

2018년 지방직 9급 컴퓨터일반 풀이

by 호이호이꿀떡

정답 체크

01	02	03	04	05	06	07	08	09	10
③	①	③	③	④	③	①	④	②	④
11	12	13	14	15	16	17	18	19	20
①	②	④	④	①	②	③	③	②	③

문 1. 선형 자료구조에 해당하지 않는 것은?

- ① 큐
- ② 스택
- ③ 이진 트리
- ④ 단순 연결 리스트

답 ③

③ 이진 트리는 비선형 구조이다.

◆ 선형 자료구조: 자료를 구성하는 데이터를 순차적으로 나열시킨 형태

▷ 리스트(list)

선형 리스트: 연속되는 기억 장소에 저장되는 리스트(배열 등)

연결 리스트: 노드의 포인터 부분으로 서로 연결시킨 리스트

▷ 스택(stack): 삽입/삭제가 한 방향에서 이루어지는 데이터 구조 (LIFO 구조)

▷ 큐(queue): 리스트의 한 방향에서 삽입되어, 반대 방향으로 삭제되는 구조(rear로 삽입되어 front로 삭제된다)

▷ 데크(deque): 삽입과 삭제가 리스트의 양쪽 끝에서 이루어지는 구조

스크롤(scroll): 입력이 한쪽 끝으로만 가능한 데크(입력 제한 데크)

셀프(shelf): 출력이 한쪽 끝으로만 가능한 데크(출력 제한 데크)

◆ 비선형 자료구조: 하나의 자료 뒤에 여러 개의 자료가 존재하는 형태

▷ 트리(tree): 사이클이 없는 구조

▷ 그래프(graph): 사이클이 있는 구조

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

- ① -4
- ② -5
- ③ -6
- ④ -7

답 ①

양수 5 = 이진수 0101 (4비트라고 가정)

1의 보수 -5 : 1010

2의 보수 -5 : 1011

즉 비트열 A는 1011이다.

A를 1의 보수로 해석하면,

1011은 0100(4)의 반전값이므로

1의 보수 1011은 -4이다.

문 3. 직원 테이블 emp 의 모든 레코드를 근무연수 wyear 에 대해서는 내림차순으로, 동일 근무연수에 대해서는 나이 age 의 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;

답 ③

- ③ 오름차순을 설정할 때는 ASC, 내림차순을 설정할 때는 DESC를 사용한다. 설정을 하지 않을 때 기본값은 오름차순이다.

문 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

답 ③

③ SCAN 스케줄링(SCAN Scheduling)

디스크의 한 끝에서 시작하여 다른 끝으로 왕복 이동하며 가는 길에 있는 모든 요청을 처리

다른 한쪽 끝에 도달하면 역 방향으로 이동하면서 오는 길에 있는 요청을 처리

SSTF의 응답시간 편차를 줄일 수 있음 스캔 스케줄링

<오답 체크> ④ RR(Round Robin)는 프로세스의 선점 스케줄링 기법 중 하나이다.

디스크 스케줄링 기법이 아니다.

각 프로세스는 할당된 시간(Time Slice Quantum)동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치함

◆ 디스크 스케줄링 기법

▷ 선입 선처리(first-come-first-served)

요청이 들어온 순서대로 서비스 하는 방식

빠른 서비스를 제공하지는 못하고 비효율적

▷ 최소 탐색 시간 우선 스케줄링(SSTF Scheduling)

현재 위치에서 탐색거리가 가장 짧은 요청을 먼저 서비스하는 방식

일괄처리 시스템에 유용하며, 기아상태 발생 가능성이 있다.

