

모의고사 정답

[문제 1]

의존

※ 답안 작성 시 주의 사항

한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 2]

- ① private
- ② #
- ③ salaryCal

※ 답안 작성 시 주의 사항

프로그래밍 언어에서 사용하는 메소드의 이름은 대소문자를 구분하기 때문에 메소드 이름을 작성할 때는 대소문자를 구분해서 정확히 작성해야 합니다.

[해설]

접근제어자는 속성과 오퍼레이션에 동일하게 적용되며, 표현법은 다음과 같습니다.

접근제어자	표현법	내용
public	+	어떤 클래스에서라도 접근이 가능합니다.
private	-	해당 클래스 내부에서만 접근이 가능합니다.
protected	#	동일 패키지 내의 클래스 또는 해당 클래스를 상속 받은 외부 패키지의 클래스에서 접근이 가능합니다.
package	~	동일 패키지 내부에 있는 클래스에서만 접근이 가능합니다.

[문제 3]

제3정규화

[해설]

<제품> 테이블에는 다음과 같은 함수적 종속이 존재합니다.

- 제품코드 → 제조사, 담당자
- 제조사 → 담당자

<제품> 테이블은 제조사와 담당자가 기본키인 제품코드에 대해 완전 함수적 종속이므로 제 2정규형입니다. 그러나 제조사가 제품코드에 함수적 종속이고, 담당자가 제조사에 함수적 종속이므로 담당자는 기본키인 제품코드에 대해 이행적 함수적 종속을 만족합니다. 즉 제품코드 → 제조사이고, 제조사 → 담당자이므로 제품코드 → 담당자는 이행적 함수적 종속이므로 <제품> 테이블은 제 3정규형이 아닙니다. 문제의 그림은 <제품> 테이블에서 이행적 함수적 종속(즉 제품코드 → 담당자)을 제거하여 <제품목록> 테이블과 <제조사 목록> 테이블로 무손실 분해하는 제3정규화 과정입니다.

[문제 4]

다음 중 하나를 쓰면 됩니다.

WSDL, Web Services Description Language

※ 답안 작성 시 주의 사항

한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 5]

- ① 자료
- ② 기능적
- ③ 3
- ④ 2

[문제 6]

- 직접 연계 방식 : DB Link, API/Open API, DB Connection, JDBC
- 간접 연계 방식 : ESB, Socket, Web Service

[문제 7]

다음 중 하나를 쓰면 됩니다.

스토리보드, Story Board

[문제 8]

- ① 화이트박스
- ② 블랙박스
- ③ 하향식
- ④ 상향식

[문제 9]

- (1) SELECT 학번, 이름 FROM 학생 WHERE 학번 LIKE '19%' AND 나이 = 21;
- (2) SELECT COUNT(*) FROM 학생 WHERE 학번 NOT IN (SELECT 학번 FROM 성적);
- (3) UPDATE 성적 SET 태도 = 80 WHERE 태도 < 80;

[풀이]

(1) <SQL문>

SELECT 학번, 이름	'학번'과 '이름'을 표시한다.
FROM 학생	<학생> 테이블을 대상으로 검색한다.
WHERE 학번 LIKE '19%'	'학번'이 19로 시작하고,
AND 나이 = 21;	'나이'가 21인 튜플을 대상으로 한다.

<결과>

학번	이름
190098E	우길산

(2) <SQL문>

SELECT COUNT(*)	튜플의 개수를 표시한다.
FROM 학생	<학생> 테이블을 대상으로 검색한다.
WHERE 학번 NOT IN (<학생> 테이블의 '학번'이 NOT IN 다음에 쓰인 하위 질의의 결과
	에 없는 자료만을 대상으로 한다.
SELECT 학번	'학번'을 표시한다.
FROM 성적);	<성적> 테이블에서 검색한다. 즉 <성적> 테이블의 '학번'을 표시한다.

<결과>

<학생> 테이블에만 있고 <성적> 테이블에는 없는 '학번'이 "180892B", "177720B" 2개이므로 결과는 2입니다.

COUNT(*)
2

(3) <SQL문>

UPDATE 성적	<성적> 테이블을 갱신하라.
SET 태도 = 80	'태도'를 80으로 갱신하라.
WHERE 태도 < 80;	'태도'가 80 미만인 튜플만을 대상으로 하라.

<결과>

갱신된 <성적> 테이블의 자료

학번	공통	교양	태도
197720E	95	90	93
207620E	85	90	88
200098E	80	75	80
190098E	65	80	80

[문제 10]

- ① OPEN P_CUR
- ② P_CUR%NOTFOUND
- ③ P_DUE

※ 답안 작성 시 주의 사항

프로그래밍 언어나 SQL에서 사용하는 변수의 이름은 대소문자를 구분하기 때문에 변수 이름을 작성할 때는 대소문자를 구분해서 정확히 작성해야 합니다.

[해설]

```
① CREATE OR REPLACE PROCEDURE PAY IS
② P_CODE ORDERS.CODE%TYPE;
③ P_DUE ORDERS.DUE%TYPE;
④ CURSOR P_CUR IS SELECT CODE, DUE FROM ORDERS WHERE PAYMENT = 'UNPAID';
⑤ BEGIN
⑥     OPEN P_CUR;
⑦     LOOP
⑧         FETCH P_CUR INTO P_CODE, P_DUE;
⑨         EXIT WHEN P_CUR%NOTFOUND;
⑩         INSERT INTO NONPAY VALUES(P_CODE, P_DUE);
⑪         IF P_DUE >= 5000 THEN
⑫             DBMS_OUTPUT.PUT_LINE(P_CODE || ' IS UNPAID FOR ' || P_DUE);
⑬         END IF;
⑭     END LOOP;
⑮     CLOSE P_CUR;
⑯ END PAY;
```

- ① 프로시저 PAY를 생성합니다. 동일한 이름의 프로시저가 있는 경우 대체합니다.
- ② <ORDERS> 테이블의 'CODE'와 동일한 자료형의 변수 P_CODE를 선언합니다.
- ③ <ORDERS> 테이블의 'DUE'와 동일한 자료형의 변수 P_DUE를 선언합니다.
- ④ <ORDERS> 테이블로부터 'PAYMENT'의 값이 "UNPAID"인 튜플들의 'CODE'와 'DUE'를 메모리에 저장한 후, 그 시작 위치를 가리키는 커서 P_CUR을 선언합니다.
- ⑤ 프로시저 BODY의 시작입니다. ⑥부터 ⑮까지가 하나의 블록이 됩니다.
- ⑥ 커서 P_CUR을 엽니다.
- ⑦ LOOP문의 시작입니다. ⑦~⑭번 문장을 반복하여 수행합니다.

- ⑧ P_CUR로부터 데이터를 가져와 P_CODE와 P_DUE에 저장합니다.
- ⑨ P_CUR로부터 가져올 데이터가 없으면 LOOP문을 빠져나가 ⑭번으로 이동합니다.
- ⑩ <NONPAY> 테이블에 P_CODE와 P_DUE에 저장된 값을 삽입합니다.
- ⑪ P_DUE가 5,000 이상이면 ⑫번을 수행하고, 아니면 ⑬번으로 이동합니다.
- ⑫ P_CODE의 값을 출력하고, “ IS UNPAID FOR ”를 출력한 뒤, P_DUE의 값을 출력합니다.
(예 : CODE101 IS UNPAID FOR 6000)
- ⑬ IF문의 끝입니다.
- ⑭ LOOP문의 끝입니다.
- ⑮ 커서 P_CUR을 닫습니다.
- ⑯ 프로시저 BODY를 종료합니다.

[문제 11]

40, 45

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마 (,) 없이 40 45로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[디버깅]

es	os	i	i % 2	출력
0	0	4	Yes	40, 45
4	5	5	No	
10	12	6	Yes	
18	21	7	No	
28	32	8	Yes	
40	45	9	No	
		10	Yes	
		11	No	
		12	Yes	
		13	No	
		14		

[문제 12]

gnTir

[해설]

```

① strA = 'Information Technology'
② strL = list()
③ for i in range(0, len(strA), 2):
④     strL.append(strA[i])
⑤ for j in range(len(strL)-1, 0, -2):
⑥     print(strL[j], end='')

```

- ① 문자열 변수 strA를 선언하고 초기값으로 “Information Technology”를 저장합니다.
- ② 변수 strL을 비어있는 리스트로 선언합니다.
- ③ 반복 변수 i에 0부터 strA의 길이(22)-1까지 2씩 증가시켜 순차적으로 저장하며 ④번 문장을 반복 수행합니다.
- ④ strA에서 i번째에 있는 문자를 리스트 strL에 추가(append)합니다. 결과적으로, 문자열 변수 strA의 값을 strA[0], strA[2], strA[4], ..., strA[20]까지 차례로 한 글자씩 건너뛰며 리스트 strL에 저장합니다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	
문자열 strA	I	n	f	o	r	m	a	t	i	o	n		T	e	c	h	n	o		l	o	g	y

↓

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
리스트 strL	I	f	r	a	i	n	T	c	n	i	g

⑤ 반복 변수 *j*에 strL의 길이(11)-1부터 1까지 -2씩 증가시켜 순차적으로 저장하며 **⑥**번 문장을 반복 수행합니다.
⑥ strL의 *j*번째에 있는 문자를 줄 나눔 없이 출력합니다. 결과적으로 리스트 strL의 값을 strL[10], strL[8], strL[6], ..., strL[2]까지 하나씩 건너 뛴 글자들을 차례로 화면에 출력합니다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
리스트 strL	I	f	r	a	i	n	T	c	n	i	g
			5		4		3		2		1 ← 출력순서

※ range(초기값, 최종값, 증가값)에서 '증가값'이 음수인 경우 '초기값'에서 '최종값'+1까지 '증가값'의 절대값 만큼 감소하면서 숫자를 생성합니다. 즉, **⑥**번의 경우 10부터 1까지 -2씩 숫자를 생성하기 때문에 strL[0]은 화면에 출력되지 않습니다.

