

# 정보처리기사

실기

길벗알앤디 지음

2025  
시나공



2024년 8월 정보처리기능사 실기 .....	4
2024년 6월 정보처리기능사 실기 .....	24
2024년 3월 정보처리기능사 실기 .....	50
2023년 11월 정보처리기능사 실기 .....	68
2023년 8월 정보처리기능사 실기 .....	86
2023년 6월 정보처리기능사 실기 .....	106
2023년 3월 정보처리기능사 실기 .....	124
2022년 11월 정보처리기능사 실기 .....	140
2022년 8월 정보처리기능사 실기 .....	159
2022년 5월 정보처리기능사 실기 .....	175

# 최신기출문제

2024년 3회 정보처리기능사 실기

2024년 2회 정보처리기능사 실기

2024년 1회 정보처리기능사 실기

2023년 4회 정보처리기능사 실기

2023년 3회 정보처리기능사 실기

2023년 2회 정보처리기능사 실기

2023년 1회 정보처리기능사 실기

2022년 4회 정보처리기능사 실기

2022년 3회 정보처리기능사 실기

2022년 2회 정보처리기능사 실기

## 동영상 강의 시청 방법

다음의 세 가지 방법을 이용하여 시나공 저자의 속 시원한 강의를 바로 동영상으로 확인하세요.

### 하나 스마트폰으로는 이렇게 이용하세요!

1. 스마트폰으로 QR코드 리더 앱을 실행하세요!
2. 동영상 강의 QR코드를 스캔하세요.
3. 스마트폰을 통해 동영상 강의가 시작됩니다!

### 둘 시나공 홈페이지에서는 이렇게 이용하세요!

1. 시나공 홈페이지(sinagong.co.kr)에 로그인 하세요!
2. 상단 메뉴중 [정보처리] → [기능사 실기] → [동영상 강좌] → [토막강의]를 클릭하세요!
3. 동영상 강의 번호를 입력하면 동영상 강의가 시작됩니다.

### 셋 유튜브에서는 이렇게 이용하세요!

1. 유튜브 검색 창에 “시나공” + 동영상 강의 번호를 입력하세요.

예 시나공5640301 

2. 검색된 항목 중 원하는 동영상 강의를 클릭하여 시청하세요.



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

**문제 1** 다음 설명에 해당하는 UNIX 명령어를 쓰시오. (5점)

5640301



네트워크 인터페이스를 구성하고 관리하기 위해 사용하는 명령어이다. 이 명령어는 네트워크 인터페이스의 현재 상태를 확인하고 IP 주소를 설정하거나 변경하는 용도로 사용하며, Windows의 ipconfig 명령과 유사하다.

**답 :****문제 2** 데이터베이스 시스템을 관리하고 운영에 관한 모든 것을 책임지며, 다음과 같은 역할을 수행하는 사람이나 그룹에 해당하는 용어를 쓰시오. (5점)

5640302



- 데이터베이스의 스키마를 정의, 생성, 삭제한다.
- 데이터베이스 구성 요소를 결정한다.
- 데이터베이스의 저장 구조 및 접근 방법을 정의한다.
- 보안 및 데이터베이스의 접근 권한 부여 정책을 수립한다.
- 장애에 대비한 예비(Back Up) 조치와 회복(Recovery)에 대한 전략을 수립한다.
- 무결성을 위한 제약 조건을 지정한다.
- 데이터 사전을 구성하고 유지 관리한다.
- 사용자의 변화 요구와 성능 향상을 위해 데이터베이스를 재구성한다.

**답 :****문제 3** 다음 설명에 해당하는 용어를 쓰시오. (5점)

5640303



데이터베이스에서 아직 알려지지 않거나 모르는 값으로서 '해당 없음' 등의 이유로 정보 부재를 나타내기 위해 사용하는, 이론적으로 아무것도 없는 특수한 데이터를 말한다.

**답 :**



**문제 4** 다음과 같이 <Product>와 <Store> 테이블을 이용하여 <Pro\_Sto\_view> 뷰를 생성한 다음 <SQL> 문을 수행한 뒤 <결과>의 ㉠에 표시되는 값을 숫자로 쓰시오. (5점)

<Product>

Product_ID	Price	Store_ID
Pro1234	1,500	LH123
Pro2583	2,000	LH345
Pro8519	3,500	LH456
Pro3845	2,500	LH567

<Store>

Store_ID	Store_ADR
LH123	합정동
LH345	서교동
LH456	망원동
LH567	성산동

<VIEW 생성> 문

```
CREATE VIEW Pro_Sto_view AS (  
  SELECT P.Product_ID, P.Price, S.Store_ID, S.Store_ADR  
  FROM Product P JOIN Store S  
  ON P.Product_ID = S.Store_ID  
);
```

<SQL> 문

```
SELECT COUNT(*) FROM Pro_Sto_view  
GROUP BY Product_ID;
```

<결과>

COUNT(*)
㉠

**답 :**

5640305



**문제 5** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>

int main( ) {
    int i, j, sum = 0;
    for (i = 2; i <= 5; i++)
        for (j = 1; j < i; j++)
            sum = sum + j;
    printf("%d", sum);
    return 0;
}
```

답 :

5640306



**문제 6** 다음 Java 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
public class Main {
    public static void main(String[] args) {
        int o = 0, e = 0, cnt;
        String ans = "149854322124";
        for (int i = 0; i < ans.length(); i++)
            if (Character.getNumericValue(ans.charAt(i)) % 2 == 1)
                o++;
            else
                e++;
        cnt = e - o;
        System.out.printf("%d", cnt);
    }
}
```

답 :

**문제 7** 현재 작업중인 디렉터리 경로를 화면에 표시하는 UNIX 명령어를 쓰시오. (5점)



답 :

**문제 8** 다음 <보기>에서 전송 계층 프로토콜이 아닌 것을 모두 골라 기호(㉠~㉦)로 쓰시오. (5점)



<보기>

㉠ tcp	㉡ ftp	㉢ udp	㉣ dccp	㉤ ipsec	㉦ http
-------	-------	-------	--------	---------	--------

답 :

**문제 9** OSI 7계층 중 다음과 같은 특징을 갖는 계층은 무엇인지 쓰시오. (5점)



- 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의한다.
- 물리적 전송 매체와 전송 신호 방식을 정의한다.
- RS-232C, X.21 등의 표준이 있다.
- 프로토콜 데이터 단위(PDU)는 비트(Bit)이다.
- 관련 장비 : 리피터, 허브

답 :

**문제 10** 다음 설명에 해당하는 용어를 쓰시오. (5점)



- 아직 COMMIT되지 않은 변경된 모든 내용들을 취소하고 데이터베이스를 이전 상태로 되돌리는 명령어이다.
- 트랜잭션 전체가 성공적으로 끝나지 못하면 일부 변경된 내용만 데이터베이스에 반영되는 비일관성(Inconsistency)인 상태를 가질 수 있기 때문에 일부분만 완료된 트랜잭션은 이 명령이 수행되어야 한다.

답 :



**문제 11** 다음 Java 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

5640311



```
public class Main {
    public static void main(String[] args) {
        int num1 = 3, num2 = 7;

        if (++num1 < 5 || ++num2 > 5)
            System.out.println(num1);
        System.out.println(num2);
    }
}
```

답 :

**문제 12** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

5640312



```
#include <stdio.h>
#define MAX 10

int main( ) {
    int cur = 1, prev = 0, tmp;

    for (int i = 0; i < MAX; i++)
    {
        printf("%d", cur);
        tmp = prev;
        prev = cur;
        cur = cur + tmp;
        if (i%5 == 4)
            printf("\n");
        else
            printf("-");
    }
    return 0;
}
```

답 :



**문제 13** 다음에서 설명하는 내용에 해당하는 데이터베이스 관련 용어를 쓰시오. (5점)

- 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
- 어떤 데이터 개체의 구성 요소로서 그 개체의 성질이나 상태를 기술해 주는 역할을 하며, 그 자체로는 중요한 의미를 가지지 못한다.
- 릴레이션에서는 열(Column)에, 파일 시스템에서는 필드(Field)에 해당한다.
- 개체를 구성하는 항목이다.

**답 :**



**문제 14** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>
#define SIZE 4

int main( ) {
    int test[SIZE][SIZE] = {0, };
    int i, j;
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE ; j++) {
            if (i >= j)
                test[i][j] = i-j;
            else
                test[i][j] = j-i;
        }
    }
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++) {
            printf("%d", test[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

**답 :**

**문제 15** 규정 시간 또는 시간 조각(Slice)을 미리 정의하여 CPU 스케줄러가 준비 상태 큐에서 정의된 시간만큼 각 프로세스에 CPU를 제공하는 시분할 시스템에 적절한 스케줄링 기법은 무엇인지 쓰시오. (5점)

5640315



**답 :**

**문제 16** 다음에서 설명하는 내용에 해당하는 테스트 기법을 쓰시오. (5점)

5640316



- 소프트웨어의 내부 구조나 작동 원리를 모르는 상태에서 소프트웨어의 동작을 테스트하는 방법이다.
- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.
- 사용자의 요구사항 명세를 보면서 테스트하는 것으로, 주로 구현된 기능을 테스트한다.
- 소프트웨어 인터페이스에서 실시되는 테스트이다.
- 부정확하거나 누락된 기능, 인터페이스 오류, 초기화와 종료 오류 등을 발견하기 위해 사용되며, 테스트 과정의 후반부에 적용된다.
- 종류에는 동치 분할 검사, 경계값(한계값) 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등이 있다.

**답 :**

**문제 17** 다음 설명에 해당하는 프로토콜을 쓰시오. (5점)

5640317



- 전자 우편을 전송하는 프로토콜이다.
- TCP 포트번호는 25번이다.
- 메일 서버간의 송수신뿐만 아니라, 메일 클라이언트에서 메일 서버로 메일을 보낼 때에도 사용된다.

**답 :**



**문제 18** 다음의 <급여> 테이블의 'salary' 필드의 값을 변경하고자 한다. <처리조건>에 부합하는 <SQL> 문이 완성되도록 괄호에 적합한 명령을 쓰시오. (5점)

<급여>

id	name	salary	dept
123	최준호	1500	마케팅
456	유현숙	1800	경영
789	최규대	2200	회계
012	한경선	1900	기획

<처리조건>

- 갱신을 위해 UPDATE문을 사용한다.
- 'salary'가 2000 미만인 사원의 'salary'를 200 인상한다.

<SQL> 문

UPDATE 급여 ( ) salary = salary + 200 WHERE salary < 2000;

답 :



**문제 19** 다음은 <학생>과 <성적> 테이블을 이용하여 학년별 점수의 평균을 조회하는 <SQL>문이다. 괄호에 들어갈 알맞은 명령을 적어 <SQL>문을 완성하십시오. (5점)

<학생>

학번	이름	학년
K1	문지영	1
K2	강순동	1
K3	이지연	2
K4	이영주	2

<성적>

학번	점수
K1	90
K1	85
K2	80
K3	70
K3	75
K4	88

<SQL문>

SELECT 학년, AVG(점수) AS 평균점수  
FROM 학생 JOIN 성적 ( ) 학생.학번 = 성적.학번  
GROUP BY 학년;

답 :

**문제 20** 다음 Java 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

5640320



```
class q1{
    private String ans;

    q1( ) {
        this.ans = "";
    }

    void test( ) {
        System.out.print(ans);
    }

    <T> void test(T i) {
        this.ans += i.toString( );
    }

    void test(int i) {
        this.ans += i*2;
    }
}

public class Main {
    public static void main(String[] args) {
        q1 a = new q1( );
        a.test("abs");
        a.test( );
        a.test(1.0);
        a.test(2);
    }
}
```

**답 :**

[문제 1]  
ifconfig

[문제 2]

※ 다음 중 하나를 쓰면 됩니다.

데이터베이스 관리자, DBA, DataBase Administrator

[문제 3]

※ 다음 중 하나를 쓰면 됩니다.

널, 널 값, null, null value

[문제 4]

0

#### 해설

##### 〈VIEW 생성〉문

```
CREATE VIEW Pro_Sto_view AS (
    SELECT P.Product_ID, P.Price, S.Store_ID, S.Store_ADR

    FROM Product P JOIN Store S

    ON P.Product_ID = S.Store_ID
);
```

생성한 뷰의 이름은 〈Pro\_Sto\_view〉이다.  
〈Product〉 테이블의 'Product\_ID', 'Price' 필드와 〈Store〉 테이블의 'Store\_ID', 'Store\_ADR' 필드를 가져온다.  
〈P〉라는 별칭의 〈Product〉와 〈S〉라는 별칭의 〈Store〉 테이블을 이용한다.  
〈Product〉 테이블의 'Product\_ID'와 〈Store〉 테이블의 'Store\_ID'를 기준으로 서로 JOIN한다.

##### 〈SQL〉문

```
SELECT COUNT(*)
FROM Pro_Sto_view
GROUP BY Product_ID;
```

조회된 레코드의 개수를 표시한다.  
〈Pro\_Sto\_view〉 뷰에서 검색한다.  
'Product\_ID'를 기준으로 그룹을 지정한다.

※ VIEW 생성 시 〈Product〉 테이블의 'Product\_ID'와 〈Store〉 테이블의 'Store\_ID'를 기준으로 서로 JOIN하지만 〈Product〉 테이블의 'Product\_ID'와 〈Store〉 테이블의 'Store\_ID'에는 서로 다른 값이 저장되어 있으므로 생성된 〈Pro\_Sto\_view〉 뷰에는 레코드가 없습니다. 그러므로 〈SQL〉문을 수행한 뒤 〈결과〉의 ㉠에 표시되는 값은 0입니다.

