

2018-지방직-컴퓨터일반-B형-해설-곽후근

1. 선형 자료구조에 해당하지 않는 것은?

- ① 큐
- ② 스택
- ③ 이진 트리
- ④ 단순 연결 리스트

정답 체크 :

(3) 이진 트리 : 자식 노드를 2개 이하만 가지는 트리 구조로 1:2로 연결되어 있어 비선형 자료구조의 형태를 가진다.

오답 체크 :

- (1) 큐 : FIFO 구조로 1:1로 연결되어 있어 선형 자료구조의 형태를 가진다.
 - (2) 스택 : LIFO 구조로 1:1로 연결되어 있어 선형 자료구조의 형태를 가진다.
 - (4) 단순 연결 리스트 : 체인(Chain) 구조로 1:1로 연결되어 있어 선형 자료구조의 형태를 가진다.
- Tip! : 정리하면 1:1 연결 구조이면 선형이고, 1:N 연결 구조이면 비선형이다.

2. 비트열(bit string) A를 2의 보수로 표현된 부호 있는(signed) 2진 정수로 해석한 값은 -5이다. 비트열 A를 1의 보수로 표현된 부호 있는 2진 정수로 해석한 값은?

- ① -4
- ② -5
- ③ -6
- ④ -7

정답 체크 :

5 : 0101 // 5의 이진수는 0101이다.

-5 : 1010 (1의 보수) // 5의 1의 보수는 1010이다.

-5 : 1011 (2의 보수) // 5의 2의 보수는 1011이다. (2의 보수로 해석한 값)

주어진 조건 : 1011 // 만약 이것을 2의 보수가 아닌 1의 보수로 해석하면 값이 얼마일까? 1011을 1의 보수를 취하면 0100이 나오고 이는 4이므로 결국 -4가 1의 보수로 해석한 값이다.

3. 직원 테이블 emp의 모든 레코드를 근무연수 wyear에 대해서는 내림차순으로, 동일 근무연수에 대해서는 나이 age의 오름차순으로 정렬한 결과를 얻기 위한 SQL 질의문은?

- ① SELECT * FROM emp ORDER BY age, wyear DESC;
- ② SELECT * FROM emp ORDER BY age ASC, wyear;
- ③ SELECT * FROM emp ORDER BY wyear DESC, age;
- ④ SELECT * FROM emp ORDER BY wyear, age ASC;

정답 체크 :

(3) ORDER BY wyear DESC, age;

순서상으로 wyear DESC를 적용하고, 해당 결과에 age를 적용한다.

wyear DESC는 wyear에 대해서 내림차순(DESC) 정렬을 의미한다.

age에 차순 조건이 없다면 디폴트(default)로 오름차순(ASC) 정렬을 적용한다. (오름차순이

default로 생략)

4. 다음에서 설명하는 디스크 스케줄링은?

디스크 헤드가 한쪽 방향으로 트랙의 끝까지 이동하면서 만나는 요청을 모두 처리한다. 트랙의 끝에 도달하면 반대 방향으로 이동하면서 만나는 요청을 모두 처리한다. 이러한 방식으로 헤드가 디스크 양쪽을 계속 왕복하면서 남은 요청을 처리한다.

- ① 선입 선처리(FCFS) 스케줄링
- ② 최소 탐색 시간 우선(SSTF) 스케줄링
- ③ 스캔(SCAN) 스케줄링
- ④ 라운드 로빈(RR) 스케줄링

정답 체크 :

(3) SCAN(스캔) : 입출력 헤드가 디스크의 한쪽 끝에서 다른 끝으로 이동하며, 한쪽 끝에 도달했을 때는 역방향으로 이동하면서 요청한 트랙을 처리한다.

오답 체크 :

(1) FCFS(선입 선처리) : 요청이 도착한 순서에 따라 처리하는 가장 간단한 스케줄링 알고리즘이다. 공정성을 보장할 수 있다.

(2) SSTF(최소 탐색 시간 우선) : 디스크 요청을 처리하려고 헤드가 먼 곳까지 이동하기 전에, 현재 헤드 위치에 가까운 모든 요구를 먼저 처리하는 방법으로 공정성을 보장할 수 없다.

