

2019년 지방직 9급 컴퓨터일반 풀이

by 호이호이꿀떡

정답 체크

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

문 1. 저급언어에 해당하는 프로그래밍 언어는?

- ① 어셈블리어(Assembly Language)
- ② 자바(Java)
- ③ 코볼(COBOL)
- ④ 포트란(Fortran)

▶ 저급언어(low-level programming language)

컴퓨터가 이해하기 쉽게 작성된 프로그래밍 언어로, 일반적으로 기계어와 어셈블리어를 일컫는다. 하지만 단순한 2진수, 16진수 등의 숫자로만 표현하기 때문에 사람이 읽고 쓰기에는 어렵다. 또한 각 컴퓨터마다 작동 구조가 다르기 때문에 저급언어 작성 방식이 달라 호환성이 떨어진다.

▶ 고급언어(high-level programming language)

사람이 이해하기 쉽게 작성된 프로그래밍 언어로서, 가독성이 높고 다루기 간단하고 호환성이 좋다는 장점이 있다. 고급언어는 컴파일러나 인터프리터에 의해 저급언어로 번역되어 실행되며, 번역 과정 때문에 저급언어에 비해 실행속도는 느리다. C, JAVA, BASIC, Kotlin 등 프로그래머들이 사용하는 대부분의 언어들은 고급언어에 속한다.

답 ①

문 2. 중앙처리장치(CPU)의 구성 요소로만 묶은 것은?

ㄱ. ALU	ㄴ. DRAM
ㄷ. PCI	ㄹ. 레지스터
ㅁ. 메인보드	ㅂ. 제어장치

- ① ㄱ, ㄴ, ㄹ
- ② ㄱ, ㄹ, ㅂ
- ③ ㄹ, ㅁ, ㅂ
- ④ ㄱ, ㄷ, ㄹ, ㅂ

CPU 구성요소는 크게 제어장치, 연산장치(ALU), 레지스터로 구성된다.

- ㄷ. 제어장치는 명령어를 해독하고 동작을 제어하는 역할,
- ㄱ. 연산장치(ALU, Arithmetic and Logical unit)는 산술, 논리, 관계 등의 연산을 수행하는 역할,
- ㄹ. 레지스터는 명령어나 연산의 중간값을 임시 기억하는 역할

<오답 체크> ㄴ. DRAM은 CPU와 데이터를 주고받을 뿐, CPU의 구성요소는 아니다. 데이터를 각기 분리된 축전기(Capacitor)에 저장하는 기억장치로, 데이터를 유지하기 위해 주기적인 충전이 필요하다. SRAM보다 느리지만, 전력소모가 적고 저렴하다.

ㅁ. PCI는 컴퓨터의 메인보드에 주변장치를 연결하는 버스(bus)의 한 규격 방식이다. 개인 PC에 가장 많이 사용되는 버스 방식이며, 현재는 한 단계 더 발전된 PCI-E(PCI 익스프레스)로 많이 넘어가고 있는 추세이다.

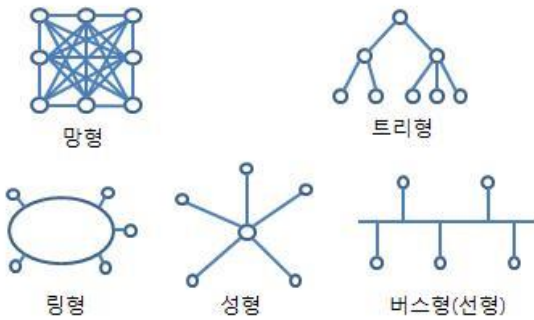
ㅂ. 메인보드(mainboard) 또는 마더보드(motherboard) 컴퓨터 부품들을 통합하여 연결해주는 인쇄회로기판을 의미한다. 사람으로 따지면 기본 뼈대와 비슷하다. 메인보드에 CPU, 램(RAM), 그래픽카드(VGA), 랜카드 등을 부착하면, 메인보드 내의 회로들을 통해 통합 연결된다.

답 ②

문 3. 다음에서 설명하는 네트워크 구조는?

- 구축 비용이 저렴하고 새로운 노드를 추가하기 쉽다.
- 네트워크의 시작과 끝에는 터미네이터(Terminator)가 붙는다.
- 연결된 노드가 많거나 트래픽이 증가하면 네트워크 성능이 크게 저하된다.

- ① 링(Ring)형
- ② 망(Mesh)형
- ③ 버스(Bus)형
- ④ 성(Star)형



답 ③

문 4. 다음에서 설명하는 객체지향 프로그래밍의 특징은?

- 객체를 구성하는 속성과 메서드가 하나로 묶여 있다.
- 객체의 외부와 내부를 분리하여 외부 모습은 추상적인 내용으로 보여준다.
- 객체 내의 정보를 외부로부터 숨길 수도 있고, 외부에 보이게 할 수도 있다.
- 객체 내부의 세부 동작을 모르더라도 객체의 메서드를 통해 객체의 기능을 활용할 수 있다.

- ① 구조성
- ② 다형성
- ③ 상속성
- ④ 캡슐화

④ 보기는 모두 캡슐화에 대한 설명이다.