## [문제 5]

20

## 해설

```
#include <stdio.h>

int main( ) {
{
① int i, j, sum = 0;
② for (i = 2; i <= 5; i++)
③ for (j = 1; j < i; j++)
④     sum = sum + j;
⑤ printf("%d", sum);
⑥ return 0;
}
```

- ① 정수형 변수 i, j, sum을 선언하고, sum을 0으로 초기화한다.  
 ② 반복 변수 i가 2부터 1씩 증가하면서 5보다 작거나 같은 동안 ③번을 반복 수행한다.  
 ③ 반복 변수 j가 1부터 1씩 증가하면서 i보다 작은 동안 ④번을 반복 수행한다.  
 ④ sum에 j의 값을 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	j	sum
2	1	0
	2	1
3	1	2
	2	4
	3	
4	1	5
	2	7
	3	10
	4	
5	1	11
	2	13
	3	16
	4	20
	5	
6		

- ⑤ sum의 값 20을 출력한다.

결과 20

- ⑥ main() 함수에서의 return 0은 프로그램의 종료를 의미한다.

[문제 6]

2

해설

```
public class Main {
    public static void main(String[] args) {
        ① int o = 0, e = 0, cnt;
        ② String ans = "149854322124";
        ③ for (int i = 0; i < ans.length(); i++)
        ④     if (Character.getNumericValue(ans.charAt(i)) % 2 == 1)
        ⑤         o++;
        ⑥     else
        ⑦         e++;
        ⑧ cnt = e - o;
        System.out.printf("%d", cnt);
    }
}
```

- ① 정수형 변수 o, e, cnt를 선언하고, o와 e를 0으로 초기화한다.
- ② 문자열 변수 ans를 선언하고 "149854322124"로 초기화한다.
- ③ 반복 변수 i가 0부터 1씩 증가하면서 ans 문자열의 길이인 12보다 작은 동안 ④번을 반복 수행한다.
- **length**: 문자열 클래스의 속성으로, 문자열의 길이가 저장되어 있음
- ④ ans 문자열에서 i번째 문자를 가져와 정수로 변환한 값을 2로 나눈 나머지가 1이면, ⑤번을 수행하고 아니면 ⑥번을 수행한다.
- **getNumericValue()**: 숫자 형태의 문자열을 정수형으로 변환하는 메소드로, 숫자 형태가 아닌 문자열인 경우에는 -1을, 분수처럼 정수형으로 표현할 수 없는 문자열인 경우에는 -2를 반환함
  - **charAt()**: 문자열에서 인수에 해당하는 위치의 문자를 반환하는 메소드
- ⑤ 'o = o + 1;'과 동일하다. o의 값에 1을 누적시킨다.
- ⑥ 'e = e + 1;'과 동일하다. e의 값에 1을 누적시킨다.
- 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	ans.charAt(i)	ans.charAt(i)%2	o	e	ans											
0	1	1	0	0	<div>[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]</div> <div>1 4 9 8 5 4 3 2 2 1 2 4</div>											
1	4	0	1	1												
2	9	1	2	2												
3	8	0	3	3												
4	5	1	4	4												
5	4	0	5	5												
6	3	1	6	6												
7	2	0	7	7												
8	2	0														
9	1	1														
10	2	0														
11	4	0														
12																

- ⑦ e - o의 값을 cnt에 저장한다.
- ⑧ cnt의 값 2를 출력한다.

결과 2

[문제 7]

PWD

※ 답안 작성 시 주의 사항 : UNIX에서 사용되는 명령어는 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.



## [문제 8]

㉞, ㉟, ㊱

## 해설

ftp와 http는 응용 계층, ipsec은 네트워크 계층의 프로토콜입니다.

## [문제 9]

※ 다음 중 하나를 쓰면 됩니다.

물리 계층, Physical Layer

## [문제 10]

ROLLBACK

## [문제 11]

4

7

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값 사이에 공백을 넣어 4 7이나 쉼표를 넣어 4, 7과 같이 작성했을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

```
public class Main {
    public static void main(String[] args) {
        ①      int num1 = 3, num2 = 7;

        ②      if (++num1 < 5 || num2++ > 5)
        ③          System.out.println(num1);
        ④          System.out.println(num2);
    }
}
```

① 정수형 변수 num1과 num2를 선언하고, 각각 3과 7로 초기화한다.

② num1을 1 증가한 값이 5보다 작거나 num2가 5보다 크면, ③번을 수행하고 아니면 ④번으로 이동한다. num1을 1 증가한 값이 4다. 즉 첫 번째 피연산자가 조건에 만족하므로 두 번째 피연산자를 확인하지 않고 바로 ③번을 수행한다.

※ 논리적 OR(||) 연산자는 피연산자 중 하나 이상이 참인 경우 참을 반환하는데, 피연산자 중 참이 확인되면, 이후 피연산자를 확인하지 않습니다.

③ num1의 값 4를 출력하고 커서를 다음 줄의 처음으로 옮긴다.

결과 4

④ num2의 값 7을 출력하고 커서를 다음 줄의 처음으로 옮긴다.

※ ②번에서 num2++이 수행되지 않았으므로 num2는 초기값 7이 그대로 유지된 상태입니다.

결과 4

7

## [문제 12]

1-1-2-3-5

8-13-21-34-55

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값 사이에 공백을 넣어 1-1-2-3-5 8-13-21-34-55와 같이 작성하거나, 쉼표를 넣어 1-1-2-3-5, 8-13-21-34-55와 같이 작성했을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

알고리즘의 이해

문제의 코드는 피보나치 수를 만들어 출력하는 것으로, 피보나치 수는 0과 1로 시작하며, 다음 피보나치 수는 바로 앞의 두 피보나치 수의 합이 됩니다.  
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

```
#include <stdio.h>
#define MAX 10 // 숫자 10을 MAX로 정의한다. 프로그램 안에서 MAX는 10과 동일하게 쓰인다.
int main( ) {
    ❶ int cur = 1, prev = 0, tmp;
    ❷ for (int i = 0; i < MAX; i++)
    {
        ❸ printf("%d", cur);
        ❹ tmp = prev;
        ❺ prev = cur;
        ❻ cur = cur + tmp;
        ❼ if (i%5 == 4)
        ❽ printf("\n");
        else
        ❾ printf("-");
    }
    ❿ return 0;
}
```

- ❶ 정수형 변수 cur, prev, tmp를 선언하고, cur을 1로, prev를 0으로 초기화한다.
- ❷ 반복 변수 i가 0부터 1씩 증가하면서 MAX, 즉 10보다 작은 동안 ❸~❿번을 반복 수행한다.
- ❸ cur의 값을 출력한다.
- ❹ prev의 값을 tmp에 저장한다.
- ❺ cur의 값을 prev에 저장한다.
- ❻ cur의 값에 tmp의 값을 누적시킨다.
- ❼ i를 5로 나눈 나머지가 4면 ❽번을 수행하고, 아니면 ❾번을 수행한다.
- ❽ 커서를 다음 줄의 처음으로 옮긴다.
- ❾ -를 출력한다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	tmp	prev	cur	i%5	출력
		0	1		
0	0	1	1	0	1
1	1	1	2	1	1-
2	1	2	3	2	1-1
3	2	3	5	3	1-1-
4	3	5	8	4	1-1-2
5	5	8	13	0	1-1-2-
6	8	13	21	1	1-1-2-3
7	13	21	34	2	1-1-2-3-
8	21	34	55	3	1-1-2-3-5
9	34	55	89	4	출 바꿈
10					8
					8-
					8-13
					8-13-
					8-13-21
					8-13-21-
					8-13-21-34
					8-13-21-34-
					8-13-21-34-55
					출 바꿈

⑩ main() 함수에서의 **return 0**은 프로그램의 종료를 의미한다.

### [문제 13]

※ 다음 중 하나를 쓰면 됩니다.

속성, Attribute

### [문제 14]

0123

1012

2101

3210

※ **답안 작성 시 주의 사항** : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값 사이에 공백을 넣어 0123 1012 2101 3210이나 점표를 넣어 0123, 1012, 2101, 3210과 같이 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

#### 해설

```
#include <stdio.h>
#define SIZE 4                                // 숫자 4를 SIZE로 정의한다. 프로그램 안에서 SIZE는 4와 동일하게 쓰인다.

int main( ) {
    ① int test[SIZE][SIZE] = {0, };
    ② int i, j;
    ③ for (i = 0; i < SIZE; i++) {
    ④     for (j = 0; j < SIZE ; j++) {
    ⑤         if (i >= j)
    ⑥             test[i][j] = i-j;
    ⑦         else
    ⑧             test[i][j] = j-i;
    ⑨     }
    ⑩ for (i = 0; i < SIZE; i++) {
    ⑪     for (j = 0; j < SIZE; j++) {
    ⑫         printf("%d", test[i][j]);
    ⑬     }
    ⑭     printf("\n");
    ⑮ }
    ⑯ return 0;
}
```

① SIZE가 4이므로, 4행 4열의 2차원 배열 test를 선언하고 모든 요소를 0으로 초기화한다.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

② 정수형 변수 i와 j를 선언한다.

③ 반복 변수 i가 0부터 1씩 증가하면서 4보다 작은 동안 ④~⑦번을 반복 수행한다.

④ 반복 변수 j가 0부터 1씩 증가하면서 4보다 작은 동안 ⑥~⑦번을 반복 수행한다.

⑤ i가 j보다 크거나 같으면 ⑥번을 수행하고, 아니면 ⑦번을 수행한다.

- ⑥  $i-j$ 의 값을  $test[i][j]$ 에 저장한다.  
⑦  $j-i$ 의 값을  $test[i][j]$ 에 저장한다.  
반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	j	i>=j	test[i][j]	test			
0	0	Yes	0	0	1	2	3
	1	No	1				
	2	No	2				
	3	No	3				
	4						
1	0	Yes	1	0	1	2	3
	1	Yes	0	1	0	1	2
	2	No	1				
	3	No	2				
	4						
2	0	Yes	2	0	1	2	3
	1	Yes	1	1	0	1	2
	2	Yes	0	2	1	0	1
	3	No	1				
	4						
3	0	Yes	3	0	1	2	3
	1	Yes	2	1	0	1	2
	2	Yes	1	2	1	0	1
	3	Yes	0	3	2	1	0
	4						
4							

- ⑧ 반복 변수  $i$ 가 0부터 1씩 증가하면서 4보다 작은 동안 ⑨, ⑩번을 반복 수행한다.  
⑨ 반복 변수  $j$ 가 0부터 1씩 증가하면서 4보다 작은 동안 ⑩번을 반복 수행한다.  
⑩  $test[i][j]$ 의 값을 정수형으로 출력한다.  
⑪ 커서를 다음 줄의 처음으로 옮긴다.  
반복문 실행에 따른 변수들의 변화는 다음과 같다.

배열 test	i	j	출력																
<table><tr><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td></tr><tr><td>2</td><td>1</td><td>0</td><td>1</td></tr><tr><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	0	1	2	3	1	0	1	2	2	1	0	1	3	2	1	0	0	0 1 2 3 4	0123
	0	1	2	3															
	1	0	1	2															
	2	1	0	1															
	3	2	1	0															
1	0 1 2 3 4	0123 1012																	
2	0 1 2 3 4	0123 1012 2101																	
3	0 1 2 3 4	0123 1012 2101 3210																	
4																			

- ⑫ `main()` 함수에서의 `return 0;`은 프로그램의 종료를 의미한다.

## [문제 15]

※ 다음 중 하나를 쓰면 됩니다.

라운드 로빈, Round Robin

## [문제 16]

※ 다음 중 하나를 쓰면 됩니다.

블랙박스 테스트, Black Box Test

## [문제 17]

※ 다음 중 하나를 쓰면 됩니다.

SMTP, Simple Mail Transfer Protocol, 간이 우편 전송 프로토콜

## [문제 18]

SET

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대 · 소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
UPDATE 급여
SET salary = salary + 200
WHERE salary < 2000;
```

〈급여〉 테이블을 갱신한다.  
'salary'를 200 인상한다.  
'salary'가 2000 미만인 자료만을 대상으로 한다.

SQL 실행 후 〈급여〉 테이블

id	name	salary	dept
123	최준호	1700	마케팅
456	유현숙	2000	경영
789	최규대	2200	회계
012	한경선	2100	기획

## [문제 19]

ON

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대 · 소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
SELECT 학년, AVG(점수) AS 평균점수
FROM 학생 JOIN 성적
ON 학생.학번 = 성적.학번
GROUP BY 학년;
```

'학년', '점수'의 평균을 표시하되, '점수'의 평균은 '평균점수'로 표시한다.  
〈학생〉과 〈성적〉 테이블을 이용하여 검색한다.  
〈학생〉 테이블의 '학번'과 〈성적〉 테이블의 '학번'을 기준으로 JOIN한다.  
'학년'을 기준으로 그룹을 지정한다.

SQL 실행 결과

학년	평균점수
1	85.0
2	77.67

## [문제 20]

abs

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, ABS와 같이 대문자로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

```
class q1{                                // q1 클래스를 정의한다.
  ① private String ans;

  ② q1( ) {
    ③   this.ans = "";
  }

  ④ void test( ) {
    ⑤   System.out.print(ans);
  }

  ⑥ <T> void test(T i) {
    ⑦   this.ans += i.toString( );
  }

  ⑧ void test(int i) {
    ⑨   this.ans += i*2;
  }
}

public class Main {
  public static void main(String[] args) {
    ⑩ q1 a = new q1( );
    ⑪ a.test("abs");
    ⑫ a.test( );
    ⑬ a.test(1.0);
    ⑭ a.test(2);
  }
}
```

① 문자열 변수 ans를 선언한다.

② q1 클래스의 생성자로, 객체를 생성할 때 자동으로 호출된다. 호출되면 ans를 빈 문자열("")로 초기화한다.

④ 반환값 없는 메소드 test( )를 정의한다.

⑥ 반환값 없는 제네릭 메소드 test(T i)를 정의한다. 호출되면 인수로 전달받은 값을 문자열로 변환하여 ans 변수에 추가한다.