(4) RR(라운드 로빈) : 디스크 스케줄링이 아니라 프로세스 스케줄링이다.

Tip! : 프로세스 스케줄링을 묻는 질문에 디스크 스케줄링이 나오고, 디스크 스케줄링을 묻는 질문에 프로세스 스케줄링이 나온다. 그러므로 서로를 구별하면 오답을 빠르게 제거할 수 있다.

5. 정보량의 크기가 작은 것에서 큰 순서대로 바르게 나열한 것은?

(단, PB, TB, ZB, EB는 각각 petabyte, terabyte, zettabyte, exabyte이다)

- ① 1 PB, 1 TB, 1 ZB, 1 EB
- ② 1 PB, 1 TB, 1 EB, 1 ZB
- ③ 1 TB, 1 PB, 1 ZB, 1 EB
- ④ 1 TB, 1 PB, 1 EB, 1 ZB

정답 체크 :

(4)

정보량의 크기를 테이블로 정리하면 다음과 같다.

10 ⁿ	접두어	기호	배수	십진수
10 ²⁴	요타 (yotta)	Y	자	1 000 000 000 000 000 000 000 000
10 ²¹	제타 (zetta)	Z	십해	1 000 000 000 000 000 000 000
10 ¹⁸	엑사 (exa)	E	백경	1 000 000 000 000 000 000
10 ¹⁵	페타 (peta)	P	천조	1 000 000 000 000 000
10 ¹²	테라 (tera)	T	조	1 000 000 000 000
10 ⁹	기가 (giga)	G	십억	1 000 000 000
10 ⁶	메가 (mega)	M	백만	1 000 000
10 ³	킬로 (kilo)	k	천	1 000
10 ²	헥토 (hecto)	h	백	100
10 ¹	데카 (deca)	da	십	10
10 ⁰			일	1
10 ⁻¹	데시 (deci)	d	십분의 일	0.1
10 ⁻²	센티 (centi)	c	백분의 일	0.01
10 ⁻³	밀리 (milli)	m	천분의 일	0.001
10 ⁻⁶	마이크로 (micro)	μ	백만분의 일	0.000 001
10 ⁻⁹	나노 (nano)	n	십억분의 일	0.000 000 001
10 ⁻¹²	피코 (pico)	p	일조분의 일	0.000 000 000 001
10 ⁻¹⁵	펨토 (femto)	f	천조분의 일	0.000 000 000 000 001
10 ⁻¹⁸	아토 (atto)	a	백경분의 일	0.000 000 000 000 000 001
10 ⁻²¹	젠토 (zepto)	z	십해분의 일	0.000 000 000 000 000 000 001
10 ⁻²⁴	욕토 (yocto)	y	일자분의 일	0.000 000 000 000 000 000 000 001

Tip! : 단위를 묻는 질문이 많이 나오므로 해당 단위를 위아래로 숙지해두는 것이 좋다.

6. 다음에서 설명하는 RAID 레벨은?

- 블록 단위 스트라이핑(striping)을 통해 데이터를 여러 디스크에 분산 저장한다.
- 패리티를 패리티 전용 디스크에 저장한다.

- ① RAID 레벨 1
- ② RAID 레벨 2
- ③ RAID 레벨 4
- ④ RAID 레벨 5

정답 체크 :

(3) RAID 4 : RAID 3은 바이트 단위로 데이터를 분할하고 패리티 정보를 계산하지만, RAID 4는 미리 정해진 블록 단위로 데이터를 분할하고 패리티를 계산한다. RAID 3과 마찬가지로 오직 1개의 별도 볼륨으로 패리티 정보를 저장한다.