<오답 체크> ① 구조성(structured)은 객체지향 프로그래밍이 아닌 구조적 프로그래밍의 특징이다.

프로그래밍을 작성할 때 알고리즘을 계층적인 구조가 되도록 설계하는 방식을 말한다. 순차, 선택, 반복의 3가지로 기본 구조로 구성된다.

② 다형성(polymorphism)이란 하나의 메소드나 클래스가 객체에 따라 다양한 방법으로 동작하는 것을 의미한다. 오버로딩, 오버라이딩 등이 있다.

③ 상속성(inheritance)은 기존의 클래스로부터 속성과 동작을 물려받는 것을 의미한다. 상속을 물려주는 클래스가 상위 클래스(부모 클래스)가 되며, 물려받는 클래스가 하위 클래스(자식 클래스, 파생 클래스)가 된다.

답 ④

문 5. 하나의 프로세스가 CPU를 할당받은 후에는, 스스로 CPU를 반납할 때까지 다른 프로세스가 CPU를 차지할 수 없도록 하는 스케줄링 기법에 해당하는 것만을 모두 고르면?

- ㄱ. FCFS(First Come First Served)
- ㄴ. RR(Round Robin)
- ㄷ. SRT(Shortest Remaining Time)

- ① ㄱ
- ② ㄱ, ㄷ
- ③ ㄴ, ㄷ
- ④ ㄱ, ㄴ, ㄷ

스스로 반납할 때까지 다른 프로세스가 차지할 수 없는 기법은 비선점(Nonpreemptive) 기법을 말한다.
 (ㄱ) FCFS는 비선점 기법에 해당하고,
 (ㄴ) RR과 (ㄷ) SRT는 선점 기법에 해당한다.

- ◆ 비선점(Non-Preemptive) 스케줄링 종류
 - ▷ FCFS(First-Come First-Service) = FIFO(First In First Out)
준비상태 큐에 도착한 순서에 따라 차례로 CPU를 할당
 - ▷ SJF(Shortest Job First)
실행시간이 가장 짧은 프로세스에 먼저 CPU를 할당하는 기법
 - ▷ HRN(Highest Responderatio Next)
우선순위 계산 공식 = 대기시간 + 서비스시간 / 서비스시간
 - ▷ 기한부(Deadline)
프로세스에게 일정한 시간을 주어 그 시간 안에 프로세스를 완료하도록 하는 기법
 - ▷ 우선순위(Priority)
준비상태 큐에서 기다리는 각 프로세스마다 우선순위를 부여하는 기법
 - ▷ 에이징(Aging) 기법
기아 상태를 해결하기 위해, 기다린 시간에 비례하여 일정 시간이 지나면 우선순위를 한 단계씩 높여 가까운 시간 안에 자원을 할당받도록 하는 기법

- ◆ 선점(Preemptive) 스케줄링 종류
 - ▷ 선점 우선순위
준비 상태 큐의 프로세스들 중에서 우선순위가 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법
 - ▷ SRT(Shortest Remaining Time)
비선점 기법 SJF 알고리즘을 선점 형태로 변경한 기법
현재 실행 중 프로세스의 남은 시간과 준비상태 큐에 새로 도착한 프로세스의 실행시간을 비교해 가장 짧은 실행 시간을 요구하는 프로세스에게 CPU를 할당하는 기법
 - ▷ RR(Round Robin, 라운드 로빈)
각 프로세스는 할당된 시간(Time Slice Quantum)동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치함
 - ▷ 다단계 큐(Multi level Queue)
프로세스를 특정 그룹으로 분류할 수 있는 경우 그룹에 따라 각기 다른 준비상태 큐를 사용하는 기법
 - ▷ 다단계 피드백 큐(Multi level Feedback Queue)
특정 그룹의 준비상태 큐에 들어간 프로세스가 다른 준비상태 큐로 이동할 수 없는 다단계 큐 기법을 준비상태 큐 사이를 이동할 수 있도록 개선한 기법

문 6. 프로그램 내장 방식에 대한 설명으로 옳지 않은 것은?

- ① 프로그램 내장 방식을 사용한 최초의 컴퓨터는 에니악(ENIAC)이다.
- ② 현재 사용되는 대부분의 컴퓨터는 프로그램 내장 방식을 사용하고 있다.
- ③ 컴퓨터가 작업을 할 때마다 설치된 스위치를 다시 세팅해야 하는 번거로움을 해결하기 위해 폰 노이만이 제안하였다.
- ④ 프로그램과 자료를 내부의 기억장치에 저장한 후 프로그램 내의 명령문을 순서대로 꺼내 해독하고 실행하는 개념이다.

① 프로그램 내장 방식을 사용한 최초의 실용적인 컴퓨터는 **에드삭(EDSAC)**이다.

참고로, 맨체스터 대학의 SSEM이 세계 최초의 프로그램 내장방식의 컴퓨터이긴 하나, 실용적인 면에서 에드삭이 현재 컴퓨터의 기초가 되기 때문에 대부분 최초의 프로그램 내장방식으로 에드삭을 꼽는다.

