2017년 지방직 하반기 9급 컴퓨터일반 풀이

by 호이호이꿀떡

정답 체크

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

문 1. 주기억장치로 사용될 수 없는 기억장치는?

① EPROM
② 블루레이(Blu-ray) 디스크
③ SRAM
④ DRAM

답 ②

② 주기억장치는 컴퓨터가 처리할 프로그램이나 데이터를 일시적으로 저장하는 기억장치로, 기억된 데이터를 읽을 수만 있는 ROM(Read Only Memory)과 읽고 쓸 수 있는 RAM(Random Access Memory)이 있다.
블루레이 디스크는 CD와 DVD보다 용량이 큰 휴대용 저장매체로, 보조기억장치에 해당한다.

<오답 체크> ① **EPROM**(Erasable PROM)
ROM의 한 종류로, 자외선을 이용하여 기억된 내용을 지우고 다시 기록할 수 있는 롬이다.

③ **SRAM**(Static RAM, 정적램)
전원이 공급되는 한 그 내용이 계속 보존되는 램으로 DRAM보다 빠르지만, 전력소모가 많고 비싸다.

④ **DRAM**(Dynamic RAM, 동적램)
데이터를 각기 분리된 축전기(Capacitor)에 저장하는 기억장치로, 데이터를 유지하기 위해 주기적인 충전이 필요하다. SRAM보다 느리지만, 전력소모가 적고 저렴하다.

문 2. 스택(stack)에 대한 설명으로 옳지 않은 것은?

① 임의의 위치에서 데이터의 삽입과 삭제가 가능하다.
② 연결 리스트(linked list)를 사용하여 구현할 수 있다.
③ 푸시(push) 연산에 의해 데이터를 삽입한다.
④ 가장 나중에 삽입된 데이터가 가장 먼저 삭제된다.

답 ①

① 스택(Stack)은 리스트의 한쪽 끝으로만 데이터의 삽입, 삭제 작업이 이루어지는 자료 구조이다. 데이터의 삽입과 삭제가 한 쪽 끝에서만 가능하기 때문에 임의의 위치에 데이터를 삽입하거나 삭제하는 것은 불가능하다.

<오답 체크> ③ 스택에 데이터를 삽입할 때는 푸시(Push) 연산, 데이터를 삭제할 때는 팝(Pop) 연산을 수행한다.

④ 스택은 후입선출(LIFO, Last In First Out) 방식을 통해, 나중에 삽입된 데이터가 먼저 삭제된다.

문 3. 통신 시스템에서 아날로그 신호를 디지털 신호로 변환하는 과정 중 시간적으로 연속적인 아날로그 신호로부터 신호 값을 일정한 시간 간격으로 추출하는 단계는?

① 표본화(sampling)
② 부호화(encoding)
③ 복호화(decoding)
④ 양자화(quantization)

답 ①

아날로그 신호를 디지털 신호로 변환할 때는 **표본화 -> 양자화 -> 부호화**의 과정을 거친다.

① **표본화**(sampling) 일정한 간격으로 아날로그 신호의 값을 추출하는 과정
④ **양자화**(Quantization 추출한 샘플링 신호의 레벨을 몇 단계로 나눠 대푯값으로 변환하는 과정
아날로그 값이 3.1415926535... 일 때, 이것을 약속한 디지털 단위(예를 들어, 0.01) 에 맞춰 3.14로 변환하는 것을 의미한다.
② **부호화**(Encoding) 양자화로 나눈 값을 이진수로 변환하는 과정
③ **복호화**(Decoding) 디지털 신호(PCM)을 PAM 신호로 돌리는 과정

문 4. OSI 참조 모델에서 데이터 링크 계층의 프로토콜 데이터 단위(PDU : Protocol Data Unit)는?