▷ SCAN 스케줄링(SCAN Scheduling)

디스크의 한 끝에서 시작하여 다른 끝으로 왕복 이동하며 가는 길에 있는 모든 요청을 처리

다른 한쪽 끝에 도달하면 역 방향으로 이동하면서 오는 길에 있는 요청을 처리

SSTF의 응답시간 편차를 줄일 수 있음

▷ C-SCAN 스케줄링 (C-SCAN Scheduling)

항상 바깥쪽에서 안쪽으로 움직이며 가는 길에 있는 요청을 처리 한쪽 끝에 다다르면 처음 시작했던 자리로 다시 되돌아가서 서비스를 시작▷

▷ LOOK 스케줄링 (LOOK Scheduling)

SCAN 기법처럼 디스크를 왕복 이동하며 요청을 처리하나, 진행 방향에 더 이상의 요청이 없으면 끝까지 이동하지 않고 역으로 스캔

▷ C-LOOK 스케줄링 (C-LOOK Scheduling)

항상 바깥쪽에서 안쪽으로 움직이며 가는 길에 있는 요청을 처리 하되, 진행방향에 더 이상 요청이 없으면 바로 처음으로 돌아가 다시 서비스를 시작

▷ N-STEP SCAN

SCAN 스케줄링과 유사하나 디스크 헤드가 이동 중에 새로 들어 오는 요청은 무시하고 미리 대기 중인 요청만 처리

SSTF나 SCAN에 비해 응답시간의 분산이 적음

▷ SLTF 스케줄링

회전지연시간 최적화를 위한 알고리즘

디스크 헤드가 특정 실린더에 도착하면 그 실린더 내의 여러 트랙에 대한 요청들을 검사 후, 회전지연시간이 가장 짧은 요청부터 서비스

문 5. 정보량의 크기가 작은 것에서 큰 순서대로 바르게 나열한 것은?(단, PB, TB, ZB, EB 는 각각 petabyte, terabyte, zettabyte, exabyte 이다)

- 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

답 ④

◆ 데이터 용량

10²⁴ 요타 (yotta) 1YB
 10²¹ 제타 (zetta) 1ZB
 10¹⁸ 엑사 (exa) 1EB
 10¹⁵ 페타 (peta) 1PB
 10¹² 테라 (tera) 1TB
 10⁹ 기가 (giga) 1GB
 10⁶ 메가 (mega) 1MB
 10³ 킬로 (kilo) 1KB
 1바이트 1Byte

◆ 실행시간

1초 1second
 10⁻³ 밀리 (milli) 1ms
 10⁻⁶ 마이크로 (micro) 1μs
 10⁻⁹ 나노 (nano) 1ns
 10⁻¹² 피코 (pico) 1ps
 10⁻¹⁵ 펨토 (femto) 1fs
 10⁻¹⁸ 아토 (atto) 1as
 10⁻²¹ 젱토 (zepto) 1zs
 10⁻²⁴ 옥토 (yocto) 1ys

문 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

답 ③

◆ RAID(Redundant Array of Inexpensive/Independent Disk)

▶ RAID 0 스트라이핑(striping)

Striped disk array without fault tolerance
 단순 분산 저장 / 백업 기능 없음 / 오류 정정 없음

▶ RAID 1 미러링(mirroring)

Mirroring and duplexing
 완전 동일하게 복제 디스크 / 실제 저장용량은 절반

▶ RAID 2 해밍코드

Hamming code ECC
 오류 정정을 위해 해밍 코드 사용

▶ RAID 3 바이트 단위 패리티

Parallel transfer with parity
 패리티 전용 디스크 1개(바이트 단위)

▶ RAID 4 블록 단위 패리티

Independent data disks with shared parity disk
 패리티 전용 디스크 1개(블록 단위)

▶ RAID 5 분산 패리티

Independent data disks with distributed parity blocks
 패리티 블록을 각 디스크에 분산 저장 / 병목 현상 없음

▶ RAID 6 이중 패리티

Independent data disks with two independent distributed parity schemas
 패리티를 이 중으로 두 개의 디스크에 저장

▶ RAID 7

컨트롤러로서 내장되어 있는 실시간 운영체계를 사용
 속도가 빠른 버스를 통한 캐시

▶ RAID 0+1보다 RAID 1+0이 안정성이 좋다.

문 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.