답 ①

문 7. CISC(Complex Instruction Set Computer)와 RISC(Reduced Instruction Set Computer)에 대한 설명으로 옳지 않은 것은?

- ① CISC 구조에서 명령어의 길이는 가변적이다.
- ② 전형적인 RISC 구조의 명령어는 메모리의 피연산자를 직접 처리한다.
- ③ RISC 구조는 명령어 처리구조를 단순화시켜 기계어 명령의 수를 줄인 것을 말한다.
- ④ CISC 구조는 RISC 구조에 비해서 상대적으로 명령어 실행 단계가 많고 회로 설계가 복잡하다.

② 메모리의 피연산자를 직접 처리하는 것은 CISC 구조에 해당한다. 예를 들어, 복잡한 명령어 길이도 가능한 CISC 구조에서는 $X = (A * B) + C$ 라는 계산을 하나의 명령어로 바로 실행할 수 있다면, 간단하고 고정된 길이의 명령어로 구성된 RISC 구조에서는 LOAD A -> LOAD B -> MUL(곱하기) -> LOAD C -> ADD(더하기) -> STORE X 이러한 방식으로 단계별로 처리한다.

- ◆ CISC (Complex Instruction Set Computer)
 - 다양한 길이의 명령어
 - 회로가 복잡하고 동시에 여러 개의 명령어 처리가 어려움
 - 컴파일러 작성이 쉽고 호환성이 좋음
 - 마이크로 프로그래밍 방식
- ◆ RISC (Reduced Instruction Set Computer)
 - 고정된 길이의 명령어
 - 명령어 처리 속도가 빠르며 동시에 여러 명령어 처리 가능
 - 파이프라인 성능에 최적화(파이프라인 해저드 주의)
 - 컴파일러 최적화 과정이 복잡하고 코드 효율성이 낮음
 - 많은 수의 범용 레지스터
 - 하드와이어드(Hardwired) 방식

답 ②

문 8. 릴레이션 R = {A, B, C, D, E}이 함수적 종속성들의 집합 FD = {A → C, {A, B} → D, D → E, {A, B} → E}를 만족할 때, R이 속할 수 있는 가장 높은 차수의 정규형으로 옳은 것은? (단, 기본키는 복합속성 {A, B}이고, 릴레이션 R의 속성 값은 더 이상 분해될 수 없는 원자 값으로만 구성된다)

- ① 제1정규형
- ② 제2정규형
- ③ 제3정규형
- ④ 보이스 / 코드 정규형

이행 함수 종속은 X → Y이고 Y → Z일 때 X → Z를 만족하는 관계를 의미한다.

{A, B} → D, D → E 를 보면,

D는 {A, B}에 의해 결정되고, E는 D에 의해 결정된다. 따라서 현재는 이행 함수 종속이 남아있는 상태이며, 제3정규형이 되지 않는다.

(이행 함수 종속을 제거하기 위해 D와 E를 별도의 릴레이션으로 구성해야 한다.)

부분 함수 종속은 어떤 속성 Y가 다른 복합 속성 X의 부분집합에 종속하는 관계를 의미한다.

현재 기본키는 {A, B}인데 A → C 종속이 남아있으며, 부분 함수 종속에 해당한다. 따라서 현재 제2정규형이 되지 않는다.

(부분 함수 종속을 제거하기 위해 A와 C를 별도의 릴레이션으로 구성해야 한다.)

현재 릴레이션 R의 속성 값은 더 이상 분해될 수 없는 원자 값으로만 구성되어 있다고 하였으므로 제1정규형만 만족되어 있는 상태이다.

- ▶ 제1정규형 (1NF)
도메인이 원자값만으로 되어 있어야 함
- ▶ 제2정규형 (2NF)
부분 함수적 종속 관계를 제거해야 함
- ▶ 제3정규형 (3NF)
이행 함수적 종속 (x → y → z) 제거
- ▶ BC(Boyce-codd) 정규형 (BCNF)
결정자이면서 후보키가 아닌 것 제거
- ▶ 제4정규형
다치 종속 제거
- ▶ 제5정규형
모든 조인 종속성이 후보키를 통해서만 연결. 그 외 조인 종속성 제거

답 ①

문 9. 인터넷의 전송 계층에서 사용하는 프로토콜로 TCP와 UDP가 있다. TCP와 UDP 모두에서 제공하지 않는 기능은?

- ① 연결 설정(Connection Setup)
- ② 오류 검출(Error Detection)
- ③ 지연시간 보장(Delay Guarantee)
- ④ 혼잡 제어(Congestion Control)

③ 연결 설정, 오류 검출, 혼잡 제어 모두 TCP에서 제공하는 기능에 해당하나, 지연시간 보장은 TCP와 UDP 모두 제공하지 않는다.

TCP의 특징: 연결지향적, 신뢰성 있는 전송, 흐름 제어, 혼잡 제어, 에러 검출 등

UDP의 특징: 비연결성, 신뢰성 없음, 흐름 제어 없음, 혼잡 제어 없음, 확인응답 없음, 오류 검출 기능 부족(최소한의 검사합 기능은 있음)

답 ③

문 10. 유비쿼터스를 응용한 컴퓨팅 기술에 대한 설명으로 옳지 않은 것은?