① 비트(bit)
② 프레임(frame)
③ 패킷(packet)
④ 메시지(message)

답 ②

각 계층별 PDU(Process Data Unit)
1. 물리 계층 : **비트**(bits), 1계층의 PDU를 비트로 보기도 하고, 그냥 하나의 신호로 보기도 한다.
2. 데이터 링크 계층 : **프레임**(frames)
3. 네트워크 계층 : **패킷**(packets)
4. 전송 계층 : **세그먼트**(segments)
5. 세션 계층 ~ 7. 응용 계층 : **메시지**(messages) 또는 **데이터**(data)

문 5. 다음 C 프로그램의 실행 결과는?

```
#include <stdio.h>

int main(void)
{
    int i;
    char buf[]="123456789012345678901234567890";
    char *str, ch;

    str = buf;
    for(i = 0; i <= 20; i += 4)
    {
        printf("%c ", *str++);
        ch = *++str;
    }
    return 0;
}
```

① 1 3 5 7 9 1 ② 1 5 9 3 7 1
③ 2 4 6 8 0 2 ④ 2 6 0 4 8 2

답 ①

(이번 문제는 쓸데없는 변수 ch와 헷갈릴 만한 숫자 4를 집어넣어 실수를 유도하였으니 주의!)

str = buf; 이라고 대입하였기 때문에 str 변수는 buf와 같은 배열을 가리키고 제일 앞 문자인 1을 가리키게 된다.

for문은 처음 i = 0일 때로 시작한다.
printf문에서 *str++이라고 되었기 때문에, 먼저 *str에 있는 값을 우선 출력을 한 뒤, 1 증가를 하게 된다.
따라서 먼저 **'1'이 출력**된 뒤, str 포인터 변수는 1 다음의 문자인 2를 가리키는 상태가 된다.

그 다음 문장은 ch = *++str; 인데, 이 부분에서 ch 변수는 생각할 필요가 없다. 왜냐하면 ch 변수가 출력이나 다른 연산에 쓰이지 않기 때문이다. 여기에서는 str 포인터가 1이 증가한다는 것만 기억하면 된다.
(이제 str 포인터는 배열에서 세 번째인 3을 가리키는 상태가 된다.
참고로 이번 연산에선 str 변수를 먼저 증가시키고 ch 변수에 값을 대입하였기 때문에 ch 변수에는 3이 들어간다.)

그 다음 i는 4가 증가해 4가 되며, 마찬가지로 20보다 작거나 같기 때문에 for문을 다시 한 번 실행한다.

printf문에서 첫 번째와 마찬가지로, str이 가리키는 문자 하나를 출력한 뒤 str포인터를 1 증가시킨다.
따라서 **'3'이 출력**된 뒤, str은 4로 증가한다.
그 다음 줄에서 역시 마찬가지로 str 포인터가 1 증가한 뒤, 그 값을 ch변수에 집어넣는다.
(str 포인터 값은 5의 주소)

이처럼 for문을 한 번 실행할 때마다 먼저 str이 가라키는 문자 하나를 출력한 뒤, str 포인터 값은 2 증가한다.

다음 i는 4 증가해 8이 되며, 역시 for문 실행
printf문을 통해 '5'출력 후 str 포인터는 7 가리킴

이렇게 i=20일 될 때까지 for문이 반복되며, for문은 총 6번이 반복된다.
(20 ÷ 4 = 5라고 5번이라고 실수하지 말 것. 0부터 시작해서 같거나 작을 때까지이다.)

따라서 전체 출력되는 값은 처음 '1'부터 2칸씩 건너 뛰어 총 6개의 숫자를 출력하면 된다.

1 3 5 7 9 1 출력

문 6. 데이터 통신의 오류 검사 방식 중 다항식 코드를 사용하며 집단(burst) 오류 검출에 적합한 방식은?