답 ①

① 애자일 개발 방법론(Agile software development)

일정한 주기를 가지고 끊임없이 프로토타입을 만들어내며 그때 그때 필요한 요구를 더하고 수정하여 하나의 커다란 소프트웨어를 개발해 나가는 방법론이다.

애자일 개발 프로세스란 어느 특정 개발 방법론을 가리키는 말은 아니고, 애자일(Agile=기민한, 좋은것을 빠르고 낭비없게 만드는 것) 개발을 가능하게 해 주는 다양한 방법론 전체를 일컫는 말이다. 대표적인 기법으로 익스트림 프로그래밍 (XP, eXtreme Programming)이 있다.

<오답 체크> ②④ 애자일 기법은 계획이 지나치게 많은 개발 방법론의 단점을 보완하고자 등장한 개발론으로, 문서를 통한 개발 방법이 아니라 실질적인 코딩을 통한 방법론(code-oriented)이라는 점이다.

문서를 통한 개발 방법론(document-oriented)으로는 고전적인 방법론, 폭포수 모델이 있다.

- ③ 애자일 기법은 환경의 변화에 맞추며 단계적으로 개발하는 적응적 모델(adaptive style)이다.

문 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

- ① 128 개
- ② 256 개
- ③ 512 개
- ④ 1,024 개

답 ④

2-way 집합 연관 사상이기 때문에, 하나의 집합(set)은 2개의 라인으로 구성되어 있다.

주소 구조에서 세트 필드가 9비트이므로 집합 번호를 9비트로 표현한다는 의미이다.

$2^9 = 512$, 총 집합의 개수는 512개이다.

→ 그러므로, 캐시의 전체 라인 수는 $2 \times 512 = 1024$ 개이다.

▷ 태그(tag)는 캐시로 적재된 데이터가 주기억장치 어느 곳에서 온 데이터인지 구분하기 위한 번호

오프셋(offset) 값은 해당 페이지나 세그먼트 내에서 몇 번째 위치에 들어있는지를 나타내는 변위값

문 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

답 ②

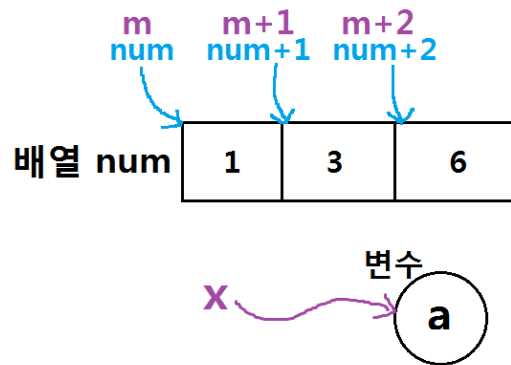
먼저 main()함수에서
 num배열과 정수형 변수 a, b가 지역변수로 선언된다.

func(num, &a, b);에서 num배열과 변수 a는 주소값으로 넘겨주고, b는 값을 직접 복사해서 넘겨준다.

배열은 뒤에 배열번호를 쓰지 않고 변수명만 쓸 경우 그 자체로 주소 값을 의미한다.

(num: 배열 num의 주소값, 첫 번째 요소의 주소값)

일반 변수 앞에 &를 붙이면 그 변수의 주소값을 의미한다.



아래 func(num, &a, b)를 통해

m은 num배열의 주소 값을 받고, x는 a의 주소 값을 받고, y는 b의 값을 복사 받아 y=30이 들어간다.

*x는 주소인 x가 가리키는 값을 의미하는데, x에는 a의 주소가 들어있으므로, *x는 a에 해당한다.

y = *x; 에서 y에 a의 값인 10이 들어간다.

n = *(m + 1) + (*m + 2);

m은 배열 num의 주소(num의 첫 번째 주소)이며,

m + 1은 배열 num의 2번째 요소의 주소가 된다.

따라서 *(m + 1)은 2번째 요소의 값인 3이 된다.

(*m + 2)에서 *m은 배열 num의 첫 번째 요소인 1이며,

따라서 *m + 2는 1 + 2 = 3 이 된다.

따라서 n = 3 + 3 = 6 이 들어간다.