- ① 엑조틱 컴퓨팅(Exotic Computing)은 스스로 생각하여 현실세계와 가상세계를 연계해 주는 컴퓨팅 기술이다.
- ② 노매딕 컴퓨팅(Nomadic Computing)은 장소에 상관없이 다양한 정보기기가 편재되어 있어 사용자가 정보기기를 휴대할 필요가 없는 컴퓨팅 기술이다.
- ③ 디스포절 컴퓨팅(Disposable Computing)은 컴퓨터가 센서 등을 통해 사용자의 상황을 인식하여 사용자가 필요로 하는 정보를 제공해 주는 컴퓨팅 기술이다.
- ④ 웨어러블 컴퓨팅(Wearable Computing)은 컴퓨터를 옷이나 안경처럼 착용할 수 있게 해줌으로써 컴퓨터를 인간의 몸의 일부로 여길 수 있도록 하는 컴퓨팅 기술이다.

③ 감지 컴퓨팅(Sentient Computing)에 대한 설명이다.
 디스포절 컴퓨팅(Disposable Computing, 1회용 컴퓨팅)은 1회용 종이처럼 한 번 쓰고 버릴 수 있을 정도로 저렴한 컴퓨터 기술이며, 이 기술은 어떤 물건에나 컴퓨터를 내장하여 활용할 수 있게 해준다.

<오답 체크> ① 엑조틱 컴퓨팅(Exotic Computing)은 스스로 생각하여 현실세계와 가상세계를 연계해 주는 기술로, 주변의 상황과 사용자의 업무 등을 지능적으로 파악하여 수행하도록 도와주는 것이 가능하다.

② 노매딕 컴퓨팅(Nomadic Computing)은 네트워크의 이동성을 극대화해, 어떠한 장소에든 이미 다양한 정보기기가 편재되어 있어 사용자가 정보기기를 굳이 휴대할 필요가 없게 해주는 기술이다.

(참고) exotic: 외국의, 이국적인
 nomadic: 유목의 방랑의
 disposable: 사용 후 버리게 되어 있는, 일회용의, 이용가능한
 sentient: 지각이 있는

답 ③

문 11. 컴퓨터 명령어 처리 시 필요한 유효 주소(Effective Address)를 찾기 위한 주소 지정 방식에 대한 설명으로 옳지 않은 것은?

- ① 즉시 주소 지정 방식(Immediate Addressing Mode)은 유효 데이터가 명령어 레지스터 내에 있다.
- ② 간접 주소 지정 방식(Indirect Addressing Mode)으로 유효 데이터에 접근하는 경우 주기억장치 최소접근횟수는 2이다.
- ③ 상대 주소 지정 방식(Relative Addressing Mode)은 프로그램 카운터와 명령어 내의 주소필드 값을 결합하여 유효 주소를 도출한다.
- ④ 레지스터 주소 지정 방식(Register Addressing Mode)은 직접 주소 지정 방식(Direct Addressing Mode)보다 유효 데이터 접근속도가 느리다.

④ 레지스터 주소지정방식은 레지스터에만 유효 데이터가 들어있어 주기억장치에는 접근하지 않는다. 따라서 주기억장치에 접근해야 하는 직접 주소지정방식보다 빠르다.

<오답 체크> ② 간접 주소지정 방식은 명령어의 오퍼랜드에 유효 데이터의 주소를 기록한 곳의 주소를 담는 방식이다.

예를 들어, 주기억장치 1001번지에 'apple'이라는 유효 데이터가 들어있다. 그리고 주기억장치 500번지에는 '1001번지'라는 주소 값을 저장한다. 그리고 명령어의 오퍼랜드에는 '500번지'라고 기록되어 있다.

유효 데이터를 구할 때 먼저 오퍼랜드에 기록된 '500번지'를 찾아간다(첫 번째 접근). 그리고 여기에 기록된 주소값 '1001번지'를 이용해, 다시 주기억장치 주소 1001번지를 찾아간다(두 번째 접근). 그 결과 1001번지에 기록된 유효 데이터 'apple'를 찾아 가는 방식이다. 따라서 주기억장치에 두 번 접근한다.