① FEC(Forward Error Correction)
② 단일 패리티 비트(parity bit) 검사
③ 블록 합(block sum) 검사
④ CRC(Cyclic Redundancy Check)

답 ④

④ **CRC**(Cyclic Redundancy Check, 순환 중복 검사)
다항식 코드를 사용하여 오류를 검출하는 방식으로, 임의의 비트 열 검사에 사용
집단 오류를 검출할 수 있꼬, 검출률이 높음
동기식 전송에 주로 사용
HDLC 프레임의 FCS에 사용

<오답 체크> ① **FEC**(Forward Error Correction, 전진 에러 수정)
송신측이 전송할 문자나 프레임에 부가적 정보(Redundancy)를 첨가하여 전송하고, 수신측이 에러를 발견시 이 부가적 정보를 이용해 스스로 에러 검출 및 정정을 하는 방식

② 단일 **패리티** 비트(parity bit) 검사
1비트의 오류만 검출 및 수정이 가능

③ **블록합**(block sum) 검사
수직 패리티와 수평 패리티를 혼합한 방법으로, 2비트의 오류를 검출할 수 있다.(2비트 오류의 발생 여부는 알 수 있으나, 정확한 위치는 알 수 없다.)
그러나 연속된 두 개의 문자에서 같은 위치의 두 개의 비트들이 오류인 경우는 검출이 불가능하다.
블록합 검사도 2비트의 오류를 검출할 수 있기 때문에 집단오류를 할 수 있는 거 아니냐는 의문도 들 수 있는데, **집단 오류**(burst error)는 데이터 두 개 이상의 연속적인 비트 오류를 의미한다. 인접하지 않은 두 개 이상의 오류는 **다중 비트 오류**(multiple-bit error)라고 한다.

문 7. 컴퓨터의 수 표현에 대한 설명으로 옳지 않은 것은?

① 기본적으로 0과 1을 사용하여 수를 표현한다.
② 1의 보수 표기법을 사용하여 부호 있는(signed) 2진 정수를 표현할 수 있다.
③ IEEE 754 표준 부동소수점 수는 부호(sign), 지수(exponent), 소수(fraction)로 구성된다.
④ IEEE 754 표준 단정도(single precision) 부동소수점 수가 표현할 수 있는 값의 개수는 2의 보수 표기법에 의한 32비트의 부호 있는 2진 정수보다 많다.

답 ④

④ (4번 선지는 미리 외우지 않았다면 수학적 개념 경우의 수를 생각해서 풀 수 있다.)
IEEE 754 표준 단정도는 **1.F의 형태**로, 1비트의 부호, 8비트의 지수, 23비트의 가수(F)로 구성된다.
표현 가능한 수의 개수를 계산해보면,
부호는 양수, 음수 2가지,
가수부의 구성이 2^{23}가지가 나오고,
지수부에 따른 소수점의 위치가 256(=2^8)곳이다.
(127 바이어스법을 이용하여, -127제곱부터 128제곱까지)
따라서 총 표현 가능한 수의 개수는

$$\mathbf{2 \times 2^{23} \times 2^8 = 2^{32}}$$ 가지이다.

2의 보수법으로 표현 가능한 32비트 정수는
-2^{n-1} 부터 $2^{n-1}-1$ 까지 총 $\mathbf{2^{32}}$ 가지이다.

2의 보수법에 의한 표기법은 0이 한 가지이기 때문에 2^{32} 가지가 표현이 가능하다.
(1의 보수법이나 부호화 절대값 표기법은 0이 두 가지이기 때문에 $2^{32}-1$ 가지를 표현할 수 있다.)

IEEE 754 부동소수점 표현 방식은 지수를 이용하여 수의 표현 범위가 넓은 것일 뿐, 표현 가능한 수의 개수가 많아지는 건 아니다. 0과 1을 표현할 수 있는 비트 32개를 이용해 2^{32} 가지 이상을 표현하는 것은 불가능하다.
우리가 아라비아 숫자 10가지를 이용하여 8자리 수를 표현한다면, 아무리 숫자를 다양하게 나열해도 10^8(=1억) 가지 이상은 표현 불가능한 것과 마찬가지이다.