*x = ++n; 에서 ++이 앞에 붙어있으므로 먼저 n 값을 1 증가시킨 뒤, x가 가리키는 곳(a)에 넣는다. (a = 7)

이후 다시 main()함수로 복귀하여 a와 b값을 출력하는데,

a는 변경 된 값 7이 출력되며,

b는 func()함수의 y에 값만 복사해줬을 뿐, 바뀐 적이 없다.

따라서 **b의 값은 처음 30 그대로 출력**이 된다.

문 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.

답 ④

- ④ TCP에서 헤더 체크섬(checksum)은 필수 사항이며, UDP에서 헤더 체크섬은 선택 사항이다.

<오답 체크> ①③ TCP는 OSI 4계층 전송계층에서 작동하는 신뢰성, 연결 지향형의 프로토콜이다. 전이중 전송방식/양방향성(full-duplex)의 특성을 가지고 있으며, 흐름제어, 혼잡제어 등의 기능을 수행한다.

② SYN → SYN-ACK → ACK의 3way-handshake 과정을 통해 연결을 확립한다.

문 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

답 ①

- ① 클라우드 컴퓨팅(cloud computing)은 인터넷 기반 컴퓨팅의 일종으로 정보를 자신의 컴퓨터가 아닌 인터넷에 연결된 다른 컴퓨터로 처리하는 기술을 의미한다.

▷ Infrastructure as a service(IaaS)

클라우드를 이용해, 서버 자원, IP, 네트워크, 저장 공간, 전력 등 컴퓨팅 인프라를 사용하는 데 필요한 자원을 서비스하는 방식이다. 인터넷 상에서 단순 데이터 저장 기능만 제공하는 클라우드 저장 서버가 여기에 해당한다.

▷ Platform as a service(PaaS)

서비스를 개발 할 수 있는 안정적인 플랫폼(platform)과 그 환경을 이용하는 응용 프로그램을 개발 할 수 있는 API를 제공하는 서비스 방식이다.

▷ Software as a service(SaaS)

모든 서비스가 클라우드에서 이뤄지는 서비스 방식으로, 별도의 소프트웨어 설치 없이, 클라우드 환경에서 소프트웨어를 이용할 수 있는 방식이다.

이메일 서비스, MS 오피스 365, 구글 문서 등

▷ Mobile Backend as a Service(MBaaS)

모바일 앱개발에 필요한 위치기반서비스, 푸시알림, 포토 콜렉션, 사용자 인증, 소셜(social) 네트워크와의 통합 등, 서버와 통신하는 백엔드를 서비스 형태로 개발자에게 제공해 서버 기술을 몰라도 그 환경에 연결되는 모바일 앱을 만들 수 있도록 지원하는 서비스 방식

<오답 체크> ② USN(Ubiquitous Sensor Network)

필요로하는 모든 것에 통신기능이 있는 스마트 RFID 태그 및 센서를 부착하여, 사물의 인식정보 및 주변의 환경정보(온도, 습도, 오염정보,균열정보 등)를 탐지하고, 실시간으로 네트워크에 연결하여 정보를 관리하는 기술

③ 웨어러블 컴퓨터(wearable computer)

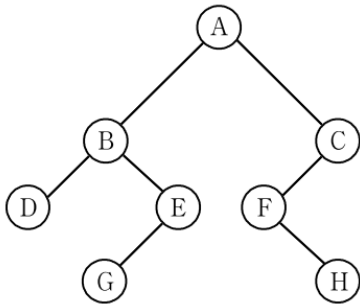
안경, 시계, 의복 등과 같이 착용할 수 있는 형태로 된 컴퓨터로, 사용자가 거부감 없이 신체의 일부처럼 항상 착용하고 인간의 능력을 보완하거나 배가시키기 위한 컴퓨터이다.

언제 어디서나(항시성), 쉽게 사용할 수 있고(편의성), 착용하여 사용하기에 편하며(착용감), 안전하고 보기 좋은(안정성/사회성) 특성이 요구된다.