- ▷ 암시적(묵시적) 주소지정방식(Implied Addressing Mode) 데이터의 저장 위치를 따로 명시하지 않고, 누산기나 스택의 데이터를 사용하도록 암묵적으로 약속해놓은 방식. 오퍼랜드가 없거나 1개인 명령어 형식에 사용
- ▷ 즉시 주소지정방식(Immediate Addressing Mode) 명령어의 오퍼랜드 부분에 실제 데이터를 기록하는 방식. 별도의 주소가 필요없다.
- ▷ 직접 주소지정방식(Direct Addressing Mode) 오퍼랜드 부분에 실제 데이터의 주소를 기록하는 방식. 주기억장치에 한 번 접근해야 한다.
- ▷ 간접 주소지정방식(Indirect Addressing Mode) 오퍼랜드에 실제 데이터의 주소를 가리키는 곳의 주소를 기록하는 방식. 주소를 두 번 해석하므로 주기억장치에 두 번 접근해야 한다.
- ▷ 레지스터 주소지정방식(Register Addressing Mode) 실제 데이터를 레지스터에 기록하는 방식. 오퍼랜드에는 주기억장치의 주소가 아닌 레지스터의 주소가 기록된다. 따라서 주기억장치에 접근할 필요가 없다.
- ▷ 레지스터 간접 주소지정방식(Register Indirect Addressing Mode) 오퍼랜드는 레지스터를 가리키며, 해당 레지스터는 실제 데이터의 주소를 가리키는 방식. 한 번의 레지스터 접근과 한 번의 주기억장치 접근이 필요.
- ▷ 상대 주소지정방식(Relative Addressing Mode) 명령어의 주소 부분에 PC(프로그램 카운터) 값을 더해 유효 주소를 구하는 방식
- ▷ 베이스 레지스터 주소지정방식(Base Register Addressing Mode) 명령어의 주소 부분에 베이스 레지스터의 값을 더해 유효주소를 구하는 방식
- ▷ 인덱스 레지스터 주소지정방식(Indexed Addressing Mode) 명령어의 주소 부분에 인덱스 레지스터의 값을 더해 유효주소를 구하는 방식

문 12. 컴퓨터 시스템에서 교착상태의 해결 방안에 대한 설명으로 옳지 않은 것은?

- ① 교착상태가 발생할 가능성을 사전에 없앤다.
- ② 하나의 프로세스만이 한 시점에서 하나의 자원을 사용할 수 있게 한다.
- ③ 교착상태가 탐지되면, 교착상태와 관련된 프로세스와 자원을 시스템으로부터 제거한다.
- ④ 교착상태가 발생할 가능성을 인정하고, 교착상태가 발생하려고 할 때 이를 회피하도록 한다.

교착상태의 발생 조건은 상호배제(Mutual Exclusion), 점유 및 대기(Hold and Wait), 비선점(Preemptive), 환형 대기(Circular Wait)이다.

- ② 한 시점에서 하나의 프로세스만 독점하도록 하는 것은 교착상태의 발생 조건인 비선점에 해당하므로 해결 방안이 아닌 발생 조건에 해당한다.

<오답 체크> ① 교착상태가 발생할 가능성을 사전에 없애는 것은 교착상태 예방(prevention) 기법에 대한 설명이다. 교착상태의 예방 기법은 위에 나열된 교착상태 발생 조건들을 부정하는 것이다.

③ 교착상태의 회복(Recovery) 기법에 대한 설명이다.

④ 교착상태의 회피(Avoidance) 기법에 대한 설명이다. 대표적으로 은행원 알고리즘(Banker's Algorithm)이 있다.

답 ②

문 13. 다음 Java 프로그램은 3의 배수를 제외한 1부터 10까지 정수의 누적 합이 10을 초과하는 최초 시점에서의 합을 출력하는 프로그램이다. ㉠과 ㉡에 들어가는 내용으로 적절한 것은?

```
public class JavaApplication {
    public static void main(String[] args) {
        int i = 0, sum = 0;
        while(i < 10) {
            i++;
            if(i % 3 == 0) ㉠ ;
            if(sum > 10) ㉡ ;
            sum += i;
        }
        System.out.println("sum=" + sum);
    }
}
```

- | | |
|------------|----------|
| ㉠ | ㉡ |
| ① break | goto |
| ② continue | break |
| ③ final | continue |
| ④ return | break |

- ㉠ 3의 배수를 제외한 정수의 합을 구하는 것이므로, 3의 배수일 경우에는 계산을 하지 않고 다음 정수로 넘어가는 식을 작성해야 한다. 단, 이 때 전체 반복문은 빠져 나가지 않아야 하므로, **continue** 문을 사용한다.
- ㉡ 누적 합이 10을 초과하는 경우 반복문을 중지하는 식을 작성해야 한다. 전체 반복문을 빠져 나갈 때는 **break** 문을 사용한다.

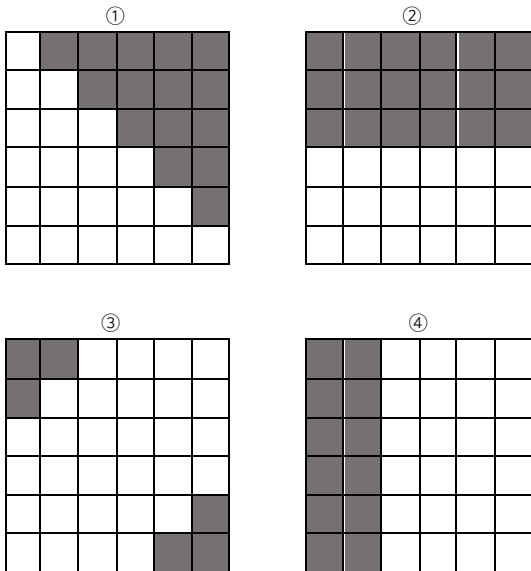
답 ②

문 14. 다음과 같은 압축되지 않은 비트맵 형식의 이미지를 RLE(Run Length Encoding) 방식을 이용하여 압축했을 때 압축률이 가장 작은 것은? (단, 모든 이미지의 가로와 세로의 길이는 동일하고, 가로 방향 우선으로 픽셀을 읽어 처리한다)



RLE(Run Length Encoding) 방식은 가장 간단한 비손실 압축 방법으로, 데이터에서 연속적으로 반복되는 값을 개수로 표기하는 방법이다.