<오답 체크> ③ IEEE 754 표준 부동소수점 표현 방식은 부호(sign), 지수(exponent), 소수(fraction, =가수)로 구성된다.
단일 정밀도(단정도, single-precision) 방식은 1비트의 부호, 8비트의 지수부, 23비트의 가수부로 구성되며,
복수 정밀도(배정도, double-precision) 방식은 1비트의 부호, 11비트의 지수부, 52비트의 가수부로 구성된다.

문 8. 어떤 컴퓨터에서 프로그램 P를 실행할 때, 실행시간 중 60 %의 시간이 연산 A를 실행하는데 소요된다. 다른 조건의 변화 없이 연산 A만을 n배 빠르게 실행하도록 컴퓨터의 성능을 향상시킨 후 프로그램 P에 대한 실행시간이 50 % 감소했다면, n의 값은? (단, 실행시간은 프로그램 P만 실행하여 측정한다)

① 2
② 4
③ 6
④ 10

답 ③

프로그램 P의 전체 실행시간 100% 중 A 연산을 제외한 40%를 B라고 부르겠다. B는 변화가 없는데, P의 전체 실행시간이 50%가 감소했다면, A 연산의 실행시간 10%로 줄어들어야 가능하다.
변경 전 A(60%) + B(40%) = 100%
변경 후 A(10%) + B(40%) = 50%
따라서 A의 연산이 6배 빨라졌다는 계산이 된다.

(암달의 법칙으로 계산이 가능하다.)

문 9. 데이터베이스 무결성 제약조건에 대한 설명으로 옳지 않은 것은?

① 무결성 제약조건은 사용자에 의한 데이터베이스 갱신이 데이터의 일관성을 손상하지 않도록 보장하는 데에 사용된다.
② DBMS는 무결성 제약조건을 검사하는 기능을 가진다.
③ 도메인 무결성 제약조건은 기본 키가 널(NULL) 값을 가질 수 없고 튜플을 유일하게 식별해야 한다는 것이다.
④ 참조 무결성 제약조건은 릴레이션 사이의 참조를 위해 사용되는 외래키에 대한 것이다.

답 ③

③ 릴레이션의 기본 키를 구성하는 어떤 애트리뷰트도 널(NULL)값을 가질 수 없고 튜플을 유일하게 식별해야 한다는 것은 **개체 무결성 제약조건**(Entity Integrity Constraint)에 대한 설명이다.
도메인 무결성 제약조건(Domain Constraint)은 각 애트리뷰트 값이 반드시 원자값이어야 하며, 허용된 유형의 데이터 값들만 입력이 가능하다는 것이다.

<오답 체크> ④ **참조 무결성 제약조건**(Referential Integrity Constraint)
두 개의 릴레이션이 기본키·외래키를 통해 참조 관계를 형성할 경우, 참조하는 외래키의 값은 참조되는 릴레이션에 기본키로 존재해야 한다는 것이다.

◈기타 제약조건
▷ **키 제약조건**(Key Constraint)
애트리뷰트에 중복된 값이 존재해서는 안 된다는 것
▷ **사용자 정의 무결성 제약조건**(user define integrity constraint)
사용자가 다른 무결성 범주에 포함되지 않는 특정 업무 규칙을 정의하여 사용하는 것

문 10. 다음은 2진 입력 A, B, C와 2진 출력 X, Y, Z 사이의 관계를 나타낸 것이다. X, Y, Z에 대한 출력 함수를 옳게 짝지은 것은?

> ○ 입력 C = 0일 때, 출력 X = 0, Y = 0, Z = 0
> ○ 입력 B = 0이고 C = 1일 때, 출력 X = 0, Y = 0, Z = 1
> ○ 입력 B = 1이고 C = 1일 때, 출력 X = A, Y = B, Z = C

① X = AC, Y = BC, Z = C
② X = A'C, Y = B'C, Z = C'
③ X = ABC, Y = BC, Z = C
④ X = A'B'C, Y = B'C, Z = C'

답 ③

이 문제는 카르노맵이나 벤다이어그램을 통해서도 풀 수 있지만, 구해야 하는 출력값이 3개이기 때문에 시간이 오래 걸린다.
가장 빨리 풀 수 있는 방법은 출력이 1이 나오는 것들만 빠르게 캐치해서 불 대수를 이용해 푸는 방법이다.