[문제 13]

DB

AC

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값을 두 줄로 나누어 쓰지 않고 한 줄로 DB AC로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[디버깅]

배열 num[]	i	i % 5	출력					
<table><tr><td>14</td><td>42</td><td>35</td><td>26</td><td>8</td></tr></table>	14	42	35	26	8	14	4	DB AC
	14	42	35	26	8			
	42	2						
	35	0						
	26	1						
8	3							

※ for (int i:num)은 향상된 반복문입니다. num 배열의 요소 수만큼 switch문을 반복 수행합니다.

- int i : num 배열의 각 요소가 일시적으로 저장될 변수를 선언합니다. num 배열과 형이 같아야 합니다. num 배열이 정수면 정수, 문자면 문자여야 합니다.
- num : 배열의 이름을 입력합니다. num 배열이 5개의 요소를 가지므로 각 요소를 i에 저장하면서 switch문을 5번 수행합니다.

[문제 14]

1, 16

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마 (,) 없이 1 16으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[디버깅]

함수의 인수로 사용된 배열명 'a'는 a[0]의 주소를 가리키는 포인터이므로 return이 없어도 메모리에 저장된 값이 수정되어 변경된 값이 main() 함수나 funcA 함수에서도 그대로 적용되게 됩니다.

main() 함수			initA() 함수	funcA 함수			출력
x	y	a[]	i	x	sum	i	
1	1	1	0	5	0	0	1, 16
	16	1 2	1		1	1	
		1 2 3	2		3	2	
		1 2 3 4	3		6	3	
		1 2 3 4 5	4		10	4	
			5		15	5	

[문제 15]

- ① Scanner
- ② break

[해설]

문제의 코드는 소수인지 판별하기 위해 제곱근까지의 숫자로 나누어떨어지는지 검사합니다. 제곱근까지의 수 중 한 개의 수에 대해서라도 나누어떨어지면 소수가 아닙니다. 예를 들어 25는 2, 3, 4, 5로 나누었을 때 5로 나누어떨어지므로 소수가 아니고, 41은 2, 3, 4, 5, 6으로 나누어도 한 번도 나누어떨어지지 않으므로 소수입니다.

import java.lang.Math;	sqrt() 메소드가 정의되어 있는 헤더 파일이다.
import java.util.Scanner;	Scanner 클래스가 정의되어 있는 헤더 파일이다.
public class Test {	
public static void main(String[] args) {	
Scanner scan = new Scanner(System.in);	Scanner 클래스의 객체 변수 scan을 키보드로 입력받을 수 있도록 생성한다.
int p = 2, n = 3, m;	정수형 변수 p, n, m을 선언하고, p와 n을 각각 2와 3으로 초기화한다. p는 소수를, n은 입력값까지의 정수를, m은 제곱근 값을 저장할 변수이다.
int max = scan.nextInt();	정수형 변수 max를 선언하고, 키보드로부터 정수형 값을 입력받아 max에 저장한다.
① while (true) {	조건을 만족하는 동안 반복하는 것인데, 조건이 true, 즉 참이므로 무한 반복한다. 결국 ⑨번의 조건을 만족하여 break를 만나기 전까지 ②~⑩번 사이의 문장을 반복하여 수행한다.
② m = (int)Math.sqrt(n);	Math 클래스의 sqrt() 메소드를 사용하여 n의 제곱근을 구한 후 정수로 변환하여 m에 저장한다.
③ for (int i = 2; i <= m; i++) {	소수 판별을 위한 for 반복문의 시작이다. 반복 변수 i가 2에서 시작하여 1씩 증가하면서 m보다 작거나 같은 동안 ④~⑦번을 반복하여 수행한다.
④ if (n % i == 0)	n을 i로 나눈 나머지가 0이면 소수가 아니므로 ⑤번을 수행하여 for문을 빠져나가고, 아니면 ⑥번으로 이동한다.
⑤ break;	for문을 벗어나 ③번으로 이동한다.
⑥ if (i == m)	i와 m의 값이 같으면 n은 소수이므로 p에 n을 저장한다.
⑦ p = n;	
}	
⑧ n++;	n의 값을 1씩 누적시킨다.
⑨ if (n > max)	n의 값이 max보다 크면 ⑩번을 수행한다.
⑩ break;	while문을 벗어나 ⑪번으로 이동한다.
}	

<pre> ❶ System.out.printf("%d\n", p); scan.close(); } } </pre>	<p>p의 값을 출력한다.</p> <p>Scanner 클래스의 객체 변수 scan을 닫는다. 프로그램 종료 전에 사용하던 메모리 영역을 해제해야 다른 프로그램이 해당 영역을 사용할 수 있다.</p>
---	--

[디버깅]

변수 max에 10이 입력되었다고 가정하고 디버깅한 결과입니다.

p	n	m	max	i	n % i	출력
2	3	1	10	2	0	7
5	4	2		2	1	
7	5	2		2	0	
	6	2		3	1	
	7	2		2	0	
	8	2		2	1	
	9	3		3	0	
	10	3		2	0	
	11			2		
				3		
				2		

[문제 16]

다음 중 하나를 쓰면 됩니다.

시큐어 코딩, Secure Coding

※ 답안 작성 시 주의 사항

한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 17]

각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① UDP, User Datagram Protocol
- ② RTCP, Real-Time Control Protocol

※ 답안 작성 시 주의 사항

한글 또는 영문을 Full-name이나 약어로 쓰라는 지시사항이 없을 경우 한글이나 영문 약어로 쓰는 것이 유리합니다. 영문을 Full-name으로 풀어쓰다가 스펠링을 틀리면 오답으로 처리되니까요.

[문제 18]

- ① commit
- ② push



1회 실전 모의고사 정답 및 해설

[문제 1]

2 3 5 7

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 2, 3, 5, 7로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

문제의 코드는 7 이하의 소수들을 출력하는 프로그램입니다. 숫자를 2부터 7까지 증가시키면서 각각의 숫자가 소수인지를 판별하기 위해 2부터 1씩 증가시키면서 나누어 떨어지는지 검사합니다. 예를 들어 5는 2, 3, 4, 5로 나누었을 때 나머지가 0이 되는 순간이 자기수(5)이므로 소수이고, 6은 2, 3, 4, 5, 6으로 나누었을 때 나머지가 0이 되는 순간이 자기수(6)가 아닌 2이므로 소수가 아닙니다.

```
#include <stdio.h>
main() {
    int k = 2, j;
    ① while(1) {
    ②         j = 2;
    ③         while (k % j != 0)
    ④             j++;
    ⑤         if (k == j)
    ⑥             printf("%d ", k);
    ⑦         if (k < 7)
    ⑧             k++;
    ⑨         else
            break;
    }
}
```

- ① 조건이 참(1)이므로 ⑨번의 break를 만날 때까지 while문 내의 문장을 무한 반복한다.
- ② j의 값을 2로 치환한다.
- ③ k를 j로 나눈 나머지가 0이 아니면 ④번 문장을 실행하고, 0이면 ⑤번으로 이동한다.
- ④ 'j = j + 1;'과 동일하다. j에 1씩 누적한다.
- ⑤ k와 j의 값이 같으면 ⑥번 문장을 실행하고, 아니면 ⑦번으로 이동한다.
- ⑥ k의 값을 출력하고 이어서 공백을 한 칸 출력한다.
- ⑦ k의 값이 7보다 작으면 ⑧번 문장을 실행하고, 아니면 ⑨번 문장을 실행한다.
- ⑧ 'k = k + 1;'과 동일하다. k에 1씩 누적한다.
- ⑨ while문을 벗어나 프로그램을 종료한다.



반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

k	j	k % j	출력
2	2	0	2357
3	2	1	
4	3	0	
5	2	0	
6	2	1	
7	3	2	
	4	1	
	5	0	
	2	0	
	2	1	
	3	1	
	4	3	
	5	2	
	6	1	
	7	0	

[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 2]

시퀀스 다이어그램(Sequence Diagram)

[문제 3]

이상(Anomaly)

[문제 4]

뷰(View)

[문제 5]

① 범위 분할(Range Partitioning) ② 해시 분할(Hash Partitioning) ③ 조합 분할(Composite Partitioning)

[문제 6]

12

해설

```
public class Test {
    public static void main(String[ ] args) {
        int i = 0, hap = 0;
        ① String a = "Technology";
        ② do {
            ③     if(a.charAt(i) == 'o')
            ④         hap += i;
            ⑤     i++;
        ⑥ } while (i < a.length( ));
        ⑦ System.out.printf("%d", hap);
    }
}
```

① 문자열 변수 a를 선언하고 "Technology"로 초기화한다.