각 보기 이미지들을 간략하게 6X6의 픽셀로 표현을 하면,



(여기서 흰색은 W, 검은색은 B로 표현을 하겠다)

가로 방향 우선으로 픽셀을 처리한다고 하였으므로 윗줄부터 한 줄씩 처리하면 된다.

①번은 첫 줄이 W B B B B B 이므로, W 1개, B 5개가 연속해서 나오므로, W 1 B 5로 표현된다.

두 번째 줄은 W 2 B 4, 세 번째 줄은 W 3 B 3,

네 번째 줄은 W 4 B 2, 다섯 번째 줄은 W 5 B 1,

마지막 여섯 번째 줄은 모두 W 이므로, W 6으로 간단해진다.

따라서 전체 36개의 픽셀을

W 1 B 5 W 2 B 4 W 3 B 3 W 4 B 2 W 5 B 1 W 6

으로 압축할 수 있다.

②번은 가로줄이 모두 같은 색이기 때문에 더 압축해서 간단하게 표현할 수 있는데,

윗 부분 세 줄은 18개의 픽셀이 모두 W이고,

아래 부분 세 줄은 18개의 픽셀이 모두 B이므로,

W 18 B 18 로 압축할 수 있다.

③번과 ④번도 마찬가지로 방법으로

③ **B 2 W 4 B 1 W 22 B 1 W 4 B 2**

④ **B 2 W 4 B 2 W 4 B 2 W 4 B 2 W 4 B 2 W 4 B 2 W 4**

정리하면,

① W 1 B 5 W 2 B 4 W 3 B 3 W 4 B 2 W 5 B 1 W 6

(11 문자로 표현)

② W 18 B 18

(4 문자로 표현)

③ B 2 W 4 B 1 W 22 B 1 W 4 B 2

(7 문자로 표현)

④ B 2 W 4 B 2 W 4 B 2 W 4 B 2 W 4 B 2 W 4 B 2 W 4

(12 문자로 표현)

따라서 압축률이 가장 작은 것은 ④번이다.

나머지는 최소 한 줄은 같은 색으로 이루어진 줄이 있는데, ④번은 줄이 검은색과 흰색으로 이루어져 있어 압축률이 작은 것이다.

반대로 세로 방향 우선 방식일 경우 ④번의 압축률이 가장 크고 ②번의 압축률이 가장 작을 것이다.

(참고로 실제 RLE 압축에선 보통 3개 이하의 압축을 수행하지 않는다. 이유는 W 한 개를 그냥 표현하는 것보다 W 1 으로 표현하면 오히려 길이가 늘어나는 것처럼 비효율적이기 때문)

답 ④

문 15. 다중 스레드(Multithread)에 대한 설명으로 옳은 것만을 모두 고르면?

ㄱ. 스레드는 프로세스보다 더 큰 CPU의 실행 단위이다.
 ㄴ. 단일 CPU 컴퓨터에서 작업을 수행하는 스레드들은 CPU 자원을 공유한다.
 ㄷ. 스레드는 프로세스와 마찬가지로 독립적인 PC(Program Counter)를 가진다.
 ㄹ. 프로세스 간의 문맥교환은 동일 프로세스에 있는 스레드 간의 문맥교환에 비해 비용면에서 효과적이다.

- ① ㄱ, ㄴ ② ㄱ, ㄹ
- ③ ㄴ, ㄷ ④ ㄴ, ㄹ

ㄴ. 같은 프로세스에 속하는 스레드들은 서로 자원과 메모리를 공유한다.
 ㄷ. 각각의 스레드는 자신만의 스택과 레지스터를 가지며, PC(프로그램 카운터) 역시 레지스터에 해당된다.

<오답 체크> ㄱ. 스레드는 프로세스의 제어 부분을 제외하고 실행 부분만 전담하는 개념으로 프로세스보다 작은 실행 단위이다. 경량 프로세스(LWP, Light Weight Process)라고도 하며, 하나의 프로세스에는 여러 개의 스레드가 포함될 수 있다.
 ㄹ. 프로세스보다 스레드가 작은 단위이다. 따라서 프로세스 간의 문맥교환보다 스레드 간의 문맥교환이 비용면에서 효율적이다.

답 ③

문 16. 구매 방법에 따른 소프트웨어 분류에 대한 설명으로 옳은 것은?