오답 체크 :

- (1) RAID 1 : 여분의 디스크가 포함되지 않지만 동일한 RAID 볼륨을 추가적으로 구성된다. 추가된 볼륨이 원래의 볼륨과 동일하기 때문에 미러링(mirroring) 모드라고 한다.
- (2) RAID 2 : RAID 0의 병렬 접속 기술을 사용하며, 여분의 디스크를 추가하여 오류검사를 통해 신뢰성을 높인 방법이다. 패리티 정보는 해밍 코드(Hamming Code)를 사용한다.
- (4) RAID 5 : 패리티 비트를 저장하는 볼륨을 별도로 설치(레벨 4)하지 않고, 데이터를 저장하는 볼륨에 패리티 비트를 분산하여 저장한다.

7. 소프트웨어 개발을 위한 애자일 기법에 대한 설명으로 옳은 것은?

- ① 소프트웨어를 점증적으로 개발한다.
- ② 작동하는 소프트웨어보다 포괄적인 문서에 더 가치를 둔다.
- ③ 계획에 따라 단계적으로 개발하므로 변화에 대응하기 어렵다.
- ④ 고객과의 협업보다 계약 협상을 더 중요시한다.

정답 체크 :

(1) 소프트웨어를 점증적(점진적, incremental)이고 반복적(iterative)인 개발방식을 들 수 있다. 점증적 개발이란 하나의 완전한 아이디어를 구상하고 한번에 하나씩 구현해 나가는 것이라면, 반복적 개발은 모호한 아이디어에서 출발하여 점차 이를 구체화하고 검증하여 완성도를 높여가는 것이다.

오답 체크 :

- (2) 포괄적인 문서보다 작동하는 소프트웨어에 더 가치를 둔다.
- (3) 소프트웨어를 점증적이고 반복적으로 개발하기 때문에 변화에 대응하기 용이하다.
- (4) 계약 협상보다 고객과의 협업을 더 중요시한다.

8. 2-way 집합 연관 사상(set-associative mapping) 방식을 사용하는 캐시 기억장치를 가진 컴퓨터가 있다. 캐시 기억장치 접근(access)을 위해 주기억장치 주소가 다음 세 필드(field)로 구분된다면, 캐시 기억장치의 총 라인(line) 개수는?

태그(tag) 필드	세트(set) 필드	오프셋(offset) 필드
8비트	9비트	7비트

- ① 128개
- ② 256개
- ③ 512개
- ④ 1,024개

해설)

정답 체크 :

(4)

Set = 9bit = 2^9 // 세트 필드가 9비트이므로 512($=2^9$)개의 라인이 존재한다.

2-way = $2^9 + 2^9 = 2^{10}$ // 세트 내에 2개의 라인이 존재하므로 총 1024($=2^{10}$)개의 라인이 존재한다.

Tip! : 해당 문제를 확장하면 N-way까지 계산이 가능하다.

9. 다음 C 프로그램의 출력 결과는?

```
#include <stdio.h>
#define SIZE 3
void func(int *m, int *a, int b);
int main(void)
{
    int num[SIZE] = {1, 3, 6};
```

```

int a = 10, b = 30;
func(num, &a, b);
printf("a = %d, b = %d\n", a, b);
return 0;
}
void func(int *m, int *x, int y)
{
    int i = 0, n = 0;
    y = *x;
    n = *(m + 1) + (*m + 2);
    *x = ++n;
}

```

- ① a = 7, b = 10
- ② a = 7, b = 30
- ③ a = 10, b = 10
- ④ a = 10, b = 30

정답 체크 :

(2)

int a=10, b=30;

func(num, &a, b);

b // b는 call-by-value(값에 의한 호출)이므로 호출 후에도 값이 변하지 않는다. 그러므로 출력될 최종 b의 값은 30이다.

&a // a는 call-by-reference(주소에 의한 호출)이므로 호출 후에 값이 변한다. 이때, 함수 내에서 값이 변해 7이 될 수도 있고, 변한 값이 다시 10이 될 수도 있다. 그러므로 아래의 코드 해석 과정을 거쳐야 한다.

num; // num의 시작 주소가 func 함수 내에서 m에게 할당된다. 이 둘은 운명을 같이한다.

n = *(m + 1) + (*m + 2); // m은 num 배열의 시작주소이므로 m+1은 배열의 2번째 요소의 주소를 의미한다. 그러므로 *(m+1)은 2번째 배열 요소의 값 3이 된다. 그리고 *m은 배열의 첫 번째 요소의 값이므로 1이 된다. 이를 계산하면 6 = 3 + (1 + 2)이 된다.

