인터럽트 사이클, 실행 사이클 순서로 수행된다.
- ④ 인터럽트가 발생할 경우, 진행 중인 프로그램의 재개(resume)에 필요한 레지스터 문맥(register context)을 저장한다.

정답 체크 :

(3) 인터럽트가 추가된 명령어 사이클은 인출 사이클, 실행 사이클, 인터럽트 사이클 순서로 수행된다. 즉, 명령어 인출 및 실행은 원자성을 가진다. 인출 후 실행을 하지 않은 상태에서 인터럽트를 수행할 수는 없다.

오답 체크 :

- (1) 인터럽트는 동기 인터럽트(프로그램 오류 등)와 비동기 인터럽트(입출력 연산 또는 하드웨어 실패 등)에 의해서 발생한다.
- (2) 인터럽트 처리 우선순위 결정 방식에는 다중 인터럽트 선(Multiple Interrupt Lines), 소프트웨어 폴(Software Poll, 문제에서는 폴링이라고 언급됨), 데이지 체인(Daisy Chain), 버스 중재(Bus Arbitration) 등이 존재한다.
- (4) 인터럽트가 발생할 경우, 진행 중인 프로그램의 재개에 필요한 PC, PSW(Program Status Word, 캐리, 부호, 제로, 오버플로우 등이 발생했는지를 저장) 등을 저장한다. 문제에서는 PW, PSW 등을 통틀어 레지스터 문맥이라는 단어를 사용하였다.

14. 다음은 3년간 연이율 4%로 매월 적립하는 월 복리 정기적금의 만기지급금을 계산한 결과이다.

셀 C2에 들어갈 수식으로 옳은 것은? (단, 만기지급금의 10원 단위 미만은 절사한다)

	A	B	C
1	성명	월적립액	만기지급금
2	김**	₩ 30,000	₩ 1,145,440
3	이**	₩ 50,000	₩ 1,909,070

- ① =ROUNDDOWN(FV(4%, 3*12, -B2), -1)
 ② =ROUNDDOWN(FV(4%, 3*12, -B2), -2)
 ③ =ROUNDDOWN(FV(4%/12, 3*12, -B2), -1)
 ④ =ROUNDDOWN(FV(4%/12, 3*12, -B2), -2)

정답 체크 :

(3)

조건-1 : 3년간 연이율 4%로 매월 적립하는 월 복리 정기적금의 만기지급금 FV(4%/12, 3*12, -B2)

조건-2 : 만기지급금의 10원 단위 미만은 절사 ROUNDDOWN(만기지급금, -1)

오답 체크 :

- (1) FV(4%, 3*12, -B2) : 3년간 월이율 4%로 매월 적립하는 월 복리 정기적금의 만기지급금(조건-1에 맞지 않는다)
 (2), (4) ROUNDDOWN(만기지급금, -2) : 만기지급금의 100원 단위 미만은 절사(조건-2에 맞지 않는다)

15. 정점의 개수가 n 인 연결그래프로부터 생성 가능한 신장트리(spamming tree)의 간선의 개수는?

- ① $n-1$
 ② n
 ③ $\frac{n(n-1)}{2}$
 ④ n^2

정답 체크 :

(1)

정점의 개수가 n 인 연결그래프로부터 생성 가능한 신장트리의 간선의 개수는 $n-1$ 개이다.

16. <보기>는 관계형 데이터베이스의 정규화 작업을 설명한 것이다. 제1정규형, 제2정규형, 제3정규형, BCNF를 생성하는 정규화 작업을 순서대로 나열한 것은?

<보기>

- ㄱ. 결정자가 후보키가 아닌 함수 종속성을 제거한다.
 ㄴ. 부분 함수 종속성을 제거한다.
 ㄷ. 속성을 원자값만 갑도록 분해한다.
 ㄹ. 이행적 함수 종속성을 제거한다.