- ① 프리웨어(Freeware)는 라이선스 없이 무료로 배포되어, 영리목적 기관에서도 자유롭게 배포할 수 있는 소프트웨어이다.
- ② 라이트웨어(Liteware)는 상용 소프트웨어의 일부 기능만을 사용할 수 있도록 하여, 낮은 가격에 판매되는 소프트웨어이다.
- ③ 오픈소스 소프트웨어(Open Source Software)는 프로그램 소스가 공개되어 있으나, 저작권자의 동의 없는 임의 수정은 불가능하다.
- ④ 셰어웨어(Shareware)는 시범적으로 사용자에게 무료로 제공한 후 일정 기간이 지나면, 유용성에 따라서 구매하도록 하는 소프트웨어이다.

④ 셰어웨어(Shareware)는 일정한 기간 동안 자유롭게 테스트 형식으로 사용해볼 수 있도록 무료로 배포하는 소프트웨어로, 일종의 체험판 같은 것이다. 무료 체험 기간이 지나면 사용자는 필요에 따라 구매하여 계속 사용할 수 있다.

<오답 체크> ① 프리웨어(Freeware)가 무료로 배포되는 소프트웨어는 맞으나, 라이선스가 있으며 저작권 보호도 받는다. 또한 각각의 제품에 따라 개인에겐드 무료로 배포되나 기업이나 기관에는 유료로 판매하는 경우도 있다.
 ② 라이트웨어(Liteware)는 일부 핵심 기능은 제거하고 일부 기능만 사용할 수 있게 하여 배포하는 무료 소프트웨어이다. 잠재 고객들에게 시범삼아 사용해보도록 만든 샘플과 비슷하다. 경우에 따라 라이트웨어를 셰어웨어의 한 종류로 구분하는 경우도 있으니 참고하자. 일부 기능만 사용하도록 하여 낮은 가격에 판매하는 것은, 상용 소프트웨어를 여러 버전으로 만들어 판매하는 것에 해당한다.
 ③ 오픈 소스 소프트웨어(Open Source Software)는 프로그램 소스가 공개되어 있으며 자유롭게 수정 및 재배포가 허용된 소프트웨어이다. 소스가 공개되어 있으나 수정(2차 창작)이 불가능한 경우는 보통 오픈 소스라고 부르지 않는다. 또한 오픈 소스 역시 저작권은 존재한다.

답 ④

문 17. 프로세스 상태 전이에서 준비(Ready) 상태로 전이 되는 상황만을 모두 고르면? (단, 동일한 우선순위의 프로세스가 준비 상태로 한 개 이상 대기하고 있다)

- ㄱ. 실행 상태에 있는 프로세스가 우선순위가 높은 프로세스에 의해 선점되었을 때
- ㄴ. 블록된(Blocked) 상태에 있는 프로세스가 요청한 입출력 작업이 완료되었을 때
- ㄷ. 실행 상태에 있는 프로세스가 작업을 마치지 못하고 시간 할당량을 다 썼을 때

- ① ㄱ, ㄴ ② ㄱ, ㄷ
- ③ ㄴ, ㄷ ④ ㄱ, ㄴ, ㄷ

- ㄱ. 실행중인 프로세스가 우선순위가 높은 다른 프로세스에 의해 CPU를 선점 당할 경우 다시 준비 상태로 돌아간다. 이것을 타임아웃(Timeout)이라고 한다.
- ㄴ. 실행중인 프로세스가 자료의 입출력이 필요하다면 대기 상태로 전이되는데 이것을 블록(Block, 대기)이라고 한다. 그리고 요청한 입출력 작업이 완료되면 블록 상태의 프로세스는 다시 준비 상태로 전이되며, 이것을 깨움(wakeup)이라고 한다.
- ㄷ. 실행중인 프로세스가 할당된 시간 내에 작업을 마치지 못하는 경우 다시 준비 상태로 돌아간다. 이것도 역시 타임아웃(Timeout)이다.

답 ④

문 18. CSMA/CD(Carrier Sense Multiple Access with Collision Detection)에 대한 설명으로 옳은 것만을 고르면?

- ㄱ. 버스형 토폴로지에 많이 사용한다.
- ㄴ. 데이터 전송 시간 및 충돌에 의한 데이터 지연 시간을 정확히 예측할 수 있다.
- ㄷ. 데이터를 전송하기 전에 통신회선의 사용 여부를 확인하고 전송하는 방식이다.
- ㄹ. 전송할 데이터가 없을 때에도 토큰이 통신회선을 회전하면서 점유하는 단점이 있다.