▷ X의 풀이
C = 0이면, X = 0
C = 1이라도 B = 0이면, X = 0
C = 1이고 B = 1이면, X = A
(A = 1이면 X = 1, A = 0이면 X = 0)
즉, X가 1이 나오는 경우는 A와 B와 C가 모두 1일 때이다.
X = ABC

▷ Y의 풀이
C = 0이면, Y = 0
C = 1이라도 B = 0이면, Y = 0
C = 1이고 B = 1이면, Y = B(이 때 A는 무관하다)
즉, Y가 1이 나오는 경우는 B와 C가 1 동시에 1일 때이다.
(A는 언급되지 않았기 때문에 무관하다.)
Y = BC

▷ Z의 풀이
C = 0이면, Z = 0
B = 0이고 C = 1이면, Z = 1
B = 1이고 C = 1이면, Z = C(Z = 1이라는 말)
즉, B가 0이든 1이든 C가 1이면 Z는 1이 나온다.
(A는 언급되지 않았기 때문에 무관하다.)
Z = C

문 11. 다음 시나리오에서 괄호 안에 들어갈 가장 적합한 정보 서비스 유형은?

> 회사원 갑이 출장지로 자동차를 운전하여 가던 중, 휘발유가 부족한 것을 알았다. 그래서 ( ) 유형의 앱을 실행하여 주변 주유소를 검색하고 가장 가까운 주유소를 선택하였다.

① 빅데이터 서비스
② 클라우드 서비스
③ 가상현실 서비스
④ 위치기반 서비스

답 ④

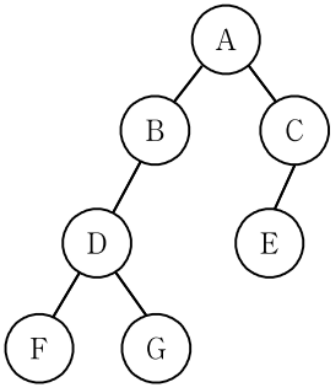
④ **위치 기반 서비스(LBS**, Location-based service)는 무선 인터넷을 통해 사용자의 변경되는 위치에 따른 특정 정보를 제공하는 무선 콘텐츠 서비스

<오답 체크> ① **빅 데이터**(Big Data)
기존 데이터베이스 관리도구의 능력을 넘어서는 대량의 정형 또는 비정형의 데이터 집합으로부터 가치를 추출하고 결과를 분석하는 기술
빅 데이터의 V3 : Volume(규모), Velocity(속도), Variety(다양성)

② **클라우드**(Cloud)
클라우드에 대해 정확하게 용어 정의가 된 것은 없다. 보통 인터넷 기반 컴퓨팅의 일종으로 정보를 네트워크에 연결된 다른 컴퓨터를 통해 데이터를 처리하는 기술을 의미한다.

③ **가상현실(VR**, virtual reality)은 컴퓨터 등을 사용한 인공적인 기술로 만들어낸, 실제와 유사하지만 실제가 아닌 어떤 특정한 환경 혹은 그 기술 자체

문 12. 다음 이진 트리의 노드를 중위 순회(inorder traversal)할 때, 4, 5, 6번째 방문 노드를 순서대로 바르게 나열한 것은?



① A, B, C
② B, A, E
③ B, E, C
④ F, G, C

답 ②

중위 순회 방문 노드 순서
F - D - G - **B** - **A** - **E** - C

문 13. CASE(Computer-Aided Software Engineering)에 대한 설명으로 옳지 않은 것은?