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
a	T	e	c	h	n	o	l	o	g	y

② do~while 반복문의 시작점이다.

③ a에서 i번째에 있는 문자가 'o'이면 ④번 문장을 실행하고, 아니면 ⑤번으로 이동한다.

• charAt() : 해당 문자열에서 인수에 해당하는 위치의 문자를 반환하는 메소드

④ hap에 i의 값을 누적한다.

⑤ 'i = i + 1;'과 동일하다. i에 1씩 누적한다.

⑥ i가 문자열 변수 a의 크기인 10보다 작은 동안 ③~⑤번을 반복 수행한다.

• length() : 변수의 크기를 반환하는 메소드

⑦ hap의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

a	i	a.charAt(i)	hap	a.length()	출력
Technology	0	T	0	10	12
	1	e	5		
	2	c	12		
	3	h			
	4	n			
	5	o			
	6	l			
	7	o			
	8	g			
	9	y			
	10				



[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 7]

미들웨어(Middleware)

[문제 8]

WAS(Web Application Server, 웹 애플리케이션 서버)

[문제 9]

2, 2

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마(,) 없이 2 2로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
public class Test {
    ① public static int a = 5;
    ② public static void main(String[] args) {
        ③ int b = 2;
        ④ cal(b);
        cal(b);
        System.out.printf("%d, %d\n", a, b);
    }
}
```

- ① 정수형 전역 변수 a를 선언하고, 초기값으로 5를 할당한다. a는 main() 메소드 밖에서 선언했기 때문에 이 클래스에 속한 모든 메소드에서 사용할 수 있다.
- ② main() 메소드의 시작이다.
- ③ 정수형 변수 b를 선언하고 2로 초기화한다.
- ④ b의 값 2를 인수로 하여 cal() 메소드를 호출한다.

```
5 static void cal(int b) {
6     if (b < a)
7         a -= 3;
    else
        b += 3;
}
```

- ⑤ cal() 메소드의 시작점이다. cal() 메소드가 호출될 때 2를 전달받았으므로 b는 2이다.
- ⑥ b의 값 2가 a의 값 5보다 작으므로 ⑦번 문장을 실행한다.
- ⑦ 'a = a - 3'과 같다. a의 값에서 3을 뺀 값을 a에 저장하고, 메소드가 종료되었으므로 main() 메소드로 돌아간다. a는 전역 변수이므로 값의 변화가 클래스에 속한 모든 메소드에서 유지된다.



```
public static void main(String[ ] args) {
    int b = 2;
    cal(b);
    ⑧ cal(b);
    System.out.printf("%d, %d\n", a, b);
}
```

⑧ b의 값 2를 인수로 하여 cal() 메소드를 호출한다.

```
⑨ static void cal(int b) {
    ⑩ if (b < a)
        a -= 3;
    else
    ⑪ b += 3;
}
```

⑨ cal() 메소드가 호출될 때 2를 전달받았으므로 b는 2이다.

⑩ b의 값 2가 a의 값 2보다 작지 않으므로 ⑪번 문장을 실행한다.

⑪ 'b = b + 3;'과 같다. b에 3을 누적하고, 메소드가 종료되었으므로 main() 메소드로 돌아간다. b는 메소드 안에서 선언된 지역 변수이므로 cal() 메소드를 벗어나면 소멸한다.

```
public static void main(String[ ] args) {
    int b = 2;
    cal(b);
    cal(b);
    ⑫ System.out.printf("%d, %d\n", a, b);
}
```

⑫ a와 b의 값을 출력한다.

결과 2 2

사용자 정의 함수(메소드)에 따른 변수들의 값의 변화는 다음과 같다.

a	main() 메소드	cal() 메소드	출력
	b	b	
5	2	2	2 2
2		5	

※ main() 메소드에서 선언한 b와 cal() 메소드에서 선언한 b는 해당 영역 내에서만 유효합니다. 그러므로 cal() 메소드에서 b의 값이 변경되어도 이 변경된 값이 main() 메소드의 b 변수에 전달되지 않으므로 main() 메소드에서 b 값을 출력할 때는 main() 메소드에서 선언한 b의 초기값인 2가 그대로 출력됩니다.

[문제 10]

① 명확성(Clarify) ② 완전성(Completeness) ③ 일관성(Consistency)

[문제 11]

==

※ 답안 작성 시 주의 사항

C언어에서 사용하는 형식에 맞게 정확히 작성해야 합니다. '같다'는 표현을 '='으로 하지 않도록 주의하세요.



해설

```
#include <stdio.h>
main() {
    int i, a[5], cnt = 0;
    ❶ for (i = 0; i < 5; i++)
    ❷     scanf("%d", &a[i]);
    ❸ for (i = 0; i < 5; i++)
    ❹     if (a[i] % 2 == 0)
    ❺         cnt = cnt + 1;
    ❻ printf("짝수의 개수 : %d개", cnt);
}
```

- ❶ 반복 변수 i 가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ❷번을 반복하여 수행한다.
- ❷ $a[i]$ 에 입력받은 값을 저장한다.
- ❸ 반복 변수 i 가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ❹, ❺번을 반복하여 수행한다.
- ❹ $a[i]$ 를 2로 나눈 나머지가 0이면 ❺번 문장을 실행한다.
- ❺ cnt 에 1씩 누적한다.
- ❻ 화면에 짝수의 개수 : 를 출력하고, 이어서 cnt 의 값을 출력한 다음, 개를 출력한다.

❶번 반복문에서 배열 a 에 차례대로 1, 2, 3, 4, 5가 입력되었다고 가정하고, ❷번 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	a[i]	a[i]%2	cnt	배열 a
0	1	1	0	<div><div><div>a[i]</div><div>a[0]a[1]a[2]a[3]a[4]</div><div>12345</div></div><div>짝수의 개수 : 2개</div></div>
1	2	0	1	
2	3	1	2	
3	4	0		
4	5	1		
5				

[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 12]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

제어의 역흐름은 개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘겨 생산성을 향상시키는 것이다.

[문제 13]

직관성, 유효성, 학습성, 유연성

[문제 14]

동치 분할 검사(Equivalence Partitioning Testing)

[문제 15]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

오류-부재의 궤변은 소프트웨어의 결함을 모두 제거해도 사용자의 요구사항을 만족시키지 못하면 해당 소프트웨어는 품질이 높다고 말할 수 없는 것을 의미한다.



[문제 16]

UPDATE 상품목록 SET 제조가=100000, 생산지='Australia' WHERE 상품명='DR-725F';

해설

UPDATE 상품목록

SET 제조가=100000, 생산지='Australia'

WHERE 상품명='DR-725F';

〈상품목록〉 테이블을 갱신한다.

'제조사'를 100,000으로 '생산지'를 "Australia"로 갱신한다.

'상품명'이 "DR-725F"인 튜플만을 대상으로 한다.

[문제 17]

인증(Authentication)

[문제 18]

ARIA

[문제 19]

GUI(Graphic User Interface, 그래픽 사용자 인터페이스)

[문제 20]

① 최적 적합(Best Fit) ② 최악 적합(Worst Fit)





2회 실전 모의고사 정답 및 해설

[답안 작성 방법 안내]

'운영체제(OS: Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 1]

요구공학(Requirement Engineering)

[문제 2]

BCNF(Boyce-Codd Normal Form)

해설

〈동아리〉 릴레이션에는 다음과 같은 함수적 종속이 존재합니다.

- (회원번호, 소속) → 강사명
- 강사명 → 소속

〈동아리〉 릴레이션에는 결정자이지만 후보키가 아닌 속성이 존재합니다. 즉 함수적 종속 '강사명 → 소속'에서 강사명은 결정자이지만 〈동아리〉 릴레이션에서 후보키가 아닙니다. 왜냐하면 강사명 '김범수'는 항상 소속이 '축구'이고, 강사명 '나경락'은 항상 소속이 '테니스'로 강사명이 소속을 종속하지만 강사명 속성에는 중복된 값이 있어 강사명 속성만으로는 레코드를 유일하게 식별할 수 없기 때문입니다. 그러므로 〈동아리〉 릴레이션은 BCNF가 아닙니다. BCNF는 릴레이션의 결정자가 모두 후보키가 되는 정규형입니다.

- 문제에 제시된 내용은 〈동아리〉 릴레이션에서 BCNF를 만족하지 못하게 하는 속성인 결정자이면서 후보키가 아닌 속성(즉 강사명 → 소속)을 분리하여 〈회원〉 릴레이션과 〈강사〉 릴레이션으로 무손실 분해하는 정규화 과정입니다.

[문제 3]

트리거(Trigger)

[문제 4]

12

해설

문제의 코드는 피보나치 수를 누적하여 출력하는 프로그램입니다. 피보나치 수는 0과 1로 시작하며, 다음 피보나치 수는 바로 앞의 두 피보나치 수의 합이 되어 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... 과 같이 나열되는 수열입니다. 코드는 세 번째 피보나치 수까지 누적한 값을 초기값으로 하고, 반복문을 통해 피보나치 수를 3회 더 진행하여 추가로 누적했으므로, 총 6번째 피보나치 수(5)까지의 값들의 합이 결과로 나오게 됩니다.