※ 제네릭(Generic) 메소드는 다양한 자료형의 데이터를 처리할 수 있도록 일반화된 메소드입니다.

⟨T⟩ void test(T i)

· T: 자료형으로, 호출될 때의 자료형으로 자동 지정됨

· i: 인수

예를 들어, test("abs")가 호출되면, q1 클래스 안에는 문자열("abs")을 처리하는 test( ) 메소드가 없으므로 제네릭 메소드가 호출되는데, 이때 자료형은 호출될 때 인수로 문자열이 전달되었으므로 test(String i)와 같이 자료형이 문자열로 자동 지정됩니다.

⑧ 반환값 없는 메소드 test(int i)를 정의한다.

※ ④, ⑥, ⑧ 메소드 이름은 같지만 '인수를 받는 자료형'이 다르므로 서로 다른 메소드입니다. 즉 test( )와 test(T i), test(int i)는 다른 메소드라는 것입니다. 이렇게 이름은 같지만 인수를 받는 자료형을 달리하여 여러 기능을 정의하는 것을 오버로딩(Overloading)이라고 합니다.

모든 Java 프로그램은 반드시 `main()` 메소드에서 시작한다.

- ❶ `q1` 클래스의 객체 변수 `a`를 선언하면서 `q1` 클래스의 생성자를 호출한다.
- ❷ `q1` 클래스의 생성자 `q1()`의 시작점이다.
- ❸ `'q1.ans = ""`와 동일하다. `q1.ans`에 빈문자열("")을 저장한다. 생성자가 종료되었으므로 호출했던 ❶번의 다음 줄인 ❹번으로 이동한다.  
· **this** : 현재 실행중인 메소드가 속한 클래스를 가리키는 예약어
- ❹ 문자열 "abs"를 인수로 `a.test()`를 호출한다. `test()`라는 이름의 메소드 중 문자열을 인수로 받는 메소드가 없으므로 제너릭 메소드로 선언된 `test(T i)`가 호출된다. ❺번으로 이동한다.
- ❺ 반환값이 없는 제너릭 메소드인 `test(T i)`의 시작점이다. ❹번에서 전달받은 "abs"를 `i`가 받는다.
- ❻ `'q1.ans = q1.ans + i.toString()`과 동일하다. `i`를 문자열로 변환한 후 `ans` 변수에 추가한다. ❸번에서 `ans`에 빈문자열("")이 저장되었으므로 `ans`에는 "abs"가 저장된다. 메소드가 종료되면, 호출했던 ❹번의 다음 줄인 ❷번으로 이동한다.  
· **toString** : String 자료형으로 변환하는 메소드
- ❼ `a.test()` 메소드를 호출한다. 인수가 없으므로 ❸번으로 이동한다.
- ❽ 반환값이 없는 `test()` 메소드의 시작점이다.
- ❾ `ans`의 값 "abs"를 출력한다. 메소드가 종료되면, 호출했던 ❷번의 다음 줄인 ❶번으로 이동한다.

결과   **abs**

- ❿ `a.test(1.0)` 메소드를 호출한다. `test()`라는 이름의 메소드 중 실수 값을 인수로 받는 메소드가 없으므로 제너릭 메소드로 선언된 `test(T i)`가 호출된다. ❶번으로 이동한다.
  - ⓫ 반환값이 없는 제너릭 메소드인 `test(T i)`의 시작점이다. ❿번에서 전달받은 1.0을 `i`가 받는다.
  - ⓬ `'q1.ans = q1.ans + i.toString()`과 동일하다. `i`를 문자열로 변환한 후 `ans` 변수에 추가한다. ❸번에서 `ans`에 "abs"가 저장되었으므로 `ans`에는 "abs1.0"이 저장된다. 메소드가 종료되면, 호출했던 ❿번의 다음 줄인 ⓭번으로 이동한다.
  - ⓮ `a.test(2)` 메소드를 호출한다. 인수가 정수이므로 ❶번으로 이동한다.
  - ⓯ 반환값이 없는 `test(int i)` 메소드의 시작점이다. ⓮번에서 전달받은 2를 `i`가 받는다.
  - ⓰ `'q1.ans = q1.ans + i*2`와 동일하다. `'i*2`의 결과인 4를 `ans`에 추가한다. ⓬번에서 `ans`에 "abs1.0"이 저장되었으므로 `ans`에는 "abs1.04"가 저장된다. 메소드가 종료되면, 호출했던 ⓮번으로 이동한 후 프로그램을 종료한다.
- ※ 최종적으로 `ans`에는 "abs1.04"가 저장되어 있지만 ⓯번에서만 출력할 수 수행되므로 코드를 실행한 결과 화면에 출력되는 값은 **abs**가 됩니다.



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음



**문제 1** 다음 Java 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

5640201



```
public class Main
{
    public static void main(String[] args)
    {
        int num[] = {2, 1, 3, 7, 4, 9};
        int numb[] = new int[10];
        for (int i = 0; i < num.length; i++)
            numb[i] = num[i];
        for (int i : numb)
            System.out.print(i);
    }
}
```

**답 :**

**문제 2** 다음 Java 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

5640202



```
public class Main
{
    public static void main(String[] args)
    {
        int ans = 0;

        for (int i = 1; i < 100; i++)
        {
            if (i % 3 == 0)
                ans++;
            else if (i % 4 == 0)
                ans--;
        }
        System.out.println(ans);
    }
}
```

**답 :**



**문제 3** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
#include <stdbool.h>

bool test(int a) {
    bool ans = a < 150 ? true : false;
    return ans;
}

int fact(int i) {
    if (i < 2) return 1;
    else return i*fact(i-1);
}

int main( )
{
    int ans = 0;
    for (int i = 0; i < 10; i++)
        if (test(fact(i)))
            ans++;
    printf("%d", ans);
    return 0;
}
```

**답 :**



**문제 4** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```

#include <stdio.h>

void intswap(int *x, int *y)
{
    int t;

    t = *x;
    *x = *y;
    *y = t;
}

void testfunc(int *a, int *b, int *c)
{
    if (*c % 2 == 0)
        intswap(a, b);
    else
        intswap(a, c);
}

int gcd(int a, int b)
{
    if (b == 0)
        return 1;
    else
        return gcd(b, a % b);
}

int main( )
{
    int a = 5;
    int b = 4;
    int test1 = 15;
    int test2 = 12;
    int test = gcd(test1, test2);

    switch(test % 2)
    {
        case 1 :
            testfunc(&a, &b, &test2);
        default :
            testfunc(&a, &b, &test1);
    }
    printf("%d-%d", a, b);
    return 0;
}

```

답 :



**문제 5** 다음 Java 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
import java.lang.Math;
public class Main
{
    public static void main(String[] args)
    {
        int s, t, cnt = 0;
        int test[] = new int[5];
        for (int i = 0; i < 10; i++)
        {
            s = (int)Math.pow(i, 2);
            t = (s % 10);

            try {
                test[t] = i;
            }
            catch (ArrayIndexOutOfBoundsException e) {
                cnt++;
            }
        }
        System.out.println(cnt);
    }
}
```

**답 :**

**문제 6** <stu> 테이블을 대상으로 다음의 요구사항을 만족하도록 괄호에 적합한 명령어를 넣어 SQL문을 완성하시오. (5점)



<요구사항>

- <stu> 테이블의 '생년월일' 필드는 yyyy-mm-dd 형식이다.
- '생년월일' 필드의 값이 2004로 시작하는 레코드만 추출한다.

〈SQL문〉

```
SELECT * FROM stu WHERE 생년월일 ( ) "2004%";
```

답 :

**문제 7** 다음 〈보기〉에 제시된 내용 중에서 DBMS의 장점만을 모두 골라 기호(㉠~㉤)로 쓰시오. (5점)

5640207



〈보기〉

- ㉠ 많은 양의 데이터를 유지관리 해야 하므로 구축이 복잡하다.
- ㉡ 무결성을 유지할 수 있다.
- ㉢ 데이터를 공유할 수 있다.
- ㉣ 효율적으로 자료를 관리할 수 있다.
- ㉤ 데이터가 손상되면 복구 및 회복이 어렵다.

답 :

**문제 8** 다음 〈보기〉에 제시된 OSI 7계층을 순서대로 기호(가~사)로 쓰시오. (5점)

5640208



〈보기〉

- |            |          |              |
|------------|----------|--------------|
| 가. 네트워크 계층 | 나. 물리 계층 | 다. 데이터 링크 계층 |
| 라. 응용 계층   | 마. 세션 계층 | 바. 표현 계층     |
| 사. 전송 계층   |          |              |

답 : (    ) → (    ) → (    ) → (    ) → (    ) → (    ) → (    )

**문제 9** 다음에서 설명하는 관계형 데이터베이스 용어는 무엇인지 쓰시오. (5점)

5640209



- 릴레이션을 구성하는 각각의 행을 의미한다.
- 파일 구조에서는 레코드에 해당된다.

답 :

5640210



**문제 10** 다음은 <stu> 테이블로부터 4학년 학생들의 '학과' 필드를 검색하되, 중복된 값은 제외하고 검색하는 SQL문이다. 괄호에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

<SQL문>

SELECT (                    ) 학과 FROM stu WHERE 학년 = 4;

**답 :**

5640211



**문제 11** IP 주소는 인터넷에 연결된 모든 컴퓨터 자원을 구분하기 위한 고유한 주소이다. IP 주소는 네트워크 부분의 길이에 따라 A 클래스에서 E 클래스까지 총 5단계로 구성되는데, 이중 C 클래스의 서브넷 마스크 주소를 쓰시오. (5점)

**답 :**

5640212



**문제 12** 다음 설명에 해당하는 알맞은 용어를 <보기>에서 찾아 쓰시오. (5점)

- 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공한다.
- 헤더에 체크섬이 포함된다.
- 흐름 제어나 순서 제어가 없어 전송 속도가 빠르다.

<보기>

- TCP      • HTTP      • IPsec      • UDP      • ARP      • RARP      • SNMP      • RTCP

**답 :**

5640213



**문제 13** 다음은 Windows 운영체제의 명령 프롬프트에서 사용하는 명령어에 대한 설명이다. ①, ②의 설명에 해당하는 명령어를 쓰시오. (5점)

- ① IP 주소를 확인하기 위한 명령어이다.  
② IP 주소의 연결 상태를 확인하기 위한 명령어이다.

**답**

• ①

• ②

**문제 14** 다음 설명에 해당하는 용어를 쓰시오. (5점)

5640214



- 저속의 출력장치인 프린터를 고속의 중앙처리장치(CPU)와 병행 처리할 때, 컴퓨터 전체의 처리 효율을 높이기 위해 사용하는 기능이다.
- 인쇄할 내용을 먼저 하드디스크에 저장하고 백그라운드 작업으로 CPU의 여유 시간에 틈틈이 인쇄하기 때문에 프린터가 인쇄중이라도 다른 앱을 실행하는 포그라운드 작업이 가능하다.

**답 :****문제 15** 다음의 요구사항을 만족하도록 괄호에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

5640215



〈요구사항〉

- 〈stu1〉 테이블은 '학번', '이름', '학년', '반' 속성으로 구성된다.
- 〈stu2〉 테이블은 '학번', '국어성적', '수학성적', '영어성적', '학점', '순위' 속성으로 구성된다.
- 학년별로 '국어성적'의 평균을 계산하여 '국어평균'이라는 속성명으로 표시한다.

〈SQL문〉

```
SELECT 학년, AVG(국어성적) (      ) 국어평균
FROM stu1 a INNER JOIN stu2 b ON a.학번 = b.학번
GROUP BY 학년;
```

**답 :****문제 16** 데이터베이스 설계 과정 중 다음 설명에 해당하는 과정을 〈보기〉에서 찾아 쓰시오. (5점)

5640216



- 개념 세계의 데이터를 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계로 표현되는 구조의 데이터로 모델화한다.
- 개념 스키마를 평가 및 정제하고 DBMS에 따라 서로 다른 논리적 스키마를 설계한다.
- 트랜잭션의 인터페이스를 설계한다.

〈보기〉

- |            |             |          |
|------------|-------------|----------|
| • 요구 조건 분석 | • 개념적 설계    | • 논리적 설계 |
| • 물리적 설계   | • 데이터베이스 구현 |          |

**답 :**

5640217



**문제 17** C나 JAVA 언어에서 사용되는 서식 지정자 중에서 정수형 8진수를 출력하기 위해 사용되는 서식 지정자를 쓰시오. (5점)

답 :

5640218



**문제 18** 다음은 동일한 구조의 테이블을 추가로 생성하기 위한 SQL문이다. 괄호에 알맞은 명령어를 넣어 SQL문을 완성하시오. (5점)

(            ) TABLE dept2 SELECT \* FROM dept1;

답 :

5640219



**문제 19** 다음 설명에 해당하는 용어를 쓰시오. (5점)

구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지를 확인하기 위해 설계된 입력 값, 실행 조건, 기대 결과 등으로 구성된 테스트 항목에 대한 명세서로, 명세 기반 테스트의 설계 산출물에 해당된다.

답 :

5640220



**문제 20** 다음 설명에 해당하는 용어를 쓰시오. (5점)

- 일반적으로 CPU에 의해 처리되는 실행중인 프로그램을 의미한다.
- 비동기적 행위를 일으키는 주체이다.
- 운영체제가 관리하는 실행 단위이다.

답 :



## [문제 1]

2137490000

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값 사이에 공백을 넣어 2 1 3 7 4 9 0 0 0 0으로 쓰거나 다음과 같이

2

1

3

.

.

줄을 나눠 작성했을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