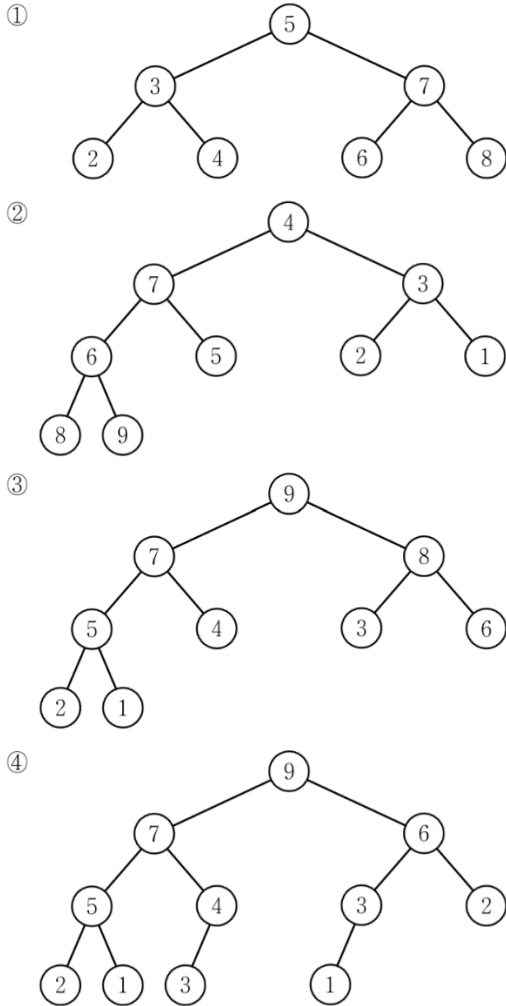
- ① ㄱ, ㄷ ② ㄱ, ㄹ
- ③ ㄴ, ㄷ ④ ㄴ, ㄹ

- ㄱ. CSMA/CD는 버스형과 트리형을 사용한다.
- ㄷ. 보통 간략하게 CSMA/CD는 데이터가 충돌하는 것을 감지하는 방식이고, CSMA/CA는 데이터를 전송하기 전 반송파 감지를 통해 회선의 사용여부를 확인하여 충돌을 회피하는 방식이라고 비교해서 공부했을 것이다.
하지만 이 말이 CSMA/CD는 전송 전에 미리 회선이 사용중인지 점검하지 않는다는 말은 아니다. CSMA/CD에서도 데이터를 전송하기 전 회선의 사용 여부를 점검한다.
- <오답 체크> ㄴ. 충돌이 발생할 경우 랜덤한 시간 동안 기다렸다가 재전송을 시도한다. 따라서 데이터 지연 시간을 정확히 예측하기 어렵다.
- ㄹ. 토큰링(token ring) 방식에 대한 설명이다. 토큰링 방식은 IEEE 802.5 표준 프로토콜이며, 토큰이 통신회선을 회전하면서 토큰을 소유한 단말기만 전송을 수행하는 방식이다. 토큰링 방식에서는 토큰이 끊임없이 회전하며 각 단말기의 데이터 전송 여부를 확인하기 때문에, 전송할 데이터가 없을 때도 통신회선을 회전하며 점유하는 문제가 발생한다.

- ▶ CSMA/CD
IEEE 802.3
이더넷(Ethernet) 전송 프로토콜
주로 버스형, 트리형도 존재
충돌을 감지하여 일정시간 이후 재전송
충돌 반복 시 재전송 시간(Back off)이 2배씩 증가하며, 최대 16회까지 반복
- ▶ CSMA/CA
IEEE 802.11
무선랜 전송 프로토콜
충돌회피 위해 전송 전, 채널의 사용 가능 여부를 확인(반송파 감지)하며, 일정시간 대기 후 전송

답 ①

문 19. 최대 힙 트리(Heap Tree)로 옳은 것은?



힙트리는 완전이진트리의 모양을 갖추어야 하며 부모 노드의 값이 자식 노드의 값보다 크거나 같아야 한다.

- <오답 체크> ① 최상위 노드의 값은 5인데, 오른쪽 자식 노드의 값이 5보다 큰 7이므로 최대 힙트리 조건에 어긋난다.
 ② 최상위 노드의 값은 4인데, 왼쪽 자식 노드의 값이 4보다 7이므로 최대 힙트리 조건에 어긋난다.
 ④ 완전 이진트리가 되기 위해서는 자식 노드들을 왼쪽부터 짝 채워야 한다. 가장 끝에 있는 노드 1을 왼쪽으로 옮겨 4번 노드의 오른쪽 자식 노드로 붙여야 한다.

답 ③

문 20. 다음 식과 논리적으로 같은 것은?

$$(x+y \geq z \text{ AND } (x+y \geq z \text{ OR } x-y \leq z) \text{ AND } x-y > z) \text{ OR } x + y < z$$

- ① $x + y < z$
- ② $x - y > z$
- ③ $x + y \geq z \text{ OR } x - y \leq z$
- ④ $x + y < z \text{ OR } x - y > z$

식이 굉장히 복잡해 보이므로, 먼저 치환을 하여 간단히 표현하는 것이 중요하다.

$x+y \geq z \Rightarrow A$ 로
 $x-y \leq z \Rightarrow B$ 로 치환하면,
 $x + y < z \Rightarrow A$ 의 여집합(A')
 $x - y > z \Rightarrow B$ 의 여집합(B')으로 표현이 가능하다.
 그리고 AND는 \cap , OR은 \cup 과 같다.

따라서 전체 식은

$$(A \cap (A \cup B) \cap B') \cup A'$$

= $(A \cap B') \cup A'$ ← 흡수법칙
 = $(A \cup A') \cap (B' \cup A')$ ← 분배법칙
 = $1 \cap (B' \cup A')$
 = $A' \cup B'$

따라서 $x + y < z \text{ OR } x - y > z$ 이 된다.

답 ④