```

#include <stdio.h>
main() {
    int a, b, c, sum;
    ① a = b = 1;
    ② sum = a + b;
    ③ for (int i = 3; i <= 5; i++) {
    ④     c = a + b;
    ⑤     sum += c;
    ⑥     a = b;
    ⑦     b = c;
    }
}

```



```
⑧ printf("%d", sum);
}
```

- ① b에 1을 저장하고, a에 b의 값을 저장한다. 즉 a와 b에 1을 저장한다.
- ② a와 b의 합을 sum에 저장한다.
- ③ 반복 변수 i가 3에서 시작하여 1씩 증가하면서 5보다 작거나 같은 동안 ④~⑦번을 반복 수행한다.
- ④ a와 b의 합을 c에 저장한다.
- ⑤ 'sum=sum+c;'과 동일하다. sum에 c의 값을 누적한다.
- ⑥ a의 값을 b의 값으로 치환한다.
- ⑦ b의 값을 c의 값으로 치환한다.
- ⑧ sum의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

a	b	c	sum	i	출력
1	1	2	2	3	12
1	2	3	4	4	
2	3	5	7	5	
3	5		12	6	

[문제 5]

00001011

※ 답안 작성 시 주의 사항

Java에서는 배열 선언 시 모든 요소가 0으로 초기화됩니다. 그러므로 a 배열 출력 시 값이 저장되지 않은 요소에는 0이 출력됩니다.

해설

문제의 코드는 10진수 11을 2진수로 변환하여 배열에 저장한 후 출력하는 프로그램입니다. 10진수를 2진수로 변환하려면 10진수를 2로 나누어 나머지를 구한 후 저장하고, 다시 몫을 2로 나누어 나머지를 구해 저장하는 과정을 반복합니다. 몫이 0이 될 때까지 이 작업을 반복한 후 마지막에 구한 나머지부터 거꾸로 출력합니다.

```
public class Test {
    public static void main(String[] args) {
        ① int a[] = new int[8];
        ② int d = 11, n = 0;
        ③ do {
        ④     a[n++] = d % 2;
        ⑤     d /= 2;
        ⑥ } while (d > 0);
        ⑦ for (n = 7; n >= 0; n--)
        ⑧     System.out.printf("%d", a[n]);
        }
    }
}
```

- ① 8개의 요소를 갖는 정수형 배열 a를 선언한다.
- ② 정수형 변수 d와 n을 선언하고, 각각 11과 0으로 초기화한다.
- ③ do~while 반복문의 시작점이다.
- ④ n++은 후지 연산이므로 a[n]에 d를 2로 나눈 나머지를 저장한 후 n의 값을 1 증가시킨다.



- ⑤ d에 d를 2로 나눈 값을 저장한다.
 ※ C언어에서 정수 나눗셈은 결과도 정수이다. 예) 정수형 변수 a가 10인 경우 : $a/3=3$, $a/7=1$
 ⑥ d가 0보다 큰 동안 ④, ⑤번을 반복 수행한다.
 ⑦ 반복 변수 n이 7에서 시작하여 1씩 감소하면서 0보다 크거나 같은 동안 ⑥번을 반복하여 수행한다.
 ⑧ a[n]의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

d	n	d%2	a[n]	배열 a	출력
11	0	1	1	a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] <div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> <div>0</div> </div>	
5	1	1	1	1 1 0 1	
2	2	0	0		
1	3	1	1		
0	4				
	7		0	a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] <div>1 1 0 1 0 0 0 0</div>	00001011
	6		0		
	5		0		
	4		0		
	3		1		
	2		0		
	1		1		
	0		1		
	-1				

[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 6]

비트맵 인덱스(Bitmap Index)

[문제 7]

API(Application Programming Interface)

[문제 8]

① 명세 기반 테스트 ② 구조 기반 테스트 ③ 경험 기반 테스트

[문제 9]

코드 커버리지(Code Coverage)



[문제 10]

25

해설

```
public class Test {
    public static void main(String[] args) {
        ① int numAry[] = { 1, 2, 3, 4, 5 };
        ② init(numAry);
        ⑥ prnt(numAry);
    }
    ③ static void init(int a[] ) {
        ④ for(int i = a.length - 1; i > 0; i--) {
        ⑤     a[i] += a[i - 1];
        }
    }
    ⑦ static void prnt(int a[] ) {
        ⑧ int sum = 0;
        ⑨ for(int i : a)
        ⑩     sum += i;
        ⑪ System.out.print(sum);
    }
}
```

- ① 배열을 선언할 때 사용할 개수를 생략하고 초기값을 지정하면, 초기값으로 지정된 값의 수와 같은 크기의 배열이 선언된다.

	numAry[0]	numAry[1]	numAry[2]	numAry[3]	numAry[4]
배열 numAry	1	2	3	4	5

- ② numAry를 인수로 하여 init() 메소드를 호출한다. ③번으로 이동한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다.
 ③ init() 메소드의 시작점이다. 정수형 배열 a가 main() 메소드에서 전달한 배열 numAry의 시작 주소를 받는다.
 ④ 반복 변수 i가 a의 길이에서 1을 뺀 값인 4에서 시작하여 1씩 감소하면서 0보다 큰 동안 ⑥번을 반복 수행한다.
 • length : length는 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. 배열 a는 5개의 요소를 가지므로 a.length는 5를 가지고 있다.
 ⑤ a[i]에 a[i-1]의 값을 누적한다. 반복이 모두 종료되면 ⑥번으로 이동한다.

i	배열 a				
4					
3	a[0]	a[1]	a[2]	a[3]	a[4]
2	1	2	3	4	5
1		3	5	7	9
0					

- ⑥ numAry를 인수로 하여 prnt() 메소드를 호출한다. ⑦번으로 이동한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다.
 ⑦ prnt() 메소드의 시작점이다. 정수형 배열 a가 main() 메소드에서 전달한 배열 numAry의 시작 주소를 받는다.
 ⑧ 정수형 변수 sum을 선언하고 0으로 초기화한다.
 ⑨ 배열 a의 요소 수만큼 ⑩번을 반복 수행한다.
 • int i : 배열 a의 각 요소가 할당될 변수를 선언한다.
 • a : 배열의 이름을 적어준다. 배열이 5개의 요소를 가지므로 각 요소를 i에 할당하면서 ⑩번을 5회 수행한다.
 ⑩ sum에 i의 값을 누적한다.



sum	i	배열 a				
0	1					
1	3					
4	5	a[0]	a[1]	a[2]	a[3]	a[4]
9	7	1	3	5	7	9
16	9					
25						

❶ sum의 값을 출력하고 main() 메소드로 돌아가 프로그램을 종료한다.

[문제 11]

① REVOKE ② CASCADE

해설

〈수강〉 테이블에 대한 UPDATE 권한을 취소하면 다른 사람에게 UPDATE 권한을 부여할 수 있는 권한도 함께 취소되기 때문에 'GRANT OPTION FOR'는 생략됩니다. 자신에게 권한이 없어지면 해당 권한을 다른 사람에게 부여할 수 없습니다.

[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 12]

JDBC(Java DataBase Connectivity)

[문제 13]

① Spring ② Django

[문제 14]

와이어프레임(Wireframe)

[문제 15]

① 제출(Submit) ② 접수(Hold) ③ 대기(Wait) 또는 블록(Block)

[문제 16]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크 또는 시스템의 상태를 불능으로 만드는 공격 방법이다.

[문제 17]

인터넷 계층(Internet Layer)

[문제 18]

형상 관리(SCM; Software Configuration Management)

[문제 19]

정의(Definition) 기능, 조작(Manipulation) 기능, 제어(Control) 기능

[문제 20]

신뢰도는 시스템이 주어진 문제를 정확하게 해결하는 정도를 의미한다.



3회 실전 모의고사 정답 및 해설

[답안 작성 방법 안내]

‘운영체제(OS: Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

[문제 1]

도출(Elicitation), 분석(Analysis), 명세(Specification), 확인(Validation)

[문제 2]

① 운영 데이터(Operational Data) ② 저장된 데이터(Stored Data)

[문제 3]

데이터 웨어하우스(Data Warehouse)

[문제 4]

15

해설

```
#include <stdio.h>
main() {
  ① int x = 7, y = 10, z;
  ②⑥ z = bitCal(x, y);
  ⑦ printf("%d", z);
}

③ int bitCal(x, y) {
  ④ if (x < y)
  ⑤ return x | y;
  else
  ⑥ return x & y;
}
```

④번 조건식이 거짓일 경우 실행할 문장의 시작점이다.
x와 y의 값을 &(비트 and) 연산한 결과를 main 함수로 반환한다.

- ① 정수형 변수 x, y, z를 선언하고, x와 y의 값을 각각 7과 10으로 초기화한다.
- ② x와 y의 값 7과 10을 인수하여 bitCal 함수를 호출한 다음 돌려받은 값을 z에 저장한다.
- ③ 리턴값이 정수인 bitCal 함수의 시작점이다. ②번에서 7과 10을 전달받았으므로 x는 7, y는 10이다.
- ④ x가 y보다 작으면 ⑤번 문장을 수행하고, 아니면 else 다음 문장을 수행한다. x(7)가 y(10)보다 작으므로 ⑤번으로 이동한다.
- ⑤ x와 y의 값을 |(비트 or) 연산한 결과를 호출한 곳(main 함수)으로 반환한다.
C 언어에서 정수형 변수는 4바이트이므로 각 변수의 값을 4바이트 2진수로 변환한 다음 각 비트를 연산한다.