*x = ++n; // n이 6이고 먼저 증가한 후 대입되므로 x(a와 운명을 같이하는 포인터)의 값은 7이 된다.

10. TCP 프로토콜에 대한 설명으로 옳지 않은 것은?

- ① 전이중(full duplex) 연결 서비스를 제공한다.
- ② 3-way 핸드셰이크(handshake)를 사용하여 연결을 설정한다.
- ③ 흐름제어(flow control)와 혼잡제어(congestion control)를 제공한다.
- ④ TCP 세그먼트(segment)에서 검사합(checksum)의 포함은 선택 사항이다.

정답 체크 :

(4) TCP에서 검사합(checksum)은 필수이다.

오답 체크 :

(1) TCP는 동시에 송수신자가 패킷을 주고받을 수 있다.

- (2) 연결 설정에 3-way handshake를 사용하고, 연결 해제에 4-way handshake를 사용한다.
 (3) 흐름 제어(수신자가 더 이상 패킷을 받을 수 없을 때 제어)와 혼잡 제어(네트워크 중간의 라우터 등이 더 이상 패킷을 받을 수 없을 때 제어)를 제공한다.

11. 사용자가 인터넷 등을 통해 하드웨어, 소프트웨어 등의 컴퓨팅 자원을 원격으로 필요한 만큼 빌려서 사용하는 방식의 서비스 기술은?

- ① 클라우드 컴퓨팅
- ② 유비쿼터스 센서 네트워크
- ③ 웨어러블 컴퓨터
- ④ 소셜 네트워크

정답 체크 :

(1) 클라우드 컴퓨팅 : 각종 소프트웨어와 데이터를 인터넷과 연결된 중앙 컴퓨터에 저장한다. 필요할 때마다 컴퓨터나 스마트폰 같은 단말기로 접속하여 데이터를 내려받아 사용하고 다시 업로드하는 방식이다. 클라우드 컴퓨팅은 유틸리티 컴퓨팅(유틸리티(수도, 전기 등)처럼 사용한 만큼 돈을 내자는 개념)의 개념을 내포한다.

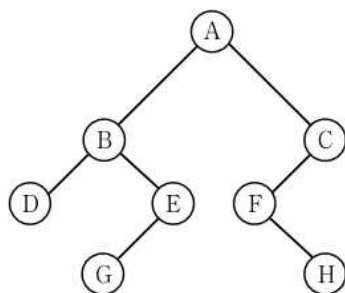
오답 체크 :

(2) 유비쿼터스 센서 네트워크(USN) : 필요로 하는 모든 곳에 수 많은 센서 노드들을 부착하여 자율적으로 정보를 수집, 관리 및 제어하는 시스템이다. 즉 물리 공간에 빛, 소리, 온도, 움직임 같은 물리적 데이터를 센서 노드에서 감지하고 측정하여 중앙의 기본 노드로 전달하는 구조를 가진 네트워크이다. 강수량 측정이나 산불 감시 등에 쓰일 수 있다.

(3) 웨어러블 컴퓨터 : 궁극적으로는 사용자가 거부감 없이 신체의 일부처럼 항상 착용하고 사용할 수 있으며 인간의 능력을 보완하거나 배가시키는 것이 목표이다. 기본 기능들로는 언제 어디서나(항시성), 쉽게 사용할 수 있고(편의성), 착용하여 사용하기에 편하며(착용감), 안전하고 보기 좋은(안정성/사회성) 특성이 요구된다.

(4) 소셜 네트워크 : 사용자 간의 자유로운 의사소통과 정보 공유, 그리고 인맥 확대 등을 통해 사회적 관계를 생성하고 강화해주는 온라인 플랫폼을 의미한다. 대표적인 예로, 페이스북, 인스타그램 등을 들 수 있다.

12. 다음 이진 트리의 노드를 전위 순회(preorder traversal)할 경우의 방문 순서는?