④ SNS(Social Networking Service, 소셜 네트워킹 서비스)

사용자 간의 자유로운 의사소통과 정보 공유, 그리고 인맥 확대 등을 통해 사회적 관계를 생성하고 강화해주는 온라인 플랫폼

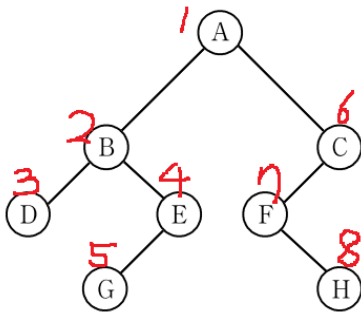
문 12. 다음 이진 트리의 노드를 전위 순회(preorder traversal)할 경우의 방문 순서는?



- 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

답 ②

전위 순회는 루트 노드를 먼저 순회한다.



문 13. 전자메일의 송신 또는 수신을 목적으로 하는 응용 계층 프로토콜에 해당하지 않는 것은?

- ① IMAP
- ② POP3
- ③ SMTP
- ④ SNMP

답 ④

④ SNMP(Simple Network Management Protocol, 간이 망 관리 프로토콜)

IP 네트워크상의 장치로부터 정보를 수집 및 관리하며, 또한 정보를 수정하여 장치의 동작을 변경하는 데에 사용 프로토콜 구조가 간단하고 구현이 용이해서 네트워크 모니터링의 목적으로 네트워크 관리에서 널리 사용된다.

<오답 체크> ① IMAP: 전자메일 수신 프로토콜, 읽어도 삭제되지 않음, 다운 여부 본인이 결정

② POP3: 전자메일 수신 프로토콜, 읽으면 자동 삭제 자동 다운

③ SMTP: 전자메일 송신 프로토콜

문 14. 모바일 기기에 특화된 운영체제에 해당하지 않는 것은?

- ① iOS
- ② Android
- ③ Symbian
- ④ Solaris

답 ④

④ Solaris

썬 마이크로시스템즈(SUN microsystems)에서 개발한 컴퓨터 운영 체제로, 유닉스 계열의 운영체제이다. Solaris 11까지는 오픈 소스였으나, 2010년 오라클(Oracle)이 썬을 인수한 후 더 이상 오픈 소스로 공개하지 않는다.

<오답 체크> ① iOS

미국 애플사(Apple)가 생산하는 모바일 기기에 탑재되는 OS 아이폰, 아이패드, 아이팟 터치 등에 탑재

② Android(안드로이드)

스마트폰과 같은 플랫폼의 모바일 운영체제와 미들웨어 및 중요 애플리케이션이 포함된 소프트웨어 집합. 구글(Google)에서 제작하였으며, 리눅스 커널을 기반으로 만들어졌다.

③ Symbian OS(심비안 OS)

핀란드 노키아(Nokia)가 영국 심비안(Symbian)을 인수하면서 개발한 모바일 기기용 운영 체제로, 라이브러리, 사용자 인터페이스, 프레임워크, 다양한 도구를 포함한다.

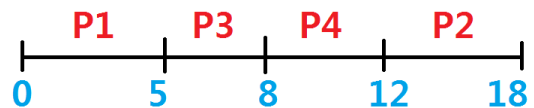
문 15. 다음 표는 단일 중앙처리장치에 진입한 프로세스의 도착 시간과 그 프로세스를 처리하는 데 필요한 실행 시간을 나타낸 것이다. 비선점 SJF(Shortest Job First) 스케줄링 알고리즘을 사용한 경우, P1, P2, P3, P4 프로세스 4 개의 평균 대기 시간은? (단, 프로세스 간 문맥 교환에 따른 오버헤드는 무시하며, 주어진 4 개의 프로세스 외에 처리할 다른 프로세스는 없다고 가정한다)

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

답 ①

▷ 비선점 SJF(Shortest Job First) 실행시간이 가장 짧은 프로세스에 먼저 CPU를 할당하는 기법이다.