```
7 = 0000 0000 0000 0000 0000 0000 0000 0111
10 = 0000 0000 0000 0000 0000 0000 0000 1010
|-----|
| 0000 0000 0000 0000 0000 0000 0000 1111 = 10진수로 15이다.
```

- ⑥ bitCal() 함수로부터 반환받은 값 15를 z에 저장한다.
- ⑦ z의 값을 출력한다.



[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 5]

파티션(Patition)

[문제 6]

ESB(Enterprise Service Bus)

[문제 7]

UDDI(Universal Description, Discovery and Integration)

[문제 8]

모듈화(Modularity)

[문제 9]

20 10 5 2

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에逗를 넣어 20, 10, 5, 2로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

모든 Java 프로그램은 반드시 main() 메소드부터 시작해야 한다.

```
public static void main(String[] args) {
    ① recursive(20);
}
```

① 20을 인수로 recursive() 메소드를 호출한다.

①회

```
② public static void recursive(int a) {
    ③ if (a <= 1)
        return;
    ④ System.out.print(a + " ");
    ⑤ recursive(a / 2);
}
```

② recursive() 메소드가 호출될 때 20을 전달받았으므로 a는 20이다.

③ a가 1보다 작거나 같지 않으므로 ④번으로 이동한다.

④ a의 값 20을 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20



5 a를 2로 나눈 값 10을 인수로 recursive() 메소드를 호출한다.

②회

```
6 public static void recursive(int a) {
7     if (a <= 1)
8         return;
9     System.out.print(a + " ");
10    recursive(a / 2);
11 }
```

6 recursive() 메소드가 호출될 때 10을 전달받았으므로 a는 10이다.

7 a가 1보다 작거나 같지 않으므로 8번으로 이동한다.

8 a의 값 10을 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20 10

9 a를 2로 나눈 값 5를 인수로 recursive() 메소드를 호출한다.

③회

```
10 public static void recursive(int a) {
11     if (a <= 1)
12         return;
13     System.out.print(a + " ");
14     recursive(a / 2);
15 }
```

10 recursive() 메소드가 호출될 때 5를 전달받았으므로 a는 5이다.

11 a가 1보다 작거나 같지 않으므로 12번으로 이동한다.

12 a의 값 5를 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20 10 5

13 a를 2로 나눈 값 2를 인수로 recursive() 메소드를 호출한다.

※ C, Java에서 정수 나눗셈은 결과도 정수이다. ㉠ 정수형 변수 a가 11인 경우 : $a / 2 = 5$, $a / 7 = 1$

④회

```
14 public static void recursive(int a) {
15     if (a <= 1)
16         return;
17     System.out.print(a + " ");
18     recursive(a / 2);
19 }
```

14 recursive() 메소드가 호출될 때 2를 전달받았으므로 a는 2이다.

15 a가 1보다 작거나 같지 않으므로 16번으로 이동한다.

16 a의 값 2를 출력하고, 이어서 공백 한 칸을 출력한다.

결과 20 10 5 2

17 a를 2로 나눈 값 1을 인수로 recursive() 메소드를 호출한다.

⑤회

```
18 public static void recursive(int a) {
19     if (a <= 1)
20         return;
21     System.out.print(a + " ");
22     recursive(a / 2);
23 }
```



18 recursive() 메소드가 호출될 때 1을 전달받았으므로 a는 1이다.

19 a가 1보다 작거나 같으므로 20번으로 이동한다.

20 메소드를 종료하고 반환값 없이 제어를 4회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

4회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    21 recursive(a / 2);
}
```

21 메소드를 종료하고 반환값 없이 제어를 3회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

3회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    22 recursive(a / 2);
}
```

22 메소드를 종료하고 반환값 없이 제어를 2회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

2회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    23 recursive(a / 2);
}
```

23 메소드를 종료하고 반환값 없이 제어를 1회 recursive(a / 2) 메소드를 호출했던 곳으로 옮긴다.

1회

```
public static void recursive(int a) {
    if (a <= 1)
        return;
    System.out.print(a + " ");
    24 recursive(a / 2);
}
```

24 메소드를 종료하고 반환값 없이 제어를 처음 recursive(20) 메소드를 호출했던 main() 메소드로 옮긴다.

```
public static void main(String[] args) {
    25 recursive(20);
}
```

25 이후 수행할 코드가 없으므로 프로그램을 종료한다.



[답안 작성 방법 안내]

'운영체제(OS: Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 10]

처리 능력(Throughput) 향상, 사용 가능도(Availability) 향상, 신뢰도(Reliability) 향상, 반환 시간(Turn Around Time) 단축

[문제 11]

① 페이퍼 프로토타입(Paper Prototype) ② 디지털 프로토타입(Digital Prototype)

[문제 12]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

시스템에 과도한 정보량이나 빈도 등을 부과하여 과부하 시에도 소프트웨어가 정상적으로 실행되는지를 확인하는 테스트이다.

[문제 13]

크로스사이트 스크립팅(XSS)

[문제 14]

① 알파 테스트 ② 베타 테스트

[문제 15]

해시(Hash)

[문제 16]

fsck

[문제 17]

① 상호 배제(Mutual Exclusion) ② 점유 및 대기(Hold and Wait) ③ 환형 대기(Circular Wait)

[문제 18]

11 5 2 1

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 코마를 넣어 11, 5, 2, 1로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
#include <stdio.h>
void func(int, int);
main(){
    ① int p, q;
    ② p = 10;
    ③ q = p++ > 10 ? p + 3 : p - 3;
    ④ func(p, q);
}
⑤ void func(int x, int y){
    ⑥ if (x > y)
    ⑦ while (x > 0) {
```



```

8      printf("%d ", x);
9      x = x / 2;
      }
10     else
11         while (y > 0) {
12             printf("%d ", y);
13             y = y / 2;
        }
    }

```

- ① 정수형 변수 p와 q를 선언한다.
- ② p에 10을 저장한다.
- ③ p++가 10보다 크면 p에 3을 더한 값을 저장하고, 아니면 q에 3을 뺀 값을 저장한다. p가 후치 연산이므로 조건식 'p(10) > 10'을 판별한 후 p의 값이 1 증가하여, 거짓인 경우의 연산 'p(11) - 3'을 수행한다.
- ④ p와 q의 값 1과 8을 인수로 func 함수를 호출한다.
- ⑤ 리턴값이 없는 func 함수의 시작점이다. ⑤번에서 11과 8을 전달받았으므로 x는 11, y는 8이다.
- ⑥ x가 y보다 크면 ⑦~⑨번 문장을 수행하고, 아니면 ⑪~⑬번 다음 문장을 수행한다. x(11)가 y(8)보다 크므로 ⑦번으로 이동한다.
- ⑦ x가 0보다 큰 동안 ⑧, ⑨번 문장을 반복 수행한다.
- ⑧ x의 값을 출력하고, 이어서 공백 한 칸을 출력한다.
- ⑨ x에 x를 2로 나눈 값을 저장한다.
※ C, Java에서 정수 나눗셈은 결과도 정수이다. 예 정수형 변수 a가 11인 경우 : $a/2 = 5$, $a/7 = 1$
- ⑩ ⑥번 조건식이 거짓일 경우 실행할 문장의 시작점이다.
- ⑪ y가 0보다 큰 동안 종괄호({ }) 안의 문장을 반복 수행한다.
- ⑫ y의 값을 출력하고, 이어서 공백 한 칸을 출력한다.
- ⑬ y에 y를 2로 나눈 값을 저장한다.

[문제 19]

- ① 페이징(Paging) 기법 ② 세그먼테이션(Segmentation) 기법

[문제 20]

- ① PRIMARY KEY ② UNIQUE ③ REFERENCES

※ 답안 작성 시 주의 사항

대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지 마세요.

해설

```

CREATE TABLE 참가자
(참가번호 INT PRIMARY KEY,
이름 CHAR(10) UNIQUE,
국가번호 INT,
FOREIGN KEY(국가번호) REFERENCES 국가목록(부여번호)

ON UPDATE CASCADE);

```

〈참가자〉 테이블을 생성한다.
참가번호 속성은 숫자이고, 기본키이다.
이름 속성은 문자 10자리이고, 중복된 값을 가질 수 없다.
국가번호 속성은 숫자이다.
국가번호 속성은 〈국가목록〉 테이블의 부여번호 속성을 참조하는 외래키이다.
〈국가목록〉 테이블에서 부여번호 속성이 변경되면 관련된 모든 튜플의 부여번호 속성의 값도 같은 값으로 변경한다.



4회 실전 모의고사 정답 및 해설

[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 1]

EAI(Enterprise Application Integration)

[문제 2]

43, 2

29, 4

54, 1

33, 3

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값을 한 줄로 43, 2, 29, 4, ... 혹은 43 2 29 4 ...로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

문제의 코드는 배열의 각 요소의 값들을 자기를 포함한 다른 요소들의 값과 비교하여 석차를 구한 후 요소의 값과 석차를 출력하는 프로그램입니다.