① 소프트웨어 품질을 효율적으로 제어할 수 있다.
② 소프트웨어 유지보수 비용을 절감할 수 있다.
③ 통합 CASE 도구는 소프트웨어 개발 주기의 전체 과정을 지원한다.
④ 하위 CASE 도구는 프로젝트 계획 수립 및 요구 분석 과정을 지원한다.

답 ④

④ 프로젝트 계획 수립 및 요구 분석, 설계 등은 상위 CASE 도구에서 지원한다.
하위 CASE 도구는 코딩 기능을 중심으로 지원한다.

◈ CASE(Computer Aided Software Engineering)
소프트웨어 개발 생명 주기 전체 과정에서 정형화된 구조 및 메커니즘을 소프트웨어 개발에 적용하여 소프트웨어 개발을 자동화하여 생산성을 향상시키고자 하는 공학기법이다.

▷ 상위 CASE
계획과 분석, 설계 단계를 지원한다. 기업이나 조직을 파악하고 기술하기 위하여 다이아그래밍, 명세서 등의 기능을 제공한다. 국내에서 판매되고 있는 CASE Tool로는 System Architect, ER-WIN 등이 있으나 하위 CASE의 성격이 보강되면서 점점 통합 CASE화하는 추세에 있다.

▷ 중위 CASE
사용자가 정보에 따른 문제점을 분석하고 이들에 대한 해결책을 찾도록 다이아그래밍과 사전(Dictionary) 구성요소로 이루어진다.

▷ 하위 CASE
코드를 생성하기 위한 기능이 대부분이며 시스템 명세서를 작성할 수 있다.

▷ 통합 CASE
전체 라이프사이클을 포괄하여 지원하는 CASE로 대개 CASE라 하면 통합 CASE를 가리킨다. 국내에서 시판되는 통합 CASE로는 COOL, ROSE 등이 있다.

문 14. 2의 보수로 표현된 n비트의 부호 있는(signed) 2진 정수가 표현할 수 있는 최댓값과 최솟값의 합은?

① -1
② 0
③ 1
④ 2^{n-1}

답 ①

n비트의 2의 보수로 표현 가능한 수의 범위는
$-2^{n-1} \sim 2^{n-1}-1$
최댓값 - 2^{n-1}
최솟값 $2^{n-1}-1$
따라서 최댓값과 최솟값의 합은 -1이다.

문 15. 다음 표는 프로세스들의 대기 시간과 예상되는 서비스 시간을 나타낸 것이다. HRRN(Highest Response Ratio Next) 스케줄링 알고리즘을 사용할 때, 우선순위가 가장 높은 프로세스는?

(단위 : 밀리초)

프로세스	대기 시간	서비스 시간
P1	10	5
P2	12	4
P3	8	12
P4	15	3

① P1
② P2
③ P3
④ P4

답 ④

스케줄링에서 서비스 시간 대비 대기 시간이 길수록 우선순위가 높아진다.

각 프로세스의 우선순위는 다음과 같다.
(대기 시간 ÷ 서비스 시간)
P1 = 10 ÷ 5 = 2
P2 = 12 ÷ 4 = 3
P3 = 8 ÷ 12 = 0.67
P4 = 15 ÷ 3 = 5

우선순위 계산 공식을
(대기 시간 + 서비스 시간) / 서비스 시간
으로 표현하기도 하는데, 우선 순위는 동일하다.

문 16. 비트맵 방식의 이미지 파일 형식 중 압축을 하지 않기 때문에 파일 크기가 크다는 단점을 가진 것은?

① AI
② BMP
③ PNG
④ JPEG

답 ②

② BMP: 압축이 하지 않기 때문에 파일 크기가 매우 크지만, 그만큼 화질이 가장 선명하다.

<오답 체크> ① AI: 어도비 일러스트레이터 프로그램에서 사용되는 그림 파일로, 벡터 방식의 이미지 파일이다.

③ PNG: GIF보다 압축률이 더 높다.
24비트 트루 컬러를 지원하며, 투명층을 지원한다.