```
public class Main
{
    public static void main(String[] args)
    {
        ① int num[] = {2, 1, 3, 7, 4, 9};
        ② int numb[] = new int[10];
        ③ for (int i = 0; i < num.length; i++)
        ④     numb[i] = num[i];
        ⑤ for (int i : numb)
        ⑥     System.out.print(i);
    }
}
```

① 배열을 선언할 때 개수를 생략하고 초기값을 지정하면, 초기값의 개수로 배열의 크기가 선언된다.

	num[0]	num[1]	num[2]	num[3]	num[4]	num[5]
배열 num	2	1	3	7	4	9

② 10개의 요소를 갖는 정수형 배열 numb를 선언한다.

	num[0]	num[1]	num[2]	num[3]	num[4]	num[5]	num[6]	num[7]	num[8]	num[9]
배열 numb	0	0	0	0	0	0	0	0	0	0

※ Java는 배열 선언 시 초기값을 넣지 않으면 0으로 초기화됩니다.

③ 정수형 반복 변수 i가 0부터 1씩 증가하면서 num 배열의 길이인 6보다 작은 동안 ④번을 반복 수행한다.

· length : 배열 클래스의 속성으로, 배열 요소의 개수가 저장되어 있음. num.length는 6임

④ num[i]의 값을 numb[i]에 저장한다.

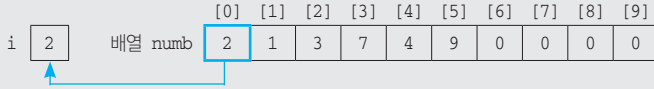
반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	num[i]	numb 배열									
		[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	2	2						0	0	0	0
1	1		1								
2	3			3							
3	7				7						
4	4					4					
5	9						9				
6											

⑤ 항상된 반복문이다. i는 numb 배열의 각 요소의 값을 차례로 받으면서 numb 배열의 요소 수만큼 ⑥번을 반복 수행한다.

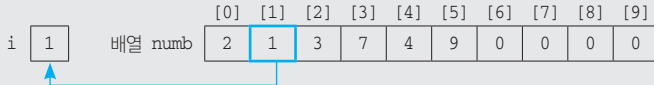
⑥ i의 값을 출력한다.

· ⑤~⑥의 첫 번째 수행: numb 배열의 첫 번째 값이 i에 저장된 후 i 값이 출력된다.



결과    2

· ⑤~⑥의 두 번째 수행: numb 배열의 두 번째 값이 i에 저장된 후 i 값이 출력된다.



결과    21

· 이런 방식으로 numb 배열의 요소 수 만큼 반복한다.

결과    2137490000

[문제 2]

17

해설

```
public class Main
{
    public static void main(String[] args)
    {
        ① int ans = 0;

        ② for (int i = 1; i < 100; i++)
        {
            ③ if (i % 3 == 0)
            ④     ans++;
            ⑤     else if (i % 4 == 0)
            ⑥         ans--;
        }
        ⑦ System.out.println(ans);
    }
}
```

① 정수형 변수 ans를 선언하고 0으로 초기화한다.

② 정수형 반복 변수 i가 1부터 1씩 증가하면서 100보다 작은 동안 ③~⑥번을 반복 수행한다.

③ i를 3으로 나눈 나머지가 0이면 ④번으로 이동하고, 아니면 ⑤번으로 이동한다.

④ 'ans = ans + 1;'과 동일하다. ans의 값을 1씩 증가시킨다.

⑤ i를 4로 나눈 나머지가 0이면 ⑥번으로 이동하고, 아니면 제어를 반복문의 처음인 ②번으로 이동한다.

⑥ 'ans = ans - 1;'과 동일하다. ans의 값을 1씩 감소시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

※ 1~99까지의 자연수 중에서 3의 배수가 나오면 1을 더하고, 4의 배수가 나오면 1을 빼는 알고리즘이다.

i	i%3==0	i%4==0	ans
1	No	No	0
2	No	No	
3	Yes		1
4	No	Yes	0
5	No	No	
6	Yes		1
7	No	No	
8	No	Yes	0
9	Yes		1
10	No	No	
11	No	No	
12	Yes		2
13	No	No	
14	No	No	
15	Yes		3
16	No	Yes	2
⋮	⋮	⋮	⋮
90	Yes		15
91	No	No	
92	No	Yes	14
93	Yes		15
94	No	No	
95	No	No	
96	Yes		16
97	No	No	
98	No	No	
99	Yes		17
100			

⑦ ans의 값을 출력한 후 커서를 다음 줄의 처음으로 이동한다.

결과 17

[문제 3]

6

## 해설

```

#include <stdio.h>
#include <stdbool.h>

⑦ bool test(int a) {
⑧     bool ans = a < 150 ? true : false;
⑨     return ans;
}
④ int fact(int i) {
⑤     if (i < 2) return 1;
        else return i*fact(i-1);
}

int main( )
{
① int ans = 0;
② for (int i = 0; i < 10; i++)
③⑥⑩ if (test(fact(i)))
⑪     ans++;
⑫ printf("%d", ans);
⑬ return 0;
}

```

모든 C 프로그램은 반드시 main() 함수에서 시작한다.

- ① 정수형 변수 ans를 선언하고 0으로 초기화한다.
- ② 정수형 반복 변수 i가 0부터 1씩 증가하면서 10보다 작은 동안 ③번을 반복 수행한다.

## 첫 번째 반복

- ③ i가 0이므로 fact(0) 함수를 호출한 결과 값을 인수로 하여 다시 test() 함수를 호출한 결과가 true(참)면 다음 줄로 이동하고 아니면 제어를 반복문의 처음인 ②번으로 이동한다.
- ④ 정수값을 반환하는 fact() 함수의 시작점이다. ③번에서 전달한 0을 i가 받는다.
- ⑤ i는 0이다. 2보다 작으므로 1을 반환하면서 제어를 fact(0)을 호출했던 ⑥번으로 옮긴다.
- ⑥ ③번에서 돌려받은 값이 1이므로 test(1) 함수를 호출한다.
- ⑦ 논리값을 반환하는 test() 함수의 시작점이다. ⑥번에서 전달한 1을 a가 받는다.
- ⑧ 논리형 변수 ans를 선언하고 a가 150보다 작으면 true(참)로, 아니면 false(거짓)로 초기화한다. a는 1이므로 ans는 true(참)로 초기화된다.
- ⑨ ans의 값 true(참)를 반환하면서 함수를 호출했던 ⑩번으로 이동한다.
- ⑩ ③번에서 돌려받은 값이 true(참)이므로 다음 줄로 이동한다.
- ⑪ ans의 값을 1 증가시킨다. ans에는 1이 저장된다.

## 두 번째 반복

- ③ i의 값이 1이 되었으므로 fact(1) 함수를 호출한다.
- ④ ③번에서 전달한 1을 i가 받는다.
- ⑤ i는 1이다. 2보다 작으므로 1을 반환하면서 제어를 fact(1)을 호출했던 ⑥번으로 옮긴다.
- ⑥ ③번에서 돌려받은 값이 1이므로 test(1) 함수를 호출한다.
- ⑦ ⑥번에서 전달한 1을 a가 받는다.
- ⑧ a가 1이므로 ans는 true(참)로 초기화된다.
- ⑨ ans의 값 true(참)를 반환하면서 함수를 호출했던 ⑩번으로 이동한다.
- ⑩ ③번에서 돌려받은 값이 true(참)이므로 다음 줄로 이동한다.
- ⑪ ans의 값을 1 증가시킨다. ans에는 2가 저장된다.

## 세 번째 반복

- ③ i의 값이 2이므로 fact(2) 함수를 호출한다.

```

4   int fact(int i) {    // i는 2입니다.
5       if (i < 2) return 1;
12      else return i*fact(i-1);
    }

```

4 ③번에서 전달한 2를 i가 받는다.

5 i는 2이다. 2보다 작지 않으므로 12번을 수행한다.

12 **return i\*fact(i-1);**

i는 2이므로 fact(1)을 호출하여 반환값을 받은 후 i값과 곱한 다음 ⑩번으로 결과를 반환한다.

```

13  int fact(int i) {    // i는 1입니다.
14      if (i < 2) return 1;
        else return i*fact(i-1);
    }

```

13 12번에서 전달한 1을 i가 받는다.

14 i는 1이다. 2보다 작으므로 1을 반환하면서 제어를 fact(1)을 호출했던 12번으로 옮긴다.

```

4   int fact(int i) {    // i는 2입니다.
5       if (i < 2) return 1;
12⑬  else return i*fact(i-1);
    }

```

⑬ **return i\*fact(i-1);**

⑭번에서 fact(1)이 1을 반환하였으므로 2를 반환하면서 제어를 fact(1)을 호출했던 ⑥번으로 옮긴다.

return i \* fact(i-1);

②            ⑥

· ②: 2 (fact(i-1)을 호출할 때 i는 2였으므로)

· ⑥: 1 (14번에서 1을 반환하였으므로)

⑥ ⑬번에서 돌려받은 값 2를 인수로 test(2) 함수를 호출한다.

7 ⑥번에서 전달한 2를 a가 받는다.

8 a가 2이므로 ans는 true(참)로 초기화된다.

9 ans의 값 true(참)를 반환하면서 함수를 호출했던 ⑩번으로 이동한다.

⑩ 9번에서 돌려받은 값이 true(참)이므로 다음 줄로 이동한다.

11 ans의 값을 1 증가시킨다. ans에는 3이 저장된다.

네 번째 반복

③ i의 값이 3이므로 fact(3) 함수를 호출한다.

```

4   int fact(int i) {    // i는 3입니다.
5       if (i < 2) return 1;
12      else return i*fact(i-1);
    }

```

4 ③번에서 전달한 3을 i가 받는다.

5 i는 3이다. 2보다 작지 않으므로 12번을 수행한다.

12 **return i\*fact(i-1);**

i는 3이므로 fact(2)을 호출하여 반환값을 받은 후 i값과 곱한 다음 ⑩번으로 결과를 반환한다.

```

13 int fact(int i) { // i는 2입니다.
14     if (i < 2) return 1;
15     else return i*fact(i-1);
16 }

```

13 12번에서 전달한 2를 i가 받는다.

14 i는 2다. 2보다 작지 않으므로 15번을 수행한다.

16 return i\*fact(i-1);

i는 2이므로 fact(1)을 호출하여 반환값을 받은 후 i값과 곱한 다음 12번으로 결과를 반환한다

```

17 int fact(int i) { // i는 1입니다.
18     if (i < 2) return 1;
19     else return i*fact(i-1);
20 }

```

17 18번에서 전달한 1을 i가 받는다.

18 i는 1이다. 2보다 작으므로 1을 반환하면서 제어를 fact(1)을 호출했던 16번으로 옮긴다.

```

13 int fact(int i) { // i는 2입니다.
14     if (i < 2) return 1;
15     else return i*fact(i-1);
16 }

```

16 return i\*fact(i-1);

13번에서 fact(1)이 1을 반환하였으므로 2를 반환하면서 제어를 fact(1)을 호출했던 19번으로 옮긴다.

return i \* fact(i-1);

㉠      ㉡

· ㉠: 2 (fact(i-1)을 호출할 때 i는 2였으므로)

· ㉡: 1 (18번에서 1을 반환하였으므로)

```

4 int fact(int i) { // i는 3입니다.
5     if (i < 2) return 1;
12 15     else return i*fact(i-1);
16 }

```

19 return i\*fact(i-1);

13번에서 fact(1)이 1을 반환하였으므로 6을 반환하면서 제어를 fact(1)을 호출했던 6번으로 옮긴다.

return i \* fact(i-1);

㉠      ㉡

· ㉠: 3 (fact(i-1)을 호출할 때 i는 3이었으므로)

· ㉡: 2 (18번에서 2를 반환하였으므로)

6 19번에서 돌려받은 값이 6이므로 test(6) 함수를 호출한다.

7 6번에서 전달한 6을 a가 받는다.

8 a가 6이므로 ans를 true(참)로 초기화한다.

9 ans의 값 true(참)를 반환하면서 함수를 호출했던 10번으로 이동한다.

10 9번에서 돌려받은 값이 true(참)이므로 다음 줄로 이동한다.

11 ans의 값을 1 증가시킨다. ans에는 4가 저장된다.

위의 과정을 통해 다음 사항들을 알 수 있다.

- `fact()` 함수는 인수가 2보다 작으면 1을 반환하고 인수가 3 이상인 경우 인수를 포함하여 그보다 작은 수의 정수들의 곱을 계산하는, 즉 팩토리얼을 계산하는 함수이다.
- `test()` 함수는 `fact()` 함수를 수행한 후 반환된 인수, 즉 계산한 팩토리얼 값이 150 미만인 경우 `true`(참)를 반환하고 150 이상인 경우 `false`(거짓)를 반환한다.
- `test()` 함수의 반환 값이 `true`(참)인 경우, 즉 계산한 팩토리얼 값이 150 미만인 경우 `ans`의 값을 1씩 증가한다.
- 결과적으로 `ans`의 값은 계산한 팩토리얼 값이 150 미만인 인수의 개수를 의미한다.
- 자세한 값의 변화는 다음 표를 통해 확인하자.

main( ) 함수		fact( ) 함수		test( ) 함수	
i	ans	i	반환값	a	반환값
0	0	0	1	1	true
1	2	1	1	1	true
2	3	2	2	2	true
3	4	3	6(1*2*3)	6	true
4	5	4	24(1*2*3*4)	24	true
5	6	5	120(1*2*3*4*5)	120	true
6		6	720(1*2*3*4*5*6)	720	false
7		7	5040(1*2*3*4*5*6*7)	5040	false
8		8	40320(1*2*3*4*5*6*7*8)	40320	false
9		9	362880(1*2*3*4*5*6*7*8*9)	362880	false
10		10	3628800(1*2*3*4*5*6*7*8*9*10)	3628800	false

㉔ `ans`의 값 6을 출력한다.

결과 6

㉔ `main()` 함수에서의 `return 0;`은 프로그램의 종료를 의미한다.

#### [문제 4]

15-5

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값 사이에 공백을 넣어 15 5나 심표를 넣어 15, 5와 같이 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

#### 해설

```
#include <stdio.h>

void intswap(int *x, int *y)
{
    int t;

    t = *x;
    *x = *y;
    *y = t;
}

void testfunc(int *a, int *b, int *c)
{
    if (*c % 2 == 0)
        intswap(a, b);
    else
        intswap(a, c);
}
```

```

int gcd(int a, int b)
{
    if (b == 0)
        return 1;
    else
        return gcd(b, a % b);
}

int main( )
{
    ❶ int a = 5;
    ❷ int b = 4;
    ❸ int test1 = 15;
    ❹ int test2 = 12;
    ❺ int test = gcd(test1, test2);

    switch( test % 2 )
    {
        case 1 :
            testfunc(&a, &b, &test2);
        default :
            testfunc(&a, &b, &test1);
    }
    printf("%d-%d", a, b);
    return 0;
}

```

- ❶ 정수형 변수 a를 선언하고 5로 초기화한다.  
 ❷ 정수형 변수 b를 선언하고 4로 초기화한다.  
 ❸ 정수형 변수 test1을 선언하고 15로 초기화한다.  
 ❹ 정수형 변수 test2를 선언하고 12로 초기화한다.  
 ❺ 정수형 변수 test를 선언하고, gcd(15, 12) 함수를 호출한 결과 값으로 초기화한다.