```

#include <stdio.h>
main() {
    ① int std[] = { 43, 29, 54, 33 };
    ② int r;
    ③ for (int i = 0; i < 4; i++) {
    ④     r = 1;
    ⑤     for (int j = 0; j < 4; j++)
    ⑥         if (std[i] < std[j])
    ⑦             r++;
    ⑧     printf("%d, %d\n", std[i], r);
    }
}

```

- ① 4개의 요소를 갖는 정수형 배열 std를 선언하고 초기화한다.
 - ② 정수형 변수 r을 선언한다.
 - ③ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 4보다 작은 동안 ④~⑧번을 반복하여 수행한다.
 - ④ r의 값을 1로 치환한다.
 - ⑤ 반복 변수 j가 0에서 시작하여 1씩 증가하면서 4보다 작은 동안 ⑥, ⑦번을 반복하여 수행한다.
 - ⑥ std[i]의 값이 std[j]의 값보다 작으면 ⑦번 문장을 실행하고, 아니면 반복문의 시작인 ⑤번으로 돌아간다.
 - ⑦ 'r = r + 1'과 동일하다. r에 1을 누적한다.
 - ⑧ std[i]의 값을 출력하고, 실패()와 공백 한 칸을 출력한 다음 r의 값을 출력하고 커서를 다음 줄로 이동한다.
- 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

	std[0]	std[1]	std[2]	std[3]
배열 std	43	29	54	33



i	j	std[i]	std[j]	r	출력
0	0	43	43	1	43, 2
	1		29	2	29, 4
	2		54		54, 1
	3		33		33, 3
	4				
1	0	29	43	1	
	1		29	2	
	2		54	3	
	3		33	4	
	4				
2	0	54	43	1	
	1		29		
	2		54		
	3		33		
	4				
3	0	33	43	1	
	1		29	2	
	2		54	3	
	3		33		
	4				
4					

[답안 작성 방법 안내]

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

[문제 3]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

클러스터는 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법이다.

[문제 4]

- ① 상품조회 ② 로그인 ③ <<extends>>

해설

- ③ 특정 조건에 부합되어 유스케이스의 기능이 확장될 때 원래의 유스케이스와 확장된 유스케이스와의 관계를 확장(Extends) 관계라고 하며, 확장 관계는 확장될 유스케이스에서 원래의 유스케이스 쪽으로 점선 화살표를 연결한 후 화살표 위에 <<extends>>라고 표기합니다.

[문제 5]

물리 계층(Physical Layer), 데이터 링크 계층(Data Link Layer), 네트워크 계층(Network Layer), 전송 계층(Transport Layer), 세션 계층(Session Layer), 표현 계층(Presentation Layer), 응용 계층(Application Layer)

[문제 6]

- ① 속성(Attribute) ② 관계(Relationship) ③ 개체(Entity)

[문제 7]

목업(Mockup)

[문제 8]

백도어(Back Door, Trap Door)

[문제 9]

21

해설

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        ① Scanner var = new Scanner(System.in);
        ② int n, i, sum = 0;
        ③ n = var.nextInt();
        ④ var.close();
        ⑤ for (i = 1; i <= n; i++)
        ⑥     sum += i;
        ⑦ System.out.printf("%d", sum);
    }
}
```

- ① Scanner 클래스의 객체 변수 var를 키보드로 입력받을 수 있도록 생성한다. System.in은 표준 입력장치, 즉 키보드를 의미한다.
- ② 정수형 변수 n, i, sum을 선언하고, sum을 0으로 초기화한다.
- ③ 키보드로부터 정수형 값을 입력받아 n에 저장한다.
- ④ Scanner 클래스의 객체 변수는 임의의 메모리 영역을 확보하여 사용하는 것이므로 프로그램 종료 전에 close() 메소드를 이용하여 사용하던 메모리 영역을 해제해야 다른 프로그램이 해당 영역을 사용할 수 있다.
- ⑤ 반복 변수 i가 1에서 시작하여 1씩 증가하면서 n보다 작거나 같은 동안 ⑥번을 반복 수행한다.
- ⑥ sum에 i의 값을 누적한다.
- ⑦ sum의 값을 출력한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

n	i	sum	출력
6	1	0	21
	2	1	
	3	3	
	4	6	
	5	10	
	6	15	
	7	21	

[문제 10]

- ① 중복 투명성(Replication Transparency) ② 장애 투명성(Failure Transparency) ③ 병행 투명성(Concurrency Transparency)

[문제 11]

- ① 하향식 ② 상향식

[문제 12]

기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)

[문제 13]

- ① DTO/VO ② DAO ③ Controller



[문제 14]

2, 0

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마 없이 2 0으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

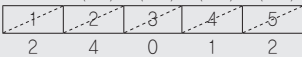
해설

모든 C프로그램은 반드시 main() 함수부터 시작해야 한다.

```
#include <stdio.h>
③ void res(int a[]) {
④     int i = 0;
⑤     while (i < 5) {
⑥         if (a[i] < 3)
⑦             a[i] *= 2;
           else
⑧             a[i] %= 3;
⑨         i++;
    }
}
main() {
①     int a[] = { 1, 2, 3, 4, 5 };
②     res(a);
⑩    printf("%d, %d", *a, *(a + 2));
}
```

- ① 5개의 요소를 갖는 정수형 배열 a를 선언하고 초기화한다.
- ② a를 인수로 하여 res 함수를 호출한다. 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다. 즉 res(a)는 res(&a[0])과 같은 의미이다.
- ③ 리턴값이 없는 res 함수의 시작점이다. ②번에서 보낸 배열 a의 시작 주소를 배열 a가 받는다.
- ④ 정수형 변수 i를 선언하고 0으로 초기화한다.
- ⑤ i가 5보다 작은 동안 ⑥~⑨번 문장을 반복 수행한다.
- ⑥ a[i]가 3보다 작으면 ⑦번 문장을 실행하고, 아니면 ⑧번 문장을 실행한다.
- ⑦ 'a[i] = a[i] * 2'와 동일하다. a[i]의 값에 2를 곱한 값을 a[i]에 저장한다.
- ⑧ 'a[i] = a[i] % 3'과 동일하다. a[i]의 값을 3으로 나눈 나머지를 a[i]에 저장한다.
- ⑨ 'i = i + 1'과 동일하다. i에 1씩 누적한다.
- ⑩ *a와 *(a+2)의 값을 출력한다. *a는 a[0]과 같은 의미이고 *(a+2)는 a[0]에서 2번지가 증가한 a[2]와 같은 의미이다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	배열 a	출력
0		
1	a[0] a[1] a[2] a[3] a[4]	
2	*a *(a+1) *(a+2) *(a+3) *(a+4)	
3		2, 0
4	2 4 0 1 2	
5		



[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 15]

테스트 오라클(Test Oracle)

[문제 16]

① 커널(Kernel) ② 셸(Shell)

[문제 17]

스래싱은 프로세스의 처리 시간보다 페이지 교체에 소요되는 시간이 더 많아지는 현상이다.

[문제 18]

분산 저장소 방식

[문제 19]

① FIFO(First In First Out) ② NUR(Not Used Recently) ③ LRU(Least Recently Used)

[문제 20]

① GROUP BY ② HAVING

※ 답안 작성 시 주의 사항

대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지 마세요.

해설

SELECT 소속지점, AVG(성과점수)
FROM 사원
GROUP BY 소속지점
HAVING AVG(성과점수) > 30;

'소속지점', '성과점수'의 평균을 표시한다.
<사원> 테이블을 대상으로 검색한다.
'소속지점'을 기준으로 그룹을 지정한다.
'성과점수'의 평균이 30 초과인 그룹만을 표시한다.



5회 실전 모의고사 정답 및 해설

[답안 작성 방법 안내]

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

[문제 1]

정형 분석(Formal Analysis)

[문제 2]

OLTP(Online Transaction Processing)

[문제 3]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

개체 무결성은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정이다.

[문제 4]

① 클러스터드(Clustered) ② 언클러스터드(Non-Clustered)

[문제 5]

① 외래키(Foreign Key) ② 후보키(Candidate Key) ③ 기본키(Primary Key)

[문제 6]

ODBC(Open DataBase Connectivity)

[문제 7]

디스패치(Dispatch)

[문제 8]

LOC(원시 코드 라인 수)

[문제 9]

rand()

※ 답안 작성 시 주의 사항

프로그래밍 언어에서 사용하는 변수, 함수의 이름은 대소문자를 구분하기 때문에 변수, 함수 이름을 작성할 때는 대소문자를 구분해서 정확히 작성해야 합니다.

해설

```
#include <stdio.h>
#include <stdlib.h>
main( ) {
    for (int i = 0; i < 10; i++) {
        printf("%d\n", (1 + rand( ) % 70));
    }
}
```

rand() 함수가 정의되어 있는 헤더 파일이다.

반복 변수 i가 0에서 시작하여 1씩 증가하면서 10보다 작은 동안 다음 문장을 반복 수행한다. 즉 다음 문장을 10회 반복 수행한다.
rand() 함수로 발생한 난수를 70으로 나눈 나머지에 1을 더하여 출력한다. 즉 1~70 사이의 난수를 출력한다.