1. 먼저 가장 먼저 들어온 **P1**을 처리한다. (0 ~ 5ms)
2. 5ms 시점에 대기열에 들어와 있는 P2와 P3 중 실행시간이 짧은 **P3**를 먼저 실행한다. (5 ~ 8ms)
3. 8ms 시점에 대기열에 있는 P2와 P4 중 실행시간이 짧은 **P4**를 먼저 실행한다. (8 ~ 12ms)
4. 남은 **P2**를 실행한다. (12 ~ 18ms)
대기시간은 완료시간에서 도착시간과 실행시간을 빼서 구한다.
 $(0 + 9 + 1 + 2) / 4 = 3ms$

프로세스	도착	실행	완료	대기
P1	0	5	5	0
P2	3	6	18	9
P3	4	3	8	1
P4	6	4	12	2

문 16. IPv4 와 IPv6 에 대한 설명으로 옳지 않은 것은?

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

답 ②

- ② IPv4는 8비트씩 4자리, 총 32비트(4바이트)
IPv6는 16비트씩 8자리, 총 128비트(16바이트)
4배 차이이다.

<오답 체크> ① IP는 OSI 3계층 네트워크 계층 프로토콜로, 비연결형이며 신뢰성을 보장하지 않는다. 인터넷 경로설정을 위한 라우팅 역할을 수행하며, 패킷의 분해(단편화), 조립, 논리적 경로지정 기능을 제공한다.
순서 검사(sequence checking)와 흐름 제어, 오류 복구 기능 등이 없다.

- ③ IPv6에는 기존 IPv4에 있던 브로드캐스트 주소 방식 대신 애니캐스트 주소 방식이 도입되었다.(유니캐스트, 멀티캐스트 주소 방식은 동일)

IPv4: 유니캐스트, 멀티캐스트, 브로드캐스트

IPv6: 유니캐스트, 멀티캐스트, 애니캐스트

- ④ IPv4와 IPv6 전환 기술

▷ 터널링(Tunneling): IPv4 망에 터널을 만들어 IPv6가 지나갈 수 있게 하는 기술

▷ 듀얼 스택(Dual Stack): 하나의 시스템에서 IPv4와 IPv6프로토콜을 함께 처리

▷ 헤더 변환(Header Translation) 패킷의 헤더 변환

문 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)

답 ③

- ③ 카운터(counter)는 순차 논리회로에 해당한다.

동기식 카운터: 모든 플립플롭이 하나의 공통 클럭에 연결되어 있어 동시에 트리거 되는 카운터

비동기식 카운터: 리플(ripple) 카운터라고도 하며, 앞쪽에 있는 플립플롭의 출력이 뒤쪽에 있는 플립플롭의 클럭으로 사용되는 카운터

- <오답 체크> ① 디코더(decoder, 복호기)

n개의 입력으로 들어오는 데이터를 받아, 2^n개의 출력선 중 한 곳으로 1을 출력하는 조합 논리회로

3-to-8 디코더는 3개의 입력을 받아, 8개의 출력선 중 한 곳으로 출력하는 것

- ② 전가산기(full adder)

1비트 이진수 2개와 하위의 자리올림수 1개를 입력으로, 총 3개의 이진수를 더하여 합과 자리올림수를 출력하는 논리회로.

전가산기는 두 개의 반가산기와 하나의 OR로 구성

- ④ 멀티플렉서(multiplexer, MUX)

여러 입력선과 하나의 출력선으로 구성되어 있으며, 여러 입력 신호 중 하나를 선택해 그 입력 신호를 출력선에 전달하는 논리회로

- ▶ 조합 논리회로(combinational logic circuit)

출력값이 입력값에 의해서만 결정

가산기, 비교기, 연산기, 디코더, 인코더, 멀티플렉서, 디멀티플렉서, 코드 변환기 등

- ▶ 순차 논리회로(sequential logic circuit)

출력값이 입력값과 이전 상태의 논리값에 의해 결정

플립플롭, 카운터, 레지스터, RAM, CPU 등

문 18. 클럭(clock) 주파수가 2 GHz 인 중앙처리장치를 사용하는 컴퓨터 A 에서 프로그램 P 를 실행하는 데 10 초가 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

답 ③

정석 풀이법으로는 **CPI**(clock cycles per instruction, 명령어당 클럭 사이클 수)를 이용하는 것이나, 객관식에서 굳이 식을 쓸 필요 없이 빨리 계산하는 것이 중요하다.