```

❻ int gcd(int a, int b)    // a는 15, b는 12입니다.
{
    ❼ if (b == 0)
        return 1;
    else
    ❽ return gcd(b, a%b);
}

```

- ❻ 정수값을 반환하는 gcd() 함수의 시작점이다. ❺번에서 전달한 15를 a가, 12를 b가 받는다.  
 ❼ b가 0이면 1을 반환하고 아니면 ❸번을 수행한다. b가 0이 아니므로 ❸번을 수행한다.  
 ❽ b는 12이고 a를 b로 나눈 나머지는 3이므로 gcd(12, 3)를 호출한다.

```

❻❾ int gcd(int a, int b)    // a는 12, b는 3입니다.
{
    ❼❿ if (b == 0)
        return 1;
    else
    ❽❾ return gcd(b, a % b);
}

```



- ⑨ ⑧번에서 전달한 12를 a가, 3을 b가 받는다.  
 ⑩ b가 0이 아니므로 ⑪번을 수행한다.  
 ⑪ b의 값은 3이고 a를 b로 나눈 나머지는 0이므로 gcd(3, 0)를 호출한다.

```

⑥⑨⑫ int gcd(int a, int b)    // a는 3, b는 0입니다.
{
  ⑦⑩⑬    if (b == 0)
          return 1;
          else
  ⑧⑪      return gcd(b, a % b);
}

```

- ⑫ ⑪번에서 전달한 3을 a가, 0을 b가 받는다.  
 ⑬ b가 0이므로 1을 반환하면서 제어를 gcd(3, 0)을 호출했던 ⑩번으로 옮긴다.

```

⑥⑨ int gcd(int a, int b)    // a는 12, b는 3입니다.
{
  ⑦⑩    if (b == 0)
          return 1;
          else
  ⑧⑪      return gcd(b, a % b);
}

```

- ⑪ ⑬번에서 돌려받은 값 1을 반환하면서 제어를 gcd(12, 3)을 호출했던 ⑧번으로 옮긴다.

```

⑥ int gcd(int a, int b)    // a는 15, b는 12입니다.
{
  ⑦    if (b == 0)
          return 1;
          else
  ⑧      return gcd(b, a % b);
}

```

- ⑧ ⑪번에서 돌려받은 값 1을 반환하면서 제어를 gcd(15, 12)를 호출했던 ⑥번으로 옮긴다.

```

int main( )
{
  ① int a = 5;
  ② int b = 4;
  ③ int test1 = 15;
  ④ int test2 = 12;
  ⑤ int test = gcd(test1, test2);

  ⑭ switch( test % 2 )
  {
    ⑮ case 1 :
    ⑯      testfunc(&a, &b, &test2);
      default :
          testfunc(&a, &b, &test1);
  }
  printf("%d-%d", a, b);
  return 0;
}

```

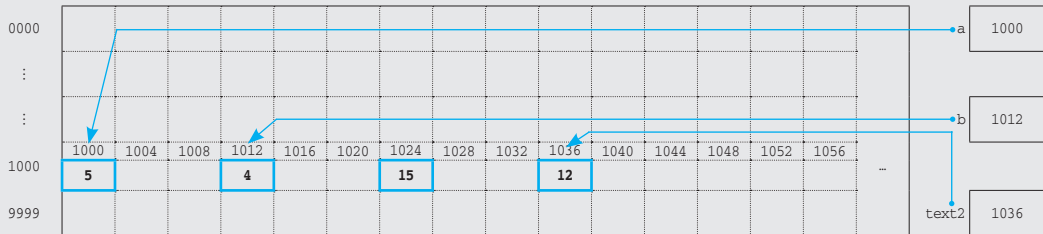
⑤ ③번에서 돌려받은 값 1로 test를 초기화한다.

⑭ test를 2로 나눈 나머지에 해당하는 숫자를 찾아간다. 현재 test는 1이므로 ⑮번으로 이동한다.

⑮ 'test % 2'가 1인 경우 찾아오는 곳이다.

⑯ a의 시작 주소, b의 시작 주소, test2의 시작 주소를 인수로 testfunc() 함수를 호출한다.

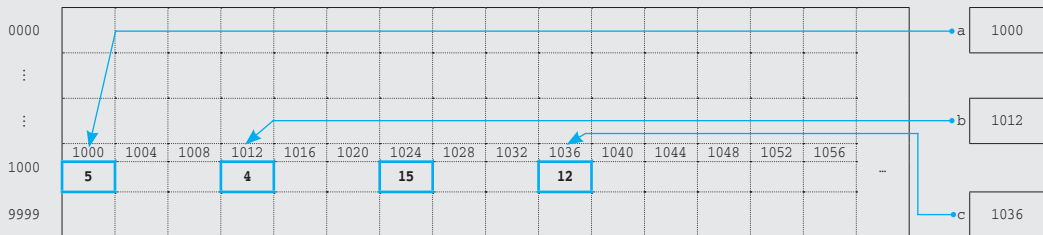
※ 여기서 지정한 주소는 임의로 정한 것이며, 이해를 돕기 위해 주소를 10진수로 표현했습니다.



```

17 void testfunc(int *a, int *b, int *c)
18 {
19     if (*c % 2 == 0)
20         intswap(a, b);
21     else
22         intswap(a, c);
23 }
    
```

⑰ 반환값이 없는 testfunc() 함수의 시작점이다. ⑯번에서 전달한 a, b, test2의 주소를 포인터 변수 a, b, c가 받는다.



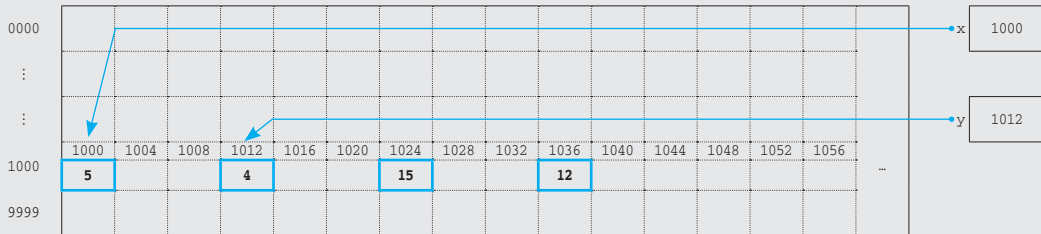
⑱ c가 가리키는 곳의 값 12를 2로 나눈 나머지가 0이면 ⑲번을 수행하고 그렇지 않으면 else 다음 문장을 수행한다. ⑲번을 수행한다.

⑲ a와 b의 주소를 인수로 intswap() 함수를 호출한다.

```

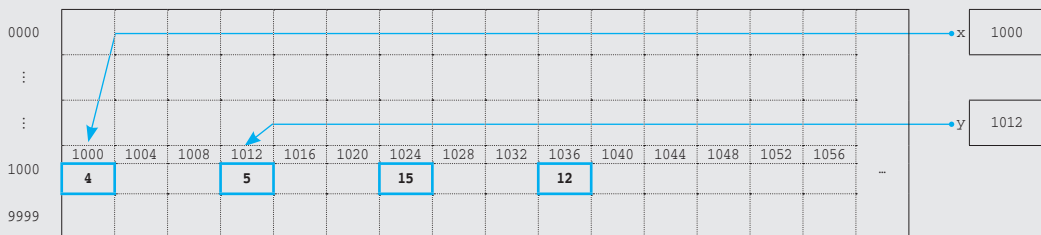
20 void intswap(int *x, int *y)
21 {
22     int t;
23
24     t = *x;
25     *x = *y;
26     *y = t;
27 }
    
```

㉑ 반환값이 없는 `intswap()` 함수의 시작점이다. ㉑번에 전달한 `a`와 `b`의 주소를 포인터 변수 `x`와 `y`가 받는다.



㉒ 정수형 변수 `t`를 선언한다.

㉓ ~ ㉑ 임시 변수 `t`를 사용하여 `x`가 가리키는 곳의 값과 `y`가 가리키는 곳의 값을 교환하는 과정이다.



※ `testfunc()` 함수를 호출할 때 인수로 `a, b, test1, test2`의 주소를 전달했으므로 `testfunc()` 함수와 `intswap()` 함수로 인한 변수들의 값 변화는 `main()` 함수의 `a, b, test1, test2` 변수도 공유하게 됩니다.

㉑ 함수를 마치고 `intswap()` 함수를 호출했던 ㉑번으로 돌아간다.

```

17 void testfunc(int *a, int *b, int *c)
18 {
19     if (*c % 2 == 0)
20         intswap(a, b);
21     else
22         intswap(a, c);
23 }

```

㉒ `intswap()` 함수를 호출한 후 돌아왔고, `if`문이 종료되었으므로 함수를 마치고 `testfunc()` 함수를 호출했던 ㉒번으로 돌아간다.

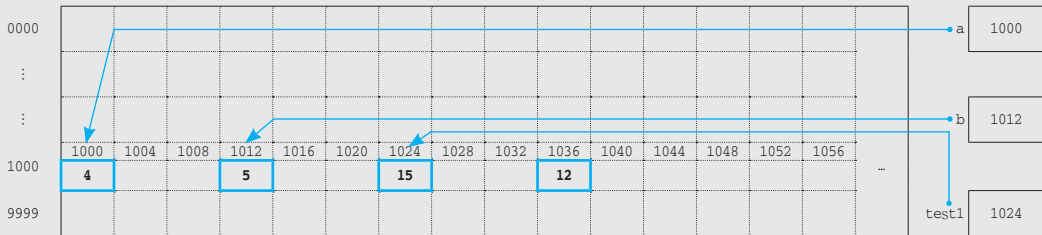
```

int main( )
{
1  int a = 5;
2  int b = 4;
3  int test1 = 15;
4  int test2 = 12;
5  int test = gcd(test1, test2);

14 switch( test % 2 )
15 {
16     case 1 :
17         testfunc(&a, &b, &test2);
18     default :
19         testfunc(&a, &b, &test1);
20 }
    printf("%d-%d", a, b);
    return 0;
}

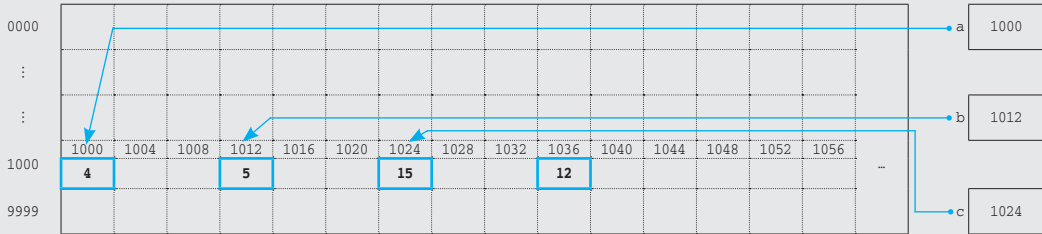
```

- ㉗ case문에 break문이 생략되었으므로 이후 switch문이 종료될 때까지 모든 문장을 실행한다. 제어를 ㉘번으로 옮긴다.  
㉘ 'default'는 ㉗번의 식의 결과에 해당하는 case가 없는 경우 찾아오는 곳이지만 앞선 case문에 break문이 생략되었으므로 제어가 옮겨온 것이다.  
㉘번을 수행한다.  
㉙ a의 시작 주소, b의 시작 주소, test1의 시작 주소를 인수로 testfunc() 함수를 호출한다.  
※ 여기서 지정한 주소는 임의로 정한 것이며, 이해를 돕기 위해 주소를 10진수로 표현했습니다.