[문제 10]

r
t

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값을 한 줄로 r, t 혹은 r t로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
public class Test {
    public static void main(String[] args) {
        ① String str = "Operation";
        ②⑨ String rst = change(str);
        ⑩ System.out.println(str.charAt(3));
        ⑪ System.out.println(rst.charAt(3));
    }

    ③ static String change(String x) {
        ④ String y = new String();
        ⑤ int n = x.length() - 1;
        ⑥ for(int i = n; i >= 0; i--)
        ⑦     y += x.charAt(i);
        ⑧ return y;
    }
}
```

- ① 문자열 변수 str을 선언하고 "Operation"으로 초기화한다.
- ② 문자열 변수 rst를 선언하고, str의 값 "Operation"을 인수로 하여 change() 메소드를 호출한 다음 돌려받은 값을 rst에 저장한다.
- ③ 메소드의 리턴값이 문자열인 change() 메소드의 시작점이다. ②번에서 전달받은 "Operation"을 문자열 변수 x로 받는다.
- ④ 문자열 변수 y를 선언한다. y는 배열과 동일한 객체 변수이므로 초기값이 없을 때는 new 예약어를 사용한다.
- ⑤ 정수형 변수 n을 선언하고, 문자열 변수 x의 길이에서 1을 뺀 8(9-1)로 초기화한다.
- ⑥ 반복 변수 i가 n에서 시작하여 1씩 감소하면서 0보다 크거나 작은 동안 ⑦번을 반복하여 수행한다.
- ⑦ 'y = y + x.charAt(i)'와 동일하다. y의 값에 x에서 i번째에 있는 문자를 더한다. 문자 간의 더하기(+) 연산은 앞의 문자에 뒤의 문자를 붙여서 반환한다.
(예) 'abc' + 'y' = 'abcy')

- ⑧ y의 값을 호출한 곳(main() 메소드)으로 반환한다.
- ⑨ change() 메소드로부터 반환받은 값을 rst에 저장한다.
- ⑩ str의 3번째에 있는 문자를 출력한 후 다음 줄로 이동한다.
- ⑪ rst의 3번째에 있는 문자를 출력한 후 다음 줄로 이동한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

x	n	i	x.charAt(i)	y	출력
Operation	8	8	n	n	r
		7	o	no	t
		6	i	noi	
		5	t	noit	
		4	a	noita	
		3	r	noitar	
		2	e	noitare	
		1	p	noitarep	
		0	O	noitarepO	
		-1			

[답안 작성 방법 안내]

'운영체제(OS; Operation System)'처럼 한글과 영문으로 제시되어 있는 경우 '운영체제', 'OS', 'Operation System' 중 1가지만 쓰면 됩니다.

[문제 11]

스프링 배치(Spring Batch)

[문제 12]

① 콤보 박스(Combo Box) ② 라디오 박스(Radio Box)

[문제 13]

함수적 종속(Functional Dependency)

[문제 14]

DDoS(Distributed Denial of Service, 분산 서비스 거부)

[문제 15]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

DNS는 문자로 된 도메인 네임을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템이다.

[문제 16]

워킹 셋(Working Set)

[문제 17]

9 2

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 9, 2로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

문제의 코드는 배열에서 입력받은 수 7개 가장 가까운 수를 찾아 출력하는 프로그램입니다. 7과 가장 가까운 수를 구하려면 7과 다른 수들의 차를 계산한 후 차이를 비교하여 차이가 가장 작은 수를 찾으면 됩니다. 주의할 점은 차이를 계산할 때 음수가 나오면 안 되므로 7과 비교할 값의 대·소를 비교한 후 큰 수에서 작은 수를 빼야 합니다.

```
#include <stdio.h>
main() {
    ① int find, tmp, result, dif = 99;
    ② int arr[ ] = { 5, 3, 9, 14, 1, 12 };
    ③ scanf("%d", &find);
    ④ for (int x = 0; x < 6; x++) {
        ⑤ tmp = arr[x] > find ? arr[x] - find : find - arr[x];
        ⑥ if (tmp <= dif) {
            ⑦ result = arr[x];
            ⑧ dif = tmp;
        }
    }
    ⑨ printf("%d %d", result, dif);
}
```



- ① 정수형 변수 find, tmp, result, dif를 선언하고, dif를 99로 초기화한다.
 - ② 6개의 요소를 갖는 정수형 배열 arr을 선언하고 초기화한다.
 - ③ 정수를 입력받아 find에 저장한다.
 - ④ 반복 변수 x가 0에서 시작하여 1씩 증가하면서 6보다 작은 동안 ⑤~⑧번을 반복하여 수행한다.
 - ⑤ arr[x]의 값이 find의 값보다 크면 tmp에 arr[x]의 값에서 find의 값을 뺀 값을 저장하고, 아니면 tmp에 find의 값에서 arr[x]의 값을 뺀 값을 저장한다.
 - ⑥ tmp의 값이 dif의 값보다 작거나 같으면 ⑦, ⑧번을 수행하고, 아니면 반복문의 처음인 ④번으로 돌아간다.
 - ⑦ result에 arr[x]의 값을 저장한다.
 - ⑧ dif에 tmp의 값을 저장한다.
 - ⑨ result를 출력하고 공백 한 칸을 띄운다. 이어서 dif의 값을 출력한다.
- 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

	arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]
arr	5	3	9	14	1	12

find	x	arr[x]	tmp	result	dif	출력
7	0	5	2	5	99	9 2
	1	3	4	9	2	
	2	9	2		2	
	3	14	7			
	4	1	6			
	5	12	5			
	6					

[문제 18]

SOAP(Simple Object Access Protocol)

[문제 19]

① INTO ② VALUES

※ 답안 작성 시 주의 사항

대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지 마세요.

해설

INSERT INTO 사원(성명, 경력)

VALUES('홍길동', 10);

〈사원〉 테이블의 '성명', '경력'에 삽입한다.

'성명'에 '홍길동'을, '경력'에 10을 삽입한다.

[문제 20]

경계값 분석(Boundary Value Analysis)



최종점검 모의고사 정답 및 해설

[답안 작성 방법 안내]

‘운영체제(OS; Operation System)’처럼 한글과 영문으로 제시되어 있는 경우 ‘운영체제’, ‘OS’, ‘Operation System’ 중 1가지만 쓰면 됩니다.

[문제 1]

오픈 소스(Open Source)

[문제 2]

자료 흐름도(DFD; Data Flow Diagram)

[문제 3]

① 수평 분할(Horizontal Partitioning) ② 수직 분할(Vertical Partitioning)

[문제 4]

1 2 5 10

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 코마를 넣어 1, 2, 5, 10으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

문제의 코드는 입력받은 수의 약수를 배열에 저장한 후 출력하는 프로그램이다. 10의 약수를 구한다고 가정했을 때, 10의 약수는 10을 1부터 10까지 수로 차례대로 나누어 나머지가 0이 되게 하는 수 1, 2, 5, 10이 약수가 된다.

```
#include <stdio.h>
main() {
    ① int a, b, k[10];
    ② scanf("%d", &a);
    ③ ④ b = aliquot(a, k);
    ⑤ for (int j = 0; j < b; j++)
    ⑥     printf("%d ", k[j]);
}
⑦ int aliquot(int a, int k[]) {
    ⑧ int cnt = 0;
    ⑨ for (int i = 1; i <= a; i++)
    ⑩     if (a % i == 0) {
        ⑪ k[cnt] = i;
        ⑫ cnt++;
    }
    ⑬ return cnt;
}
```

- ① 정수형 변수 a, b와 10개의 요소를 갖는 정수형 배열 k를 선언한다.
- ② 정수를 입력받아 a에 저장한다.
- ③ a의 값과 배열 k의 시작 주소를 인수로 aliquot 함수를 호출한 다음 돌려받은 값을 b에 저장한다.
- ④ 리턴 값이 정수인 aliquot 함수의 시작점이다. ⑤번에서 전달받은 값과 주소는 각각 a와 배열 k가 받는다.
- ⑥ 정수형 변수 cnt를 선언하고 0으로 초기화한다.
- ⑦ 반복 변수 i가 1에서 시작하여 1씩 증가하면서 a보다 작거나 같은 동안 ⑧~⑩번을 반복 수행한다.
- ⑧ a를 i로 나눈 나머지가 0이면 ⑨, ⑩번 문장을 수행하고, 아니면 반복문의 처음인 ⑥번으로 돌아간다.
- ⑪ k[cnt]에 i의 값을 저장한다.



- ⑨ 'cnt = cnt + 1;'과 동일하다. cnt의 값을 1씩 누적한다.
 - ⑩ cnt의 값 4를 호출한 곳으로 반환한다.
 - ⑪ aliquot 함수로부터 반환받은 값 4를 b에 저장한다.
 - ⑫ 반복 변수 j가 0에서 시작하여 1씩 증가하면서 b의 값인 4보다 작은 동안 ⑬번 문장을 반복 수행한다.
 - ⑬ k[j]의 값을 출력한다.
- ※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

	k[0]	k[1]	k[2]	k[3]	k[4]	k[5]	k[6]	k[7]	k[8]	k[9]
k[]	1	2	5	10						

main 함수				aliquot 함수				출력
a	b	j	k[j]	a	i	cnt	k[cnt]	
10	4	0	1	10	1	0	1	1 2 5 10
		1	2		2	1	2	
		2	5		3	2	5	
		3	10		4	3	10	
		4			5	4		
					6			
					7			
					8			
					9			
					10			
					11			

[문제 5]

갱신 이상(Update Anomaly)

[문제 6]

데이터 마트(Data Mart)

[문제 7]

① 5 ② 4

[문제 8]

① Hub & Spoke ② Message Bus 또는 ESB 방식

[문제 9]

sw *= -1

※ 답안 작성 시 주의 사항

sw 변수가 for문이 반복될 때마다 1, -1, 1, -1로 변화할 수 있는 어떠한 식도 정답이 될 수 있습니다. 예를 들어 sw = sw * -1로 작성하거나, 음수 기호만 붙여 sw = -sw로 작성해도 답이 됩니다.

해설

문제의 코드는 5행 5열의 배열에 'ㄱ'자 형태로 1부터 25까지의 수를 저장하여 출력하는 프로그램이다. 바깥쪽 for문의 i는 행 위치를, 안쪽 while문의 j는 열 위치를 담당하며, 안쪽 while문이 수행될 때마다 j의 증가값 역할을 하는 sw의 값을 +1 → -1 → +1 → -1 → +1로 바꾸는 과정을 통해 <출력>과 같이 배열에 값을 저장할 수 있다.