- ① A-B-C-D-E-F-G-H
- ② A-B-D-E-G-C-F-H
- ③ D-B-G-E-A-F-H-C
- ④ D-G-E-B-H-F-C-A

정답 체크 :

(2) 전위 순회(preorder traversal)

오답 체크 :

(1) 레벨 순회(level traversal)

(3) 중위 순회(inorder traversal)

(4) 후위 순회(postorder traversal)

13. 전자 메일의 송신 또는 수신을 목적으로 하는 응용 계층 프로토콜에 해당하지 않는 것은?

① IMAP

② POP3

③ SMTP

④ SNMP

정답 체크 :

(4) SNMP : IP 네트워크상의 장치로부터 정보를 수집 및 관리하며, 또한 정보를 수정하여 장치의 동작을 변경하는 데에 사용되는 인터넷 표준 프로토콜이다. SNMP를 지원하는 대표적인 장치에는 라우터, 스위치, 서버, 워크스테이션, 프린터, 모뎀 랙 등이 포함된다. 예를 들어, 라우터의 상태를 보거나 아니면 어떤 설정을 필요로 할 때 SNMP를 이용할 수 있다.

오답 체크 :

(1) IMAP : 메일 서버에서 메일 클라이언트가 메일을 내려 받을 때 사용하는 프로토콜이다. 메일 사본은 메일 서버에 남긴다(백업용).

(2) POP3 : 메일 서버에서 메일 클라이언트가 메일을 내려 받을 때 사용하는 프로토콜이다. 메일 사본을 메일 서버에 남기지 않는다.

(3) SMTP : 메일 클라이언트가 메일 서버로 메일을 보낼 때 또는 메일 서버가 메일 서버에게 메일을 보내거나 받을 때 사용하는 프로토콜이다.

14. 모바일 기기에 특화된 운영체제에 해당하지 않는 것은?

① iOS

② Android

③ Symbian

④ Solaris

정답 체크 :

(4) Solaris : PC 용 유닉스 운영체제이다.

오답 체크 :

(1) iOS : 아이폰의 운영체제이다.

(2) Android : 대표적으로 삼성 갤럭시폰의 운영체제이다.

(3) Symbian : 심비안에서 개발한 모바일 기기용 운영체제이다.

Tip! : 모바일 운영체제는 이외에도 블랙베리(블랙베리), 윈도우폰(MS), 바다(삼성), 타이젠(삼성) 등이 존재한다.

15. 다음 표는 단일 중앙처리장치에 진입한 프로세스의 도착 시간과 그 프로세스를 처리하는 데 필요한 실행 시간을 나타낸 것이다. 비선점 SJF(Shortest Job First) 스케줄링 알고리즘 을 사용한

경우, P1, P2, P3, P4 프로세스 4개의 평균 대기 시간은? (단, 프로세스 간 문맥 교환에 따른 오버헤드는 무시하며, 주어진 4개의 프로세스 외에 처리할 다른 프로세스는 없다고 가정한다)

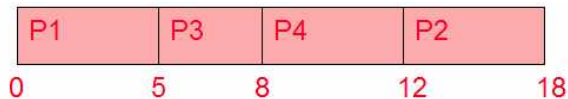
프로세스	도착 시간(ms)	실행 시간(ms)
P1	0	5
P2	3	6
P3	4	3
P4	6	4

- ① 3 ms
- ② 3.5 ms
- ③ 4 ms
- ④ 4.5 ms

정답 체크 :

(1)

주어진 조건으로 간트 차트를 그리면 다음과 같다. 비선점이므로 P1이 제일 먼저 실행되고, P1 종료 후에 도착 한 프로세스(P2, P3) 중 실행 시간이 작은 것은 P3이므로 P3가 실행된다. 그리고 P3 종료 후에 도착 한 프로세스(P2, P4) 중 실행 시간이 작은 것은 P4이므로 P4가 실행된다. P4가 끝나고 마지막 프로세스인 P2가 실행된다.



간트 차트를 기준으로 평균 대기 시간을 구하면 다음과 같다.

P1 = 0 // 0ms간 대기했고 0ms에 도착했으므로 대기 시간은 0ms이다.