```
㉙ void testfunc(int *a, int *b, int *c)
{
㉚   if (*c % 2 == 0)
       intswap(a, b);
㉛   else
       intswap(a, c);
}
```

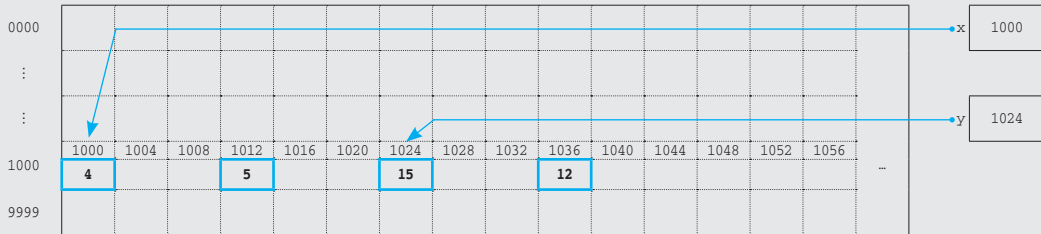
- ㉙ 반환값이 없는 testfunc() 함수의 시작점이다. ㉚번에서 전달한 a, b, test1의 주소를 포인터 변수 a, b, c가 받는다.



- ㉛ c가 가리키는 곳의 값 15를 2로 나눈 나머지가 0이면 다음 문장을 수행하고 그렇지 않으면 ㉚번을 수행한다. ㉚번을 수행한다.  
㉚ a와 c의 주소를 인수로 intswap() 함수를 호출한다.

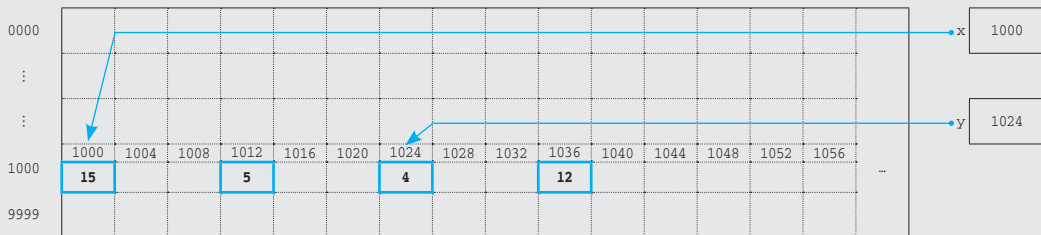
```
㉛ void intswap(int *x, int *y)
{
㉜   int t;
㉝   t = *x;
㉞   *x = *y;
㉟   *y = t;
㊱ }
```

㉔ 반환값이 없는 intswap() 함수의 시작점이다. ㉕번에 전달한 a와 c의 주소를 포인터 변수 x와 y가 받는다.



㉖ 정수형 변수 t를 선언한다.

㉗ ~ ㉙ 임시 변수 t를 사용하여 x가 가리키는 곳의 값과 y가 가리키는 곳의 값을 교환하는 과정이다.



※ testfunc() 함수를 호출할 때 인수로 a, b, test1, test2의 주소를 전달했으므로, testfunc() 함수와 intswap() 함수로 인한 변수들의 값 변화는 main() 함수의 a, b, test1, test2 변수도 공유하게 됩니다.

㉚ 함수를 마치고 intswap() 함수를 호출했던 ㉕번으로 돌아간다.

```

㉔ void testfunc(int *a, int *b, int *c)
{
㉕     if (*c % 2 == 0)
        intswap(a, b);
    else
㉖         intswap(a, c);
}

```

㉗ intswap() 함수를 호출한 후 돌아왔고, if문이 종료되었으므로 함수를 마치고 testfunc() 함수를 호출했던 ㉕번으로 돌아간다.

```

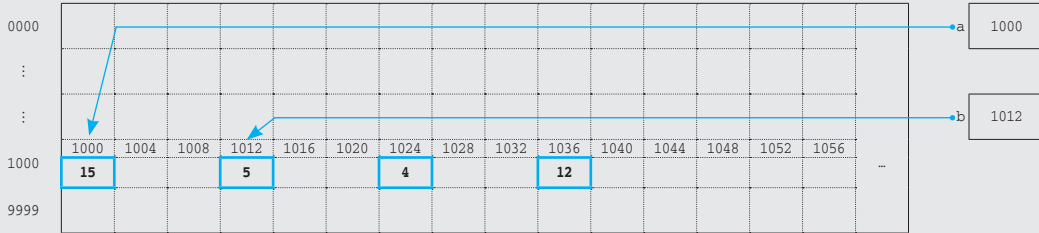
int main( )
{
㉗     int a = 5;
㉘     int b = 4;
㉙     int test1 = 15;
㉚     int test2 = 12;
㉛     int test = gcd(test1, test2);

㉜     switch( test % 2 )
    {
㉝         case 1 :
㉞㉟         testfunc(&a, &b, &test2);
㊱         default :
㊲             testfunc(&a, &b, &test1);
    }
㊳     printf("%d-%d", a, b);
㊴     return 0;
}

```

- ㉓ testfunc() 함수를 호출한 후 돌아왔고, switch문이 종료되었으므로 제어를 ㉒번으로 옮긴다.  
㉒ a의 값 15를 출력하고 -을 출력한 후 b의 값 5를 출력한다.

결과 15-5



※ testfunc() 함수를 호출할 때 인수로 a, b, test1, test2의 주소를 전달했으므로, testfunc() 함수와 intswap() 함수로 인한 변수들의 값 변화는 main() 함수의 a, b, test1, test2 변수도 공유하게 됩니다.

- ㉔ main() 함수에서의 return 0;은 프로그램의 종료를 의미한다.

[문제 5]

5

해설

```
import java.lang.Math;           // pow() 메소드가 정의되어 있는 헤더 파일이다.
public class Main
{
    public static void main(String[] args)
    {
        ① int s, t, cnt = 0;
        ② int test[] = new int[5];
        ③ for (int i = 0; i < 10; i++)
        {
            ④ s = (int)Math.pow(i, 2);
            ⑤ t = (s % 10);

            ⑥ try {
            ⑦     test[t] = i;
            }
            ⑧ catch(ArrayIndexOutOfBoundsException e) {
            ⑨     cnt++;
            }
        }
        ⑩ System.out.println(cnt);
    }
}
```

- ① 정수형 변수 s, t, cnt를 선언하고, cnt를 0으로 초기화한다.  
② 5개의 요소를 갖는 정수형 배열 test를 선언한다.

	[0]	[1]	[2]	[3]	[4]
배열 test	0	0	0	0	0

※ Java는 배열 선언 시 초기값을 지정하지 않으면 배열의 모든 요소가 0으로 초기화됩니다.

③ 반복 변수  $i$ 가 0부터 1씩 증가하면서 10보다 작은 동안 ④~⑨번을 반복 수행한다.

④ `(int)Math.pow(i, 2)`의 결과를  $s$ 에 저장한다.

· **Math**: 수학 관련 메소드들이 포함되어 있는 클래스

· **pow(A, B)**: A의 B승을 구하는 함수

⑤  $(s \% 10)$ 의 결과를  $t$ 에 저장한다.

⑥ 예외 구문의 시작이다.

⑦ `test[t]`에  $i$ 의 값을 저장한다.

⑧ `ArrayIndexOutOfBoundsException`에 해당하는 예외를 다루는 `catch`문의 시작이다.

· **ArrayIndexOutOfBoundsException**: 참조하는 배열의 위치값이 배열의 크기를 벗어나거나 음수인 경우 사용하는 예외 객체

· ⑦에서  $t$ 가 4보다 크면 실행된다.

⑨ `'cnt = cnt + 1;`과 동일하다.  $cnt$ 의 값을 1씩 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	s	t	예외발생	cnt	배열 test
0	0	0	No		
1	1	1	No		
2	4	4	No		
3	9	9	Yes	1	
4	16	6	Yes	2	
5	25	5	Yes	3	
6	36	6	Yes	4	
7	49	9	Yes	5	
8	64	4	No		
9	81	1	No		
10					

[0]	[1]	[2]	[3]	[4]
0	9	0	0	8

⑩  $cnt$ 의 값 5를 출력한다.

결과 5

#### [문제 6]

LIKE

※ 답안 작성 시 주의 사항: SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

#### 해설

```
SELECT *
FROM stu
WHERE 생년월일 "LIKE 2004%";
```

모든 필드를 표시한다.

<stu> 테이블을 대상으로 검색한다.

'생년월일' 필드의 값이 2004로 시작하는 자료만을 대상으로 한다.

#### [문제 7]

㉠, ㉡, ㉢

#### [문제 8]

나 → 다 → 가 → 사 → 마 → 바 → 라

#### [문제 9]

※ 다음 중 하나를 쓰면 됩니다.

튜플, Tuple

[문제 10]

DISTINCT

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대 · 소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

해설

```
SELECT DISTINCT 학과
FROM stu
WHERE 학년 = 4;
```

'학과' 필드를 표시하되, 같은 '학과' 필드의 값은 한 번만 표시한다.  
<stu> 테이블을 대상으로 검색한다.  
'학년' 필드의 값이 4인 자료만을 대상으로 한다.

[문제 11]

255,255,255.0

[문제 12]

UDP

[문제 13]

① IPCONFIG    ② PING

[문제 14]

※ 다음 중 하나를 쓰면 됩니다.

스풀링, Spooling

[문제 15]

AS

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대 · 소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

해설

```
SELECT 학년, AVG(국어성적) AS 국어평균
FROM stu1 a INNER JOIN stu2 b
ON a.학번 = b.학번
GROUP BY 학년;
```

'학년'과 '국어성적'의 평균을 표시하되, '국어성적'의 평균은 '국어평균'으로 표시한다.  
<a>라는 별칭의 <stu1> 테이블과 <b>라는 별칭의 <stu2> 테이블을 대상으로 검색한다.  
<stu1> 테이블의 '학번'과 <stu2> 테이블의 '학번'을 기준으로 서로 JOIN한다.  
'학년'을 기준으로 그룹을 지정한다.

[문제 16]

논리적 설계

[문제 17]

%o



## [문제 18]

CREATE

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대 · 소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
CREATE TABLE dept2
SELECT * FROM dept1;
```

테이블 &lt;dept2&gt;를 생성한다.

&lt;dept1&gt; 테이블의 모든 필드, 즉 &lt;dept1&gt; 테이블과 동일한 구조를 적용한다.

## [문제 19]

※ 다음 중 하나를 쓰면 됩니다.

테스트 케이스, Test Case

## [문제 20]

※ 다음 중 하나를 쓰면 됩니다.

프로세스, Process



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

**문제 1** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

5640101



```
#include <stdio.h>
int main( )
{
    int x = 5, y = 10, z = 20, sum;
    x += y;
    y -= x;
    z %= y;
    sum = x + y + z;
    printf("%d", sum);
    return 0;
}
```

답 :

**문제 2** 다음 설명에 해당하는 알맞은 네트워크 토폴로지를 쓰시오. (5점)

5640102



모든 지점의 장치가 다른 지점의 장치와 서로 포인트 투 포인트(Point To Point)로 연결된 형태로, 노드의 연결성이 높다. 많은 장치로부터 많은 양의 통신을 필요로 하는 경우 유리하다.

답 :

**문제 3** 데이터베이스 용어에 대한 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 <보기>에서 찾아 쓰시오. (5점)

5640103



- ( ① ) : 개체를 묘사하는 데 사용될 수 있는 특성을 의미한다.
- ( ② ) : 개체 간 또는 속성 간의 논리적인 연결을 의미한다.

<보기>

• 속성      • 기수      • 차수      • 도메인      • 개체      • 관계

답

• ①

• ②

5640104



**문제 4** 다음은 <player> 테이블의 age 속성을 제거하면서 연계된 모든 제약 조건도 함께 제거하는 SQL 문이다. 괄호에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

<SQL문>

ALTER TABLE player DROP COLUMN age (        );

**답 :**

5640105



**문제 5** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>
int main( )
{
    int ans;
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            if (i==j || i+j==4) ans++;

    printf("%d", ans);
    return 0;
}
```

**답 :**

5640106



**문제 6** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>
#include <stdbool.h>

bool test(int a)
{
    if (a % 2 == 1) return true;
    else return false;
}

int main( )
{
    int res = 0;
    int tmp;
    for (int i = 0 ; i < 6; i++)
        for (int j = 0 ; j < 6; j++)
        {
            if (test(i+j)) tmp = i;
            else tmp = 0;
            res += tmp;
        }
    printf("%d", res);
    return 0;
}
```

**답 :**



**문제 7** 다음 JAVA로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
class Pos
{
    private int x = 0, y = 0;
    Pos(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
    public int sum( ) { return x + y; }
    public int sum(int x) { return x + y; }
}

public class Main
{
    public static void main(String[] args) {
        Pos[] list = new Pos[2];
        list[0] = new Pos(1, 1);
        list[1] = new Pos(2, 2);
        int res = 0;
        for (int i = 0; i < list.length; i++)
            res += list[i].sum(i);
        System.out.print(res);
    }
}
```

**답 :**

**문제 8** 다음 JAVA로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

5640108



```
Public class Main
{
    public static void main(String[] args) {
        String teststr = "        LoremIpsum        ";
        int a = teststr.trim( ).length( );