```
#include <stdio.h>
main() {
    ① int a[5][5], j = 0, sw = 1, k = 1;
    ② for (int i = 0; i < 5; i++) {
        ③ while (j <= 4 && j >= 0) {
            ④ a[i][j] = k++;
            ⑤ j += sw;
        }
        ⑥ sw *= -1;
        ⑦ j += sw;
    }

    ⑧ for (int i = 0; i < 5; i++) {
        ⑨ for (int j = 0; j < 5; j++)
        ⑩ printf("%3d", a[i][j]);
        ⑪ printf("\n");
    }
}
```

- ① 5행 5열의 크기를 갖는 정수형 배열 a와 정수형 변수 j, sw, k를 선언하고 각각 0, 1, 1로 초기화한다.
 ② 반복 변수 i가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ③~⑦번을 반복 수행한다.
 ③ j가 4보다 작거나 같고 0보다 크거나 같은 동안 ④, ⑤번을 반복 수행한다.
 ④ a[i][j]에 k의 값을 저장한다. k는 후치 증가 연산이므로 연산을 마치면 1이 증가한다.
 ⑤ 'j = j + sw'와 같다. j에 sw의 값을 누적한다.
 ⑥ 'sw = sw * -1'과 같다. sw의 값에 -1을 곱한 값을 sw에 저장한다.
 ⑦ 'j = j + sw'와 같다. j에 sw의 값을 누적한다.
 ※ ②~⑦번 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	j	sw	k	배열 a
	0	1	1	
0	1	-1	2	1 2 3 4 5
	2		3	
	3		4	
	4		5	
	5		6	
1	3	1	7	1 2 3 4 5
	2		8	10 9 8 7 6
	1		9	
	0		10	
	-1		11	
2	1	-1	12	1 2 3 4 5
	2		13	10 9 8 7 6
	3		14	11 12 13 14 15
	4		15	
	5		16	
	4			



3	3 2 1 0 -1 0	1	17 18 19 20 21	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr> <tr><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	2	3	4	5	10	9	8	7	6	11	12	13	14	15	20	19	18	17	16					
1	2	3	4	5																									
10	9	8	7	6																									
11	12	13	14	15																									
20	19	18	17	16																									
4	1 2 3 4 5 4	-1	22 23 24 25 26	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td></tr> <tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr> <tr><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr> <tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr> </table>	1	2	3	4	5	10	9	8	7	6	11	12	13	14	15	20	19	18	17	16	21	22	23	24	25
1	2	3	4	5																									
10	9	8	7	6																									
11	12	13	14	15																									
20	19	18	17	16																									
21	22	23	24	25																									
5																													

- ㉓ 반복 변수 i 가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ㉓~㉓번을 반복 수행한다.
- ㉔ 반복 변수 j 가 0에서 시작하여 1씩 증가하면서 5보다 작은 동안 ㉔번을 반복 수행한다.
- ㉕ 3칸을 확보하여 $a[i][j]$ 의 값을 출력한다.
- ㉖ 다음 행을 출력하기 위해 커서를 다음 줄의 처음으로 옮긴다.

[문제 10]

동적 SQL(Dynamic SQL)

[문제 11]

- ① Quartz ② Cron

[문제 12]

스토리보드(Story Board)

[문제 13]

1 2 4 7

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 1, 2, 4, 7로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

문제의 코드는 선택 정렬 알고리즘을 이용하여 배열의 값들을 오름차순으로 정렬한 후 출력하는 프로그램이다. 선택 정렬은 첫 번째 자료를 두 번째 자료부터 마지막 자료까지 차례대로 비교하여 가장 작은 값을 찾아 첫 번째에 놓고, 두 번째 자료를 세 번째 자료부터 마지막 자료까지 차례대로 비교하여 그 중 가장 작은 값을 찾아 두 번째 위치에 놓는 과정을 반복하며 정렬을 수행한다. 1회전을 수행하고 나면 가장 작은 값의 자료가 맨 앞에 오게 되므로 그 다음 회전에서는 두 번째 자료를 가지고 비교한다. 마찬가지로 3회전에서는 세 번째 자료를 정렬한다.

```
public class Test {
    ㉓ static void arr(int[] a) {
    ㉔     int sw, temp, n = a.length;
    ㉕     for(int i = 0; i < n - 1; i++) {
    ㉖         sw = i;
    ㉗         for(int j = i + 1; j < n; j++)
    ㉘             if(a[j] < a[sw])
    ㉙                 sw = j;
```



```

10      temp = a[i];
11      a[i] = a[sw];
12      a[sw] = temp;
    }
  }
  public static void main(String[] args) {
1   int n[] = { 4, 2, 7, 1 };
2   arr(n);
3   for(int i:n)
4     System.out.printf("%d ", i);
  }
}

```

모든 Java 프로그램은 main() 메소드부터 시작해야 한다.

- ① 4개의 요소를 갖는 정수형 배열 n을 선언하고 초기화한다.
- ② 배열 n의 시작 주소를 인수로 하여 arr() 메소드를 호출한다.
- ③ 리턴 값이 없는 arr() 메소드의 시작점이다. ②번에서 전달받은 주소를 배열 a가 받는다.
- ④ 정수형 변수 sw, temp, n을 선언하고, n의 값을 배열 a의 길이인 4로 초기화한다.
 - length : 배열 클래스의 속성으로 배열 요소의 개수가 저장되어 있다. a 배열은 4개의 요소를 가지므로 a.length는 4를 가지고 있다.
- ⑤ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 n-1보다 작은 동안 ⑥~⑫번을 반복 수행한다.
- ⑥ sw에 i의 값을 저장한다.
- ⑦ 반복 변수 j가 i+1에서 시작하여 1씩 증가하면서 n보다 작은 동안 ⑧, ⑨번을 반복 수행한다.
- ⑧ a[j]의 값이 a[sw]의 값보다 작으면 ⑩번을 실행한다.
- ⑨ sw에 j의 값을 저장한다.
- ⑩~⑫ 임시 변수 temp를 사용하여 a[i]와 a[sw]의 값을 교환한다.
- ⑬ 배열 n의 요소 수만큼 ⑭번을 반복 수행하는 향상된 for문이다.
 - int i : 배열 n의 각 요소가 할당될 변수를 선언한다.
 - n : 배열의 이름을 적어준다. 배열이 4개의 요소를 가지므로 각 요소를 i에 할당하면서 ⑭번을 4회 수행한다.
- ⑭ i의 값을 공백 한 칸과 함께 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

main() 메소드		arr() 메소드									출력
i	n[]	a[]	n	i	j	sw	a[j]	a[sw]	a[i]	temp	
	4 2 7 1	4 2 7 1	4								
		4 2 7 1 1 4		0	1 2 3 4	0 1 3	2 7 1	4 2 2 4	1	4	
		1 2 7 4 2		1	2 3 4	1	7 4	2 2 2	2	2	
		1 2 7 4 4 7		2	3 4	2 3	4	7 7	4	7	
				3							
1 2 4 7	1 2 4 7										1 1 2 1 2 4 1 2 4 7



[문제 14]

① 검증 테스트(Verification Test) ② 확인 테스트(Validation Test)

[문제 15]

테스트 케이스(Test Case)

[문제 16]

① VIEW ② AS SELECT

※ 답안 작성 시 주의 사항

대 · 소문자를 구분하지 않습니다. 단 스펠링이 하나라도 틀렸을 경우 부분 점수 1도 없는 오답으로 처리된다는 것을 잊지마세요.

해설

```
CREATE VIEW 학생_v AS
SELECT 번호, 이름, 학과
FROM 학생;
```

생성한 뷰의 이름은 <학생_v>이다.
'번호', '이름', '학과' 속성을 가져온다.
<학생> 테이블에서 속성을 가져와 뷰를 생성한다.

[문제 17]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

티어드롭 공격은 분할된 패킷의 순서를 기록하는 Fragment Offset 값을 변경하여 수신 측에서 재조립 시 오류로 인한 과부하를 유도하는 공격 방법이다.

[문제 18]

SEED

[문제 19]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

국부성은 프로세스가 실행되는 동안 주기억장치를 참조할 때 일부 페이지만 집중적으로 참조하는 성질을 말한다.

[문제 20]

에이징(Aging) 기법