④ JPEG: 24비트의 색상을 지원하는 고화질이며 압축률이 높아서 용량이 작다. 손실 압축 알고리즘을 이용하므로 압축 비율을 높게 지정하면 화질이 많이 저하된다.

GIF: 256색만 지원하며, LZW 압축 방법을 사용하여 파일의 크기를 1.5:1에서 2:1의 비율로 압축률이 높다.
투명화 지원, GIF 애니메이션 지원

▷ **비트맵**(Bitmap) 방식
하나의 이미지를 여러 개의 픽셀로 나누어 각 픽셀마다 색을 지정하는 이미지를 표현하는 방식
gif, jpg, bmp, tif, png, pcx 등

▷ **벡터**(Vector) 방식
점과 점을 연결함으로써 수학적 함수 관계에 의해 이미지를 표현하는 방식
dr, cdr, wmf, ai 등

문 17. 프로세스 P가 수행 준비는 되어 있으나 다른 프로세스들이 더 우선적으로 수행되어, 프로세스 P가 계속적으로 CPU 할당을 기다리면서 수행되지 못하는 상태는?

① 교착상태(deadlock)
② 기아상태(starvation)
③ 경쟁상태(race condition)
④ 상호배제(mutual exclusion)

답 ②

② **기아상태**(starvation)
해당 프로세스가 다른 프로세스들에 의해 우선순위가 밀려 끊임없이 필요한 컴퓨터 자원을 할당받지 못하는 상황으로, 스케줄링이나 상호 배제 알고리즘의 오류에 기인하지만 자원 누수에 의해 일어날 수도 있으며 포크 폭탄과 같은 서비스 거부 공격을 통해 고의적으로 발생할 수도 있다.

<오답 체크> ① **교착상태**(deadlock)
두 개 이상의 작업이 서로 상대방의 작업이 끝나기만을 기다리고 있기 때문에 무한 대기상태가 되어있는 상태

③ **경쟁상태**(race condition)
공유 자원에 대해 여러 개의 프로세스가 동시에 접근을 시도할 때 접근의 타이밍이나 순서 등에 따라 결과값에 영향을 줄 수 있는 상태를 말한다.
여러 개의 프로세스가 동시에 접근할 때 자료의 일관성을 해치는 결과가 나타날 수 있다. 이를 방지하기 위해서는 프로세스 협력 기법이 필요하다.

④ **상호배제**(mutual exclusion)
여러 프로세스가 공유 불가능한 자원에 동시에 접근하여 사용하는 것을 피하기 위해 사용되는 알고리즘으로, 임계 구역(critical section)으로 불리는 코드 영역에 의해 구현된다.

문 18. 다음은 후위(postfix) 표기 수식을 스택을 이용하여 계산하는 과정 중에 처리되지 않고 남아 있는 수식과 스택의 상태를 나타낸 것이다. 수식 계산을 완료했을 때의 최종 결과 값은? (단, 수식에서 연산자는 +, *이다)

○ 남아 있는 수식 : * 4 * 5 + 6 +

○ 스택의 상태 :

① 35
② 68
③ 126
④ 466

답 ①

후위 표기 방식의 스택이기 때문에, 뒤에 연산자가 들어오면 앞의 두 수를 연산 기호에 맞춰 계산하면 된다.

* 삽입) 2 3 ***** => 6
4 삽입) 6 **4** <- 아직 대기
* 삽입) 6 4 ***** => 24
5 삽입) 24 **5** <- 아직 대기
\+ 삽입) 24 5 **+** => 29
6 삽입) 29 6 <- 아직 대기
\+ 삽입) 29 6 **+** => 35

문 19. 다음 Java 프로그램이 실행될 수 있도록 ㉠ ~ ㉢ 을 옳게 짝지은 것은?

```
import java.util.Stack;

public class StackDemo1 {
    public static void main(String[] args) {
        Stack< ㉠ > stack = ㉡ Stack<>();
        stack.push("java");
        stack.push("stack");
        stack.push("demo");

        ㉢ popResult = stack.pop();
        System.out.println(popResult);

        popResult = stack.pop();
        System.out.println(popResult);

        popResult = stack.pop();
        System.out.println(popResult);
    }
}
```