        System.out.print(a);
    }
}
```

답 :

**문제 9** 다음 설명에 해당하는 알맞은 용어를 <보기>에서 찾아 쓰시오. (5점)

5640109



단말장치 사용자가 일정한 시간 간격(Time Slice) 동안 CPU를 사용함으로써 단독으로 CPU를 사용하는 것과 같은 효과를 가지는 시스템으로, 라운드 로빈(Round Robin) 방식이라고도 한다. 한 시스템을 여러 명의 사용자가 공유하여 동시에 작업을 수행하며, 시스템의 전체 효율은 좋아지나 개인별 사용자 입장에서는 반응 속도가 느려질 수 있다.

<보기>

- |             |                |             |
|-------------|----------------|-------------|
| • 일괄 처리 시스템 | • 다중 프로그래밍 시스템 | • 시분할 시스템   |
| • 다중 처리 시스템 | • 실시간 처리 시스템   | • 분산 처리 시스템 |

답 :

**문제 10** 인터넷에 연결된 모든 컴퓨터의 자원을 구분하기 위한 고유한 주소인 IPv4의 크기는 몇 비트인지 쓰시오. (5점)

5640110



답 :

5640111



**문제 11** UNIX 명령어 중 영문 List의 약어로 현재 작업중인 디렉터리의 모든 파일을 표시하는 명령어가 무엇인지 <보기>에서 찾아 쓰시오. (5점)

<보기>

• pwd      • ls      • cd      • rm      • chown      • cat

**답 :**

5640112



**문제 12** 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 DDL(Data Define Language) 명령어를 <보기>에서 찾아 쓰시오. (5점)

- ( ① ) : TABLE에 대한 정의를 변경하는 데 사용함
- ( ② ) : SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의함

<보기>

• CREATE      • DROP      • UPDATE      • DELETE      • ALTER      • SELECT

**답**

- ①
- ②

5640113



**문제 13** 다음 처리조건을 참고하여 인덱스를 생성하려고 한다. 괄호에 적합한 예약어를 넣어 SQL 문을 완성하시오. (5점)

<처리조건>

- <회원> 테이블의 '번호' 속성을 기준으로 오름차순 정렬한다.
- UNIQUE나 CLUSTER 옵션은 생략할 수 있다.

<SQL문>

CREATE INDEX idx\_회원 (                      ) 회원(번호 ASC);

**답 :**



**문제 14** 다음 설명에 해당하는 키(KEY)를 <보기>에서 찾아 기호(㉠~㉤)로 쓰시오. (5점)

5640114



- 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키를 말한다.
- 릴레이션을 구성하는 모든 튜플 중 이 속성의 집합과 동일한 값은 나타나지 않는다.
- 릴레이션을 구성하는 모든 튜플에 대해 유일성(Unique)은 만족하지만, 최소성(Minimality)은 만족하지 못한다.

<보기>

㉠ 기본키      ㉡ 후보키      ㉢ 슈퍼키      ㉣ 외래키      ㉤ 대체키

**답 :**

**문제 15** 다음에서 설명하는 내용에 해당하는 테스트 기법을 쓰시오. (5점)

5640115



- 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 방법이다.
- 개발한 소프트웨어를 사용자가 직접 테스트한다.
- 사용자에게 전달되기 전에 마지막으로 수행하는 테스트로, 알파 테스트, 베타 테스트 등이 있다.

**답 :**

**문제 16** 다음 설명에 해당하는 용어를 <보기>에서 찾아 쓰시오. (5점)

5640116



그래프를 이용해서 데이터 논리 구조를 표현한 데이터 모델로, 상위(Owner)와 하위(Member) 레코드 사이에서 다 대 다(N:M) 대응 관계를 만족하는 구조이며, 레코드 타입 간의 관계는 1:1, 1:N, N:M이 될 수 있는 DBMS이다.

<보기>

• RDBMS      • NDBMS      • HDBMS      • ODBMS

**답 :**

5640117



**문제 17** 다음 처리조건을 참고하여 자료를 추출하려고 한다. 괄호에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

〈처리조건〉

- 〈대출관리〉 테이블은 '대출코드', '대출일자', '도서코드', '회원코드' 필드로 구성된다.
- 하루에 2회 이상 대출한 회원의 정보를 출력한다.

〈SQL문〉

```
SELECT 대출일자, 회원코드 FROM 대출관리
GROUP BY 대출일자, 회원코드 ( ) COUNT(*) > 1;
```

**답 :**

5640118



**문제 18** 다음은 TCP와 UDP에 대한 설명이다. 괄호(①, ②)에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

( ① )는 양방향 연결형 서비스를 제공하며, 스트림 위주의 전달을 한다. ( ① )는 신뢰성 있는 경로를 확립하고 메시지 전송을 감독한다. ( ② )는 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공한다. ( ② )는 실시간 전송에 유리하며, 신뢰성보다는 속도가 중요시되는 네트워크에서 사용된다.

**답**

- ①
- ②

5640119



**문제 19** 다음 설명에 해당하는 용어를 〈보기〉에서 찾아 쓰시오. ( 5점)

메소드(Method)의 이름은 같지만 인수를 받는 자료형과 개수를 달리하여 여러 기능을 정의할 수 있다.

〈보기〉

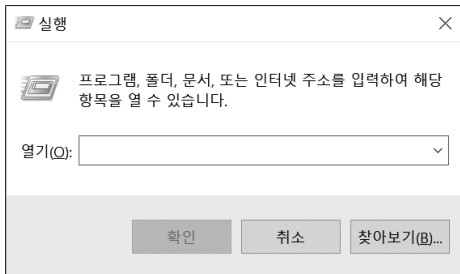
- 추상화
- 오버로딩
- 오버라이딩
- 상속
- 캡슐화

**답 :**

5640120



**문제 20** 다음과 같은 실행 대화상자를 표시하는 단축키를 <보기>에서 찾아 쓰시오. (단, 운영체제는 Windows 10이라고 가정한다.) (5점)



<보기>

• E	• R	• TAB	• SHIFT+M	• ALT+D
-----	-----	-------	-----------	---------

**답 :** Windows Key + (     )

## [문제 1]

10

## 해설

```
#include <stdio.h>
int main( )
{
    ① int x = 5, y = 10, z = 20, sum;
    ② x += y;
    ③ y -= x;
    ④ z %= y;
    ⑤ sum = x + y + z;
    ⑥ printf("%d", sum);
    ⑦ return 0;
}
```

- ① 정수형 변수 x, y, z, sum을 선언하고 x를 5, y를 10, z를 20으로 초기화한다.
- ② 'x = x + y;'와 동일하다. 'x + y'의 결과 값을 x에 저장한다. x에는 15가 저장된다.
- ③ 'y = y - x;'와 동일하다. 'y - x'의 결과 값을 y에 저장한다. y에는 -5가 저장된다.
- ④ 'z = z % y;'와 동일하다. 'z % y'의 결과 값을 z에 저장한다. z에는 0이 저장된다.
- ⑤ 'x + y + z'의 결과 값을 sum에 저장한다. sum에는 10이 저장된다.
- ⑥ sum의 값 10을 정수로 출력한다.

결과 10

- ⑦ main() 함수에서의 **return 0;**은 프로그램의 종료를 의미한다.

## [문제 2]

※ 다음 중 하나를 쓰면 됩니다.

망형, 그물형, Mesh

## [문제 3]

- ① 속성    ② 관계

## [문제 4]

CASCADE

※ **답안 작성 시 주의 사항** : SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
ALTER TABLE player
DROP COLUMN age CASCADE;
```

〈player〉 테이블에 대한 정의를 변경한다.  
age를 삭제하되, age 속성에 포함된 제약 조건도 함께 삭제한다.

## [문제 5]

9

## 해설

```
#include <stdio.h>
int main( )
{
  ❶ int ans;
  ❷ for (int i = 0; i < 5; i++)
  ❸     for (int j = 0; j < 5; j++)
  ❹         if (i==j || i+j==4) ans++;

  ❺ printf("%d", ans);
  ❻ return 0;
}
```

- ❶ 정수형 변수 `ans`를 선언한다.  
 ❷ 반복 변수 `i`가 0부터 1씩 증가하면서 5보다 작은 동안 ❸, ❹번을 반복 수행한다.  
 ❸ 반복 변수 `j`가 0부터 1씩 증가하면서 5보다 작은 동안 ❹번을 반복 수행한다.  
 ❹ `i`가 `j`와 같거나 `i+j`가 4면, `ans`의 값을 1씩 증가시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	j	i==j	i+j==4	ans
0	0	Yes	No	1
	1	No	No	
	2	No	No	
	3	No	No	
	4	No	Yes	
1	0	No	No	3
	1	Yes	No	
	2	No	No	
	3	No	Yes	
	4	No	No	
2	0	No	No	5
	1	No	No	
	2	Yes	Yes	
	3	No	No	
	4	No	No	
3	0	No	No	6
	1	No	Yes	
	2	No	No	
	3	Yes	No	
	4	No	No	
4	0	No	Yes	8
	1	No	No	
	2	No	No	
	3	No	No	
	4	Yes	No	
5				

- ❺ `ans`의 값 9를 정수로 출력한다.

결과 9

- ❻ `main()` 함수에서의 `return 0`은 프로그램의 종료를 의미한다.

## [문제 6]

45

## 해설

```

#include <stdio.h>
#include <stdbool.h>

⑥ bool test(int a)
{
  ⑦ if (a % 2 == 1) return true;
    else return false;
}

int main( )
{
  ① int res = 0;
  ② int tmp;
  ③ for (int i = 0 ; i < 6; i++)
  ④   for (int j = 0 ; j < 6; j++)
    {
      ⑤⑧   if (test(i+j)) tmp = i;
            else tmp = 0;
      ⑨   res += tmp;
    }
  ⑩ printf("%d", res);
  ⑪ return 0;
}

```

모든 C 프로그램은 반드시 main() 함수에서 시작한다.

- ① 정수형 변수 res를 선언하고 0으로 초기화한다.
- ② 정수형 변수 tmp를 선언한다.
- ③ 반복 변수 i가 0부터 1씩 증가하면서 6보다 작은 동안 ④~⑨번을 반복 수행한다.
- ④ 반복 변수 j가 0부터 1씩 증가하면서 6보다 작은 동안 ⑤~⑨번을 반복 수행한다.

## 반복 과정

- ⑤ i+j의 값 0을 인수로 하여 test() 함수를 호출한 결과가 true(참)면 tmp에 i를 저장하고, false(거짓)면 tmp에 0을 저장한다.
- ⑥ 논리값을 반환하는 test() 함수의 시작점이다. ⑤번에서 전달받은 0을 a가 받는다.
- ⑦ a의 값 0을 2로 나눈 나머지는 0이다. 조건을 만족하지 않으므로 논리값 false(거짓)를 반환하면서 ⑧번으로 이동한다.
- ⑧ ⑦번에서 돌려받은 값이 false(거짓)이므로 tmp에 0을 저장한다.
- ⑨ 'res = res + tmp'와 동일하다. res에 tmp를 누적한다. res에는 0이 저장된다.

위의 반복 과정을 통해 다음 사항들을 알 수 있다.

- test() 함수는 a, 즉 main() 함수에서 전달받은 i+j의 값을 2로 나눈 나머지가 1인 경우 true를 반환한다.
- test() 함수에서 반환된 값이 true이면 tmp에 i 값을 저장한다.
- res에 tmp의 값을 누적한다.
- 자세한 값의 변화는 다음 표를 통해 확인하자.

main()				test( )		
i	j	res	tmp	a	a%2	반환값
0	0	0	0	0	0	false
	1	0	0	1	1	true
	2	0	0	2	0	false
	3	0	0	3	1	true
	4	0	0	4	0	false
	5	0	0	5	1	true
	6	0	0			
1	0	1	1	1	1	true
	1		0	2	0	false
	2	2	1	3	1	true
	3		0	4	0	false
	4	3	1	5	1	true
	5		0	6	0	false
	6					
2	0	3	0	2	0	false
	1	5	2	3	1	true
	2		0	4	0	false
	3	7	2	5	1	true
	4		0	6	0	false
	5	9	2	7	1	true
	6					
3	0	12	3	3	1	true
	1		0	4	0	false
	2	15	3	5	1	true
	3		0	6	0	false
	4	18	3	7	1	true
	5		0	8	0	false
	6					
4	0	18	0	4	0	false
	1	22	4	5	1	true
	2		0	6	0	false
	3	26	4	7	1	true
	4		0	8	0	false
	5	30	4	9	1	true
	6					
5	0	35	5	5	1	true
	1		0	6	0	false
	2	40	5	7	1	true
	3		0	8	0	false
	4	45	5	9	1	true
	5		0	10	0	false
	6					
6						

⑩ res의 값 45를 정수로 출력한다.

결과 45

⑪ main( ) 함수에서의 return 0;은 프로그램의 종료를 의미한다.

[문제 7]

4

해설

```
class Pos // 클래스 Pos를 정의한다.
{
  ① private int x = 0, y = 0;
  ③ ⑧ Pos(int x, int y)
  {
    ④ ⑨ this.x = x;
    ⑤ ⑩ this.y = y;
    ⑥ ⑪ }
  ⑧ public int sum( ) { return x + y; }
  ④ ⑤ ⑦ public int sum(int x) { return x + y; }
}

public class Main
{
  public static void main(String[] args) {
    ① Pos[] list = new Pos[2];
    ② list[0] = new Pos(1, 1);
    ⑦ list[1] = new Pos(2, 2);
    ⑫ int res = 0;
    ⑬ for (int i = 0; i < list.length; i++)
    ⑭ ⑮ ⑯ res += list[i].sum(i);
    ⑰ System.out.print(res);
  }
}
```

- ① 정수형 변수 x, y를 선언하고, 각각을 0으로 초기화한다.
- ⑤ 정수값을 반환하는 메소드 sum( )을 정의한다.
- ⑦ 정수값을 반환하는 메소드 sum(int x)를 정의한다. 메소드 이름이 ⑤와 같지만 인수의 자료형이 다르므로 서로 다른 메소드이다. 이렇게 이름은 같지만 자료형을 달리하여 다른 기능을 정의하는 것을 오버로딩(Overloading)이라고 한다.

모든 Java 프로그램은 반드시 main( ) 메소드에서 시작한다.

① Pos 클래스 자료형의 2개 짜리 list 배열을 선언한다.

list 배열	class Pos				
	x	y	Pos ( )		sum(int x)
			x	y	x
list[0]					
list[1]					

- ② (1, 1)을 인수로 Pos 클래스의 생성자를 호출하여 처리한 값을 list[0]에 저장한다.
- ③ Pos 클래스 생성자의 시작점이다. ②번에서 전달받은 1과 1을 x와 y가 받는다.

list 배열	class Pos				
	x	y	Pos ( )		sum(int x)
			x	y	x
list[0]			1	1	



④ this.x에 x의 값 1을 저장한다.

※ **this** : 현재 실행중인 메소드가 속한 클래스를 가리키는 예약어로, 현재 실행중인 Pos() 메소드가 속한 Pos 클래스를 가리킴

⑤ this.y에 y의 값 1을 저장한다.

list 배열	class Pos				
	x	y	Pos ( )		sum(int x)
			x	y	x
list[0]	1	1	1	1	

⑥ 생성자가 종료되었으므로 생성자를 호출했던 ②번의 다음 줄인 ⑦번으로 이동한다.

⑦ (2, 2)를 인수로 Pos 클래스의 생성자를 호출하여 처리한 값을 list[1]에 저장한다.