G는 기가(giga)는 2의 30승, 10의 9승, 즉 10억을 의미한다.
클럭 주파수가 2GHz라는 건 1초에 20억 개의 클럭 사이클을 수행한다는 의미로,
프로그램 P의 전체 클럭 사이클은 200억 개가 된다.

컴퓨터 B에서 클럭 사이클 수가 1.5배로 증가하였으므로 300억 개가 되었으며,
실행 시간은 6초이므로 1초당 50억 개의 클럭 사이클을 처리한 것이다.
즉 컴퓨터 B의 클럭 주파수는 **5GHz**이다.

▶ 빠른 풀이

원래 A 컴퓨터에서 실행시간 10초 = 2GHz
B 컴퓨터에서 클럭 사이클 1.5배로 증가(처리량 1.5배)
 $2\text{GHz} \times 1.5 = 3\text{GHz}$
B 컴퓨터의 처리 시간은 10초에서 6초로 감소(역수 곱셈)
 $2\text{GHz} \times 1.5 \times (10/6) = 5\text{GHz}$

문 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

답 ②

i = 1, j = 1, k = 0 에서부터 시작
(i % 2) == 0; 이 거짓이므로 if문을 건너뛰고 아래 부분을 실행한다.
k += i * j++; 에서 ++이 j의 뒤에 붙었으므로,
k += i * j 를 먼저 연산한 후 j값을 1증가시킨다.
k += 1 * 1 를 통해 k에 1을 더해서 1이 되고,
j는 1이 증가해 2가 된다.

i가 1 증가하여
i = 2, j = 2, k = 1 에서 for문이 실행된다.
(i & 2) == 0; 이 참이므로, continue;가 실행,
continue 아래 부분을 모두 무시하고 다시 for문의 첫 줄로 돌아간다.

i가 1 증가하여 3이되고,
i = 3, j = 2, k = 1에서 for문 실행
(if % 2) == 0; 이 거짓이므로 아래 부분 실행
k += i * j++; 에서
k += 3 * 2 를 통해 k에 6을 더해 7이 되고,
j는 1이 증가해 3이 된다.

i = 4, j = 3, k = 7에서 for문 실행
(i & 2) == 0; 이 참이므로, continue;가 실행, 아래를 건너뛰고 첫 줄로 돌아간다.

i = 5가 되어, i < 5가 거짓이므로 for문을 빠져나온다.
그리고 **k값 7이 출력된다.**

문 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, X 는 무관(don't care) 조건을 나타낸다)

CD \ AB	AB			
	00	01	11	10
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 $F(A, B, C, D) = A'C + AD'$

답 ③

- ③ 카르노맵을 작성할 땐 최대한 크게 묶고 식이 간단해지도록 만든다.
 X는 무관항(don't care)이므로 카르노맵으로 묶어도 되고 묶지 않아도 된다.

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

빨간색 부분으로 4개를 묶을 수 있다. $A'B$

파란색 부분으로도 4개를 묶을 수 있다. AD'

왼쪽 줄의 X는 무관항이므로 굳이 추가하진 않는다.

따라서 최소화를 하면 $A'B + AD'$ 가 된다.

<오답 체크>

① $AD' + BC'D' + A'BC$

1을 모두 포함하긴 했지만, 4개로 묶을 수 있는 부분을 그냥 넘어갔다.

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

② $AB'D' + BC'D' + A'BC$

①번과 마찬가지로 4개로 묶을 수 있는 부분을 그냥 넘어갔다.

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

④ $A'C + AD'$

빨간색 부분에 0이 포함되어 있어 틀린 표현이다.

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