- ① ㄱ → ㄴ → ㄷ → ㄹ

- ② $\neg \rightarrow C \rightarrow R \rightarrow L$
- ③ $C \rightarrow \neg \rightarrow L \rightarrow R$
- ④ $C \rightarrow L \rightarrow R \rightarrow \neg$

정답 체크 :

(4)

C. 제1정규형 : 모든 속성의 도메인이 원자 값으로 구성된다.

L. 제2정규형 : 모든 속성이 기본키에 완전 함수 종속된다.

R. 제3정규형 : 모든 속성이 기본키에 이행적 함수 종속이 아니다.

¬. BCNF : 모든 결정자가 후보키이다.

제4정규형 : 함수 종속이 아닌 다치 종속을 제거한다.

제5정규형 : 후보키를 통하지 않는 조인 종속을 제거한다.

17. 다음 Java 프로그램의 실행 결과로 옳은 것은?

```
class Division {
    public static void main(String[] args) {
        int a, b, result;
        a = 3; b = 0;
        try {
            result = a / b;
            System.out.print("A");
        }
        catch (ArithmeticException e) {
            System.out.print("B");
        }
        finally {
            System.out.print("C");
        }
        System.out.print("D");
    }
}
```

- ① ACD
- ② BCD
- ③ ABCD
- ④ BACD

정답 체크 :

(2)

try 블록 : $a / b = 3 / 0$ 을 수행하면 오류가 발생되어 “A”를 출력하지 않고 catch 블록을 수행한다.

catch 블록 : “B”를 출력한다. 이후에 오류 발생 여부에 상관없이 finally 블록을 수행한다.

finally 블록 : “C”를 출력한다. 나머지 코드를 실행한다.

나머지 코드 : “D”를 출력한다.

이들을 종합하면 “BCD”를 출력한다.

Tip! : try, catch, finally의 실행 순서를 정리하면 다음과 같다.

1. try 블록이 성공하면 : try 문을 실행한다, finally 문을 실행한다, 나머지 코드를 실행한다.
2. try블록이 실패하면 : try문에서 예외 발생하는 위치까지 실행한다, catch 블록을 실행한다, finally 문을 실행한다, 나머지 코드를 실행한다.
3. try문 또는 catch 문에 return이 있는 경우 : try 또는 catch문을 실행한다, finally 문을 실행한다, return 한다.

18. <보기>는 공개키 암호 방식을 전자 서명(digital signature)에 적용하여 A가 B에게 메시지를 전송하는 과정에 대한 설명이다. ㉠, ㉡에 들어갈 내용으로 옳은 것은?

<보기>

- (1) A와 B는 개인키와 공개키 쌍을 각각 생성한다.
- (2) A는 (㉠)를 사용하여 암호화한 메시지를 B에게 전송한다.
- (3) B는 (㉡)를 사용하여 수신된 메시지를 해독한다.

㉠

㉡

- | | | |
|---|--------|--------|
| ① | A의 개인키 | A의 공개키 |
| ② | A의 개인키 | B의 공개키 |
| ③ | A의 공개키 | B의 개인키 |
| ④ | B의 공개키 | B의 개인키 |

정답 체크 :

(1) 개인키 암호화(디지털 서명) 방식을 나타낸다. 보내는 사람(A)의 개인키로 암호화를 수행하고, 보내는 사람(A)의 공개키로 복호화를 수행한다.

오답 체크 :

(2), (3) 해당 방식은 존재하지 않는다.

(4) 공개키 암호화 방식을 나타낸다. 받는 사람(B)의 공개키로 암호화를 수행하고, 받는 사람(B)의 개인키로 복호화를 수행한다.

Tip! : 전자 서명보다는 디지털 서명이라는 표현이 맞다. 전자 서명은 전자적으로 수행하는 모든 서명을 통칭하는 단어이고, 디지털 서명은 개인키 암호화 방식을 지칭한다.