P2 = 12 - 3 = 9 // 12ms간 대기했고 3ms에 도착했으므로 대기 시간은 9ms이다.

P3 = 5 - 4 = 1 // 5ms간 대기했고 4ms에 도착했으므로 대기 시간은 1ms이다.

P4 = 8 - 6 = 2 // 8ms간 대기했고 6ms에 도착했으므로 대기 시간은 2ms이다.

평균 대기 시간 = (0 + 9 + 1 + 2) / 4 = 3ms

16. IPv4와 IPv6에 대한 설명으로 옳지 않은 것은?

- ① IPv4는 비연결형 프로토콜이다.
- ② IPv6 주소의 비트 수는 IPv4 주소 비트 수의 2 배이다.
- ③ IPv6는 애니캐스트(anycast) 주소를 지원한다.
- ④ IPv6는 IPv4 네트워크와의 호환성을 위한 방법을 제공한다.

정답 체크 :

(2) IPv6 주소의 비트 수(128비트)는 IPv4 주소의 비트 수(32비트)의 4배이다.

오답 체크 :

(1) IPv4는 비연결형, 비신뢰성 프로토콜이다.

(3) IPv6는 애니캐스트 주소를 지원한다. 애니캐스트란 멀티캐스트와 마찬가지로 여러 인터페이스를 식별한다. 애니캐스트 주소로 지정된 패킷은 적절한 멀티캐스트 라우팅 토폴로지를 통해 주소로 식별되는 가장 가까운 인터페이스인 단일 인터페이스로 배달된다(멀티캐스트는 다중 인터페이스로 배달).

(4) IPv6는 IPv4 네트워크와의 호환성을 위해 3가지 방법(IPv4/IPv6 듀얼 스택 방식, 터널링 방식,

주소 변환 방식)을 제공한다.

17. 순차논리회로(sequential logic circuit)에 해당하는 것은?

- ① 3-to-8 디코더(decoder)
- ② 전가산기(full adder)
- ③ 동기식 카운터(synchronous counter)
- ④ 4-to-1 멀티플렉서(multiplexer)

정답 체크 :

순차논리회로와 조합논리회로는 다음과 같이 구분한다.

순차논리회로 : 현재의 입력과 과거의 출력이 현재의 출력을 결정한다. 즉, 과거의 출력에 영향을 받는다.

조합논리회로 : 현재의 입력이 현재의 출력을 결정한다. 즉, 과거의 출력에 영향을 받지 않는다. 문제의 주어진 조건을 보면 현재의 출력이 과거의 출력에 영향을 받지 않으므로 조합논리회로이다.

(3) 동기식 카운터 : 카운터의 경우 현재값이 10이 되려면 과거값 9와 현재값 1이 있어야 하므로 순차논리회로이다. 여기서 동기식이란 카운터를 구성하는 플립플롭 모두에 클럭이 인가됨을 의미한다.

오답 체크 :

(1) 디코더 : 현재의 입력(3)이 현재의 출력(8)을 결정하므로 조합논리회로이다.

(2) 전가산기 : 현재의 입력(두개의 입력과 전단계의 캐리)이 현재의 출력(합의 결과와 캐리)을 결정하므로 조합논리회로이다.

(4) 멀티플렉서 : 현재의 입력(4)이 현재의 출력(1)을 결정하므로 조합논리회로이다.

18. 클럭(clock) 주파수가 2 GHz인 중앙처리장치를 사용하는 컴퓨터 A에서 프로그램 P를 실행하는데 10초가 소요된다. 클럭 주파수가 더 높은 중앙처리장치를 사용하는 컴퓨터 B에서 프로그램 P를 실행하면, 소요되는 클럭 사이클 수는 컴퓨터 A에 대비하여 1.5 배로 증가하나 실행 시간은 6 초로 감소한다. 컴퓨터 B에 사용된 중앙처리장치의 클럭 주파수는? (단, 실행 시간은 중앙처리장치의 실행 시간만을 고려한 것이며 프로그램 P만 실행하여 측정된다)

- ① 3 GHz
- ② 4 GHz
- ③ 5 GHz
- ④ 6 GHz

정답 체크 :

(3)

프로그램의 실행 시간은 다음과 같이 구할 수 있다.