	㉠	㉡	㉢
①	String	create	String
②	Object	create	String
③	Object	new	char
④	String	new	Object

답 ④

㉠ Stack< ㉠ > stack = ㉡ Stack<>();
이 부분에서 ㉠은 스택에 들어갈 변수형을 지정하는 곳이다.
아랫줄을 보면
stack.push("java"); stack.push("stack"); stack.push("demo");
이렇게 스택에는 string 문자열이 삽입(push)되고 있으므로,
㉠ 위치에 String<**String**>이라고 지정해준다.

㉡ 자바에서 Stack은 클래스로 구현되어 있는데, 이를 사용하기 위해서는 새로운 객체를 생성해서 사용해야 한다. 객체를 생성하는 연산자는 new 예약어이므로,
㉡ 위치에 **new** Stack<>;이라고 객체를 생성한다.

㉢ 뒤의 popResult = stack.pop();과
System.out.println(popResult);를 보면,
popResult라는 변수에 stack으로부터 pop되어 나오는 값을 넘겨받은 뒤 출력하는 것임을 알 수 있다.
그런데 윗부분 어디에서도 popResult 변수를 선언한 부분이 없기 때문에, ㉢ 위치에서 변수를 선언하고 생성해야 한다. 넘겨받을 값의 데이터 형식이 문자열이기 때문에 정확한 답은 String이지만, Object는 모든 형식을 넘겨받을 수 있기 때문에 **Object**를 사용할 수도 있다.
추가로 Object는 모든 클래스의 부모이며 모든 형식을 넘겨받을 수 있기 때문에, 프로그래밍을 할 때 변수에 어떤 형식의 값이 들어갈지 모르는 상황에서는 Object를 사용해 우선 데이터를 넘겨받은 다음에 데이터 형식을 확인할 때 쓰기도 한다.

문 20. 캐시 기억장치에 대한 설명으로 옳지 않은 것은?

① 명령어 캐시 기억장치와 데이터 캐시 기억장치로 분리된 구조를 가질 수 있다.
② 2개 이상의 단계(level)를 가지는 다단계 구조를 가질 수 있다.
③ 직접 사상(direct mapping) 방식을 사용할 경우, 적절한 교체(replacement) 알고리즘이 필요하다.
④ 쓰기 버퍼(write buffer)는 즉시 쓰기(write-through) 캐시 기억장치에서 쓰기 동작이 오래 걸리는 문제를 개선할 수 있다.

답 ③

③ 직접 사상(direct mapping) 방식은 메모리에 있는 블록들이 캐시의 지정된 블록 프레임으로만 적재될 수 있는 방식이다.
교체 알고리즘은 페이지 부재(fault)가 발생할 때 새로운 페이지를 현재의 어떤 페이지와 교체할지(어느 곳으로 적재할지)를 결정하는 방법이다. 직접 사상은 각각의 메모리 블록들이 고정된 한 곳으로 적재되기 때문에, 어떤 페이지와 교체할지 고민할 필요가 없다.

<오답 체크> ④ 쓰기 버퍼(write buffer)는 메모리에 기록하기 위해 기다리는 동안 데이터를 임시적으로 저장해두는 공간이다.
즉시 쓰기 방식에서 데이터가 변경될 때마다 매번 기억장치에 데이터를 기록해야 한다면, 그 때마다 CPU는 다른 명령어의 처리를 정지하고 데이터 쓰기를 수행해야 한다. 쓰기 버퍼는 이러한 CPU의 낭비를 줄이기 위해 데이터 변경 사항을 임시 저장해두었다가, CPU가 데이터 버스를 이용하지 않는 시간에 기억장치에 쓰기 동작을 수행한다.