19. 프로그래밍 언어에 대한 설명으로 옳지 않은 것은?

- ① Objective-C, Java, C#은 객체지향 언어이다.
- ② Python은 정적 타이핑을 지원하는 컴파일러 방식의 언어이다.
- ③ ASP, JSP, PHP는 서버 측에서 실행되는 스크립트 언어이다.
- ④ XML은 전자문서를 표현하는 확장가능한 표준 마크업 언어이다.

정답 체크 :

(2) Python은 인터프리터 방식의 언어이다. C, C++은 컴파일러 방식의 언어이고, 자바는 하이브리드 방식(컴파일러를 수행 후 인터프리터를 수행)의 언어이다. 그리고 파이썬은 동적 타이핑을 지원한다.

오답 체크 :

(1) 객체지향 언어에는 시물라 67, 스몰토크, 비주얼 베이직 닷넷, 오브젝티브-C, C++, C#, 자바, 객체지향 파스칼, 델파이, 파이썬, 펄, 루비, 액션스크립트, ASP, 스위프트 등이 있다.

(3) ASP, JSP, PHP는 서버 측에서 실행되는 스크립트 언어(server-side scripting)이고, 자바스크립트는 클라이언트 측에서 실행되는 스크립트 언어(client-side scripting)이다.

(4) XML(extensible markup language, 확장 가능한 마크업 언어)은 HTML보다 홈페이지 구축 기능, 검색 기능 등이 향상되었고 클라이언트 시스템의 복잡한 데이터 처리를 쉽게 한다. 또한 인터넷 사용자가 웹에 추가할 내용을 작성, 관리하기에 쉽게 되어 있다. SGML(전자 문서를 나타내기 위한 표준 언어, HTML의 아버지, XML의 형제)의 부분 집합으로 태그를 정의할 수 있다.

20. <보기>의 설명에 해당하는 기술로 가장 적절한 것은?

<보기>

- 서비스 모델은 IaaS, PaaS, SaaS로 구분한다.
- 필요한 만큼 자원을 임대하여 사용할 수 있다.
- 가상화 기술, 서비스 프로비저닝(Provisioning) 기술, 과금 체계 등을 필요로 한다.

- ① 빅데이터(bigdata)
- ② 딥 러닝(deep learning)
- ③ 사물 인터넷(internet of things)
- ④ 클라우드 컴퓨팅(cloud computing)

정답 체크 :

(4) 클라우드 컴퓨팅 : 각종 소프트웨어와 데이터를 인터넷과 연결된 중앙 컴퓨터에 저장한다. 필요할 때마다 컴퓨터나 스마트폰 같은 단말기로 접속하여 데이터를 내려 받아 사용하고 다시 업로드하는 방식이다. 사용한 만큼 돈을 지불하는 유틸리티 컴퓨팅과 비슷한 개념이다. 서비스 유형에는 이외에도 NaaS(네트워크), SECaaS(보안) 등이 존재한다.

오답 체크 :

(1) 빅 데이터 : 굉장히 많은 양의 데이터에서 빠르게 정보를 추출 및 분석하여 가치 있는 정보를 발견하는 기술이다. 다루는 데이터는 어마어마한 대량의 정형(structured, 컴퓨터가 이미 다루고 있는 데이터) 또는 비정형(unstructured, 자연어, 이미지, 동영상) 데이터이다.

(2) 딥 러닝 : 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기할 수 있다. 어떠한 데이터가 있을 때 이를 컴퓨터가 알아 들을 수 있는 형태로 표현하고 이를 학습에 적용하기 위해 많은 연구가 진행되고 있다.

(3) 사물 인터넷 : 각종 사물에 컴퓨터 칩과 통신 기능을 내장해 인터넷에 연결하는 기술을 의미한다. 사물끼리 정보를 주고받기 위하여 인터넷으로 연결되어 있는 사물 공간 연결망이다.