프로그램의 CPU 실행시간 = 프로그램의 CPU 클럭 사이클 수 x 클럭 사이클 시간 = 프로그램의 CPU 클럭 사이클 수 / 클럭 속도

컴퓨터 A에 주어진 조건으로 프로그램의 CPU 클럭 사이클 수를 계산하면 다음과 같다.

$10 = \text{프로그램의 CPU 클럭 사이클 수} / 2\text{GHz}$, 프로그램의 CPU 클럭 사이클 수 = 20G

컴퓨터 B에 주어진 조건으로 클럭 속도(클럭 주파수)를 구하면 다음과 같다.

$6 = (20G * 1.5) / \text{클럭 속도}$, 클럭 속도 = 5GHz

Tip! : 프로그램의 CPU 클럭 사이클 수는 다음과 같이 구할 수 있다.

프로그램의 CPU 클럭 사이클 수 = 명령어 수 x 명령어당 평균 클럭 사이클 수(CPI)

이를 프로그램의 실행 시간에 적용하면 다음과 같다.

프로그램의 CPU 실행 시간 = 명령어 수 x CPI x 클럭 사이클 시간

프로그램의 CPU 실행 시간 = 명령어 수 x CPI / 클럭 속도

19. 다음 Java 프로그램의 출력 결과는?

```
public class Foo {
    public static void main(String [] args) {
        int i, j , k ;
        for (i = 1, j = 1, k = 0; i < 5; i ++ ) {
            if ((i % 2) == 0)
                continue;
            k += i * j++;
        }
        System.out.println(k);
    }
}
```

- ① 5
- ② 7
- ③ 11
- ④ 15

정답 체크 :

(2)

for 문 조건에 의해 i는 1부터 4까지 증가하는데, 다음의 조건에 의해 1과 3일 때만 주어진 수식이 동작한다.

if (i % 2) == 0) continue; // i를 2로 나누어서 나머지가 0이면 아래 수식을 수행하지 말고 다시 for 문으로 돌아간다. 그러므로 2, 4는 수식 계산에서 배제된다.

i = 1; k = k + (i * j++) = 0 + (1*1); // 수식 계산 후 k = 1, j = 2가 된다.

i = 3; k = k + (i * j++) = 1 + (3*2); // 수식 계산 후 k = 7, j = 3이 된다.

Tip! : 자바의 형태이지만 결국 C언어 문제이다.

20. 다음 카르노 맵(Karnaugh map)으로 표현된 부울 함수 F(A, B, C, D)를 곱의 합(sum of products) 형태로 최소화(minimization)한 결과는? (단, X는 무관(don't care) 조건을 나타낸다)

		AB			
		00	01	11	10
CD	00	0	1	X	1
	01	0	X	0	0
	11	X	1	0	0
	10	0	1	X	1

- ① $F(A, B, C, D) = AD' + BC'D' + A'BC$
- ② $F(A, B, C, D) = AB'D' + BC'D' + A'BC$
- ③ $F(A, B, C, D) = A'B + AD'$
- ④ $F(A, B, C, D) = A'C + AD'$

정답 체크 :

(3)

곱의 합 형태로 최소화 : 1을 기준으로 1, 2, 4, 8, 16개로 묶는다. 최대한 크게 묶는 것이 좋고 무관 조건은 묶을 수 있으면 묶고 묶을 수 없으면 버린다(don't care).

합의 곱 형태로 최대화 : 0을 기준으로 1, 2, 4, 8, 16개로 묶는다. 최대한 크게 묶는 것이 좋고 무관 조건은 묶을 수 있으면 묶고 묶을 수 없으면 버린다(don't care).

주어진 조건으로 곱의 합 형태로 최소화한 결과는 다음과 같다.

		AB			
		00	01	11	10
CD	00	0	1	X	1
	01	0	X	0	0
	11	X	1	0	0
	10	0	1	X	1

A'B

AD'