⑧ Pos 클래스 생성자의 시작점이다. ⑦번에서 전달받은 2와 2를 x와 y가 받는다.

list 배열	class Pos				
	x	y	Pos ( )		sum(int x)
			x	y	x
list[0]	1	1			
list[1]			2	2	

⑨ this.x에 x의 값 2를 저장한다.

⑩ this.y에 y의 값 2를 저장한다.

list 배열	class Pos				
	x	y	Pos ( )		sum(int x)
			x	y	x
list[0]	1	1			
list[1]	2	2			

⑪ 생성자가 종료되었으므로 생성자를 호출했던 ⑦번의 다음 줄인 ⑫번으로 이동한다.

⑫ 정수형 변수 res를 선언하고 0으로 초기화한다.

⑬ 반복 변수 i가 0부터 1씩 증가하면서 list 배열의 길이인 2보다 작은 동안 ⑭번을 반복 수행한다.

⑭ list[0].sum(0) 메소드를 호출한 후 반환된 값을 res에 누적한다. sum(0) 메소드의 인수가 정수이므로 ⑮번으로 이동한다.

⑮ 정수를 반환하는 sum( ) 메소드의 시작점이다. ⑭번에서 인수가 하나만 전달됐으므로, ⑭번에서 전달받은 0을 x가 받는다. 이어서 x+y의 결과 값 1을 메소드를 호출했던 ⑮번으로 반환한다.

※ x+y에서 x는 sum( ) 메소드가 호출될 때 전달받은 0이 사용되고, y는 전달받은 인수가 없으므로 현재 sum( ) 메소드가 속한 list[0] 배열의 y, 즉 1이 사용되어 x+y의 결과는 1입니다.

list 배열	class Pos					res
	x	y	Pos ( )		sum(int x)	
			x	y	x	
list[0]	1	1			0	
list[1]	2	2				1

⑯ ⑮번에서 반환받은 1을 res에 누적한다. 이어서 반복문에 의해 i 값이 1 증가된다.

list[1].sum(1) 메소드를 호출한 후 반환된 값을 res에 누적한다. sum(1) 메소드의 인수가 정수이므로 ⑰번으로 이동한다.

⑰ 정수를 반환하는 sum( ) 메소드의 시작점이다. ⑯번에서 인수가 하나만 전달됐으므로, ⑯번에서 전달받은 1을 x가 받는다. 이어서 x+y의 결과 값 3을 메소드를 호출했던 ⑰번으로 반환한다.

※ x+y에서 x는 sum( ) 메소드가 호출될 때 전달받은 1이 사용되고, y는 전달받은 인수가 없으므로 현재 sum( ) 메소드가 속한 list[0] 배열의 y, 즉 2가 사용되어 x+y의 결과는 3입니다.

list 배열	class Pos					res
	x	y	Pos ( )		sum(int x)	
			x	y	x	
list[0]	1	1			0	
list[1]	2	2			1	4

- ⑮ ⑮번에서 반환받은 3을 res에 누적한다. 이어서 반복문에 의해 i 값이 1 증가된다.  
증가된 반복 변수 i의 값 2는 list 배열의 길이인 2보다 작지 않으므로 반복문을 벗어나 제어를 ⑮번으로 옮긴다.  
⑮ res의 값 4를 출력한다.

결과 4

[문제 8]

10

해설

```
public class Main
{
    public static void main(String[] args) {
        ① String teststr = "        LoremIpsum        ";
        ② int a = teststr.trim( ).length( );

        ③ System.out.print(a);
    }
}
```

- ① 문자열 변수 teststr을 선언하고 " LoremIpsum "으로 초기화한다.  
② 정수형 변수 a를 선언하고, teststr 변수에 저장된 문자열에서 좌우 공백을 제거하고 남은 문자열의 길이로 초기화한다.  
· teststr 변수에 저장된 문자열에서 좌우 공백을 제거하면 "LoremIpsum"만 남는다.  
· teststr 변수에 저장된 값은 "LoremIpsum"의 길이인 10이다.  
· trim(): 변수에 저장된 문자열의 좌우 공백을 제거하는 메소드  
· length(): 변수에 저장된 문자열의 길이를 반환하는 메소드  
③ a의 값 10을 출력한다.

결과 10

[문제 9]

시분할 시스템

[문제 10]

32

[문제 11]

ls

[문제 12]

- ① ALTER    ② CREATE

[문제 13]

ON

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대 · 소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
CREATE INDEX idx_회원
ON 회원 (번호 ASC);
```

인덱스 <idx\_회원>을 생성한다.  
<회원> 테이블의 '번호' 속성을 이용하여 생성한다. '번호'는 오름차순으로 정렬한다.

## [문제 14]

㉔

## [문제 15]

※ 다음 중 하나를 쓰면 됩니다.

인수 테스트, Acceptance Test

## [문제 16]

NDBMS

## [문제 17]

HAVING

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
SELECT 대출일자, 회원코드
FROM 대출관리
GROUP BY 대출일자, 회원코드
HAVING COUNT(*) > 1;
```

'대출일자', '회원코드'를 표시한다.  
<대출관리> 테이블에서 검색한다.  
'대출일자'별 '회원코드'별로 그룹을 지정한다.  
'대출일자'별 '회원코드'의 개수가 1을 초과하는 자료만을 대상으로 한다.

## [문제 18]

① TCP    ② UDP

## [문제 19]

오버로딩

## [문제 20]

R



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

**문제 1** 데이터베이스 용어에 대한 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 <보기>에서 찾아 기호(㉠~㉣)로 쓰시오. (5점)

4640401



- 릴레이션의 행은 ( ① )으로 구성된다.
- ( ① )은 데이터베이스를 구성하는 가장 작은 논리적 단위이며, 데이터 항목 또는 데이터 필드에 해당한다.
- ( ① )의 개수를 ( ② )라고 한다.

<보기>

- |                 |                   |               |
|-----------------|-------------------|---------------|
| ㉠ 속성(Attribute) | ㉡ 기수(Cardinality) | ㉢ 차수(Degree)  |
| ㉣ 도메인(Domain)   | ㉤ 레코드(Record)     | ㉥ 스키마(Schema) |

**답**

- ①
- ②

**문제 2** <의원> 테이블을 정의하는 SQL문이다. 아래의 <요구사항>을 만족하도록 괄호에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

4640402



<요구사항>

- 'id(문자 5)', 'name(문자 10)', 'sex(문자 1)', 'phone(문자 20)' 속성을 가진다.
- 'id' 속성은 기본키이다.
- 'sex' 속성은 'f' 또는 'm' 값만 갖도록 한다(제약조건명 : sex\_ck).
- 'id'는 <doctor> 테이블에 있는 'doc\_id'를 참조한다(제약조건명 : id\_fk).

<SQL문>

```
CREATE TABLE patient (
    id CHAR(5) ( ),
    name CHAR(10),
    sex CHAR(1),
    phone CHAR(20),
    CONSTRAINT sex_ck CHECK (sex='f' or sex='m'),
    CONSTRAINT id_fk FOREIGN KEY(id) REFERENCES doctor(doc_id)
);
```

**답 :**

4640403



**문제 3** 다음에 제시된 운영체제의 발달 과정을 순서대로 기호(㉠~㉢)로 나열하시오. (5점)

- |             |             |
|-------------|-------------|
| ㉠ 다중 처리 시스템 | ㉡ 범용 시스템    |
| ㉢ 분산 처리 시스템 | ㉣ 일괄 처리 시스템 |

**답 :** (     ) → (     ) → (     ) → (     )

4640404



**문제 4** 다음의 두 릴레이션(Relation)에서 외래키(Foreign)를 찾아 쓰시오. (밑줄은 기본키를 의미함) (5점)

제품(제품코드, 제품명, 단가, 구매처)  
판매(판매코드, 판매처, 제품코드, 수량)

**답 :**

4640405



**문제 5** OSI 7계층 중 응용 계층에서 사용되는 프로토콜을 모두 골라 기호(㉠~㉣)로 쓰시오. (5점)

- |        |        |       |         |
|--------|--------|-------|---------|
| ㉠ http | ㉡ pop3 | ㉢ ftp | ㉣ tcp   |
| ㉤ icmp | ㉥ ip   | ㉦ udp | ㉧ ipsec |

**답 :**

4640406



**문제 6** 관계형 데이터베이스에서 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic) 값들의 집합을 의미하는 용어를 쓰시오. (5점)

**답 :**

**문제 7** 다음은 <Product> 테이블로부터 price가 100000 미만인 제품들의 ID, NAME, price를 가져와 <sale> 뷰를 생성하는 SQL문이다. 괄호(①, ②)에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

4640407



<SQL문>

```
create view sale ( ① )
select ID, NAME, price
from Product
( ② ) price < 100000;
```

**답**

- ①
- ②

**문제 8** 다음은 <목표> 테이블의 매출액이 5000 초과인 제품에 대해 <판매> 테이블의 수량을 <목표> 테이블의 수량으로 변경하는 SQL문이다. 괄호에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

4640408



<SQL문>

```
update 판매 a inner join 목표 b
( ) a.ID = b.ID
set a.수량 = b.수량
where b.매출액 > 5000;
```

**답 :**

**문제 9** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

4640409



```
#include <stdio.h>
int main()
{
    int calc, x = 40, y = 60, z = 80;
    calc = x < y ? y++ : --z;
    printf("%d//%d//%d", calc, y, z);
    return 0;
}
```

**답 :**

4640410



**문제 10** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>

int main()
{
    char a[] = {'1', 'B', 'C', 'D', 'E'};
    char *p;
    p = &a[2];
    printf("%c%c", *p, *(p-2));
    return 0;
}
```

**답 :**

4640411



**문제 11** 임꺽정에게 부여된 <교수> 테이블에 대한 UPDATE와 DELETE 권한을 취소하는 SQL문을 작성하려고 한다. 괄호에 적합한 예약어를 넣어 SQL문을 완성하시오. (5점)

<SQL문>

(        ) UPDATE, DELETE ON 교수 FROM 임꺽정;

**답 :**

4640412



**문제 12** 다음 설명에 해당하는 알맞은 용어를 영문 4글자로 쓰시오. (5점)

IP 주소 부족 문제를 해결하기 위해 만들어진 프로토콜로, 동적 호스트 구성 프로토콜이라고 불린다. 네트워크 관리자들이 조직 내의 네트워크 상에서 IP 주소를 중앙에서 관리하고 할당할 수 있도록 지원하며, 주어진 IP 주소가 일정한 시간 동안만 그 컴퓨터에 유효하도록 하는 '임대' 개념을 사용하여, 사용 가능한 IP 주소의 개수보다 더 많은 컴퓨터가 있는 경우에 IP 주소의 임대 시간을 짧게 함으로써 IP 주소를 효율적으로 사용할 수 있다.

**답 :**



**문제 13** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

4640413



네트워크에서 서로 다른 컴퓨터들 간에 정보 교환을 할 수 있게 해주는 통신 규약으로, 흐름 제어, 동기화, 오류 검출의 기능을 수행한다.

**답 :**

**문제 14** 리눅스 또는 유닉스에서 사용하는 명령어 중 다음에 해당하는 명령을 <보기>에서 찾아 쓰시오. (5점)

4640414



- ① 파일을 삭제한다.
- ② 파일의 보호 모드를 설정하여 파일의 사용 허가를 지정한다.

<보기>

- |       |        |         |         |
|-------|--------|---------|---------|
| • cat | • cd   | • chmod | • find  |
| • rm  | • kill | • ls    | • chown |

**답**

- ①
- ②

**문제 15** 다음에 제시된 <보기>는 소프트웨어 개발 과정에서 진행되는 소프트웨어 테스트들을 나열한 것이다. <보기>에 제시된 테스트 진행 과정을 순서대로 기호(㉠~㉤)로 나열하시오. (5점)

4640415



<보기>

- ㉠ 시스템 테스트
- ㉡ 단위 테스트
- ㉢ 통합 테스트
- ㉣ 인수 테스트

**답 :** (    ) → (    ) → (    ) → (    )



**문제 16** 운영체제의 커널을 하드디스크와 같은 보조기억장치에서 주기억장치로 옮겨 시스템을 사용할 수 있도록 초기화하는 기능을 수행하는 것을 찾아 기호(㉠~㉤)로 쓰시오. (5점)

- |        |        |                    |
|--------|--------|--------------------|
| ㉠ BIOS | ㉡ CMOS | ㉢ Bootstrap Loader |
| ㉣ RAM  | ㉤ ROM  | ㉥ MBR              |

**답 :**



**문제 17** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>
int main()
{
    int map[5][5] = {
        1, 5, 6, 7, 8,
        2, 4, 6, 4, 9,
        1, 5, 7, 4, 2,
        2, 3, 4, 5, 5,
        5, 2, 4, 1, 1 };
    int i = 0, j = 0;
    int res = map[i][j];
    while(1)
    {
        if (i==4 && j==4) break;
        else if (i==4) j++;
        else if (j==4) i++;
        else if (map[i+1][j] >= map[i][j+1]) j++;
        else
            i++;
        res += map[i][j];
    }
    printf("result : %d", res);
    return 0;
}
```

**답 :**

4640418



**문제 18** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
import java.lang.*;
public class Main
{
    public static void main(String[] args) {
        switch((int)Math.signum(-100)) {
            case -1:
                System.out.print("N");
                break;
            case 0:
                System.out.print("P");
                break;
            case 1:
                System.out.print("E");
                break;
            default:
                System.out.print("Z");
        }
    }
}
```

**답 :**



**문제 19** 다음은 결과로 100을 산출하는 프로그램을 Java로 구현한 것이다. 괄호에 들어갈 알맞은 예약어를 <보기>에서 찾아 기호(㉠~㉤)로 쓰시오. (5점)

```
interface Cals {
    public void get(int v);
}

class Test (    ) Cals {
    public void get(int v) {
        System.out.print(v*v);
    }
}

public class Main {
    public static void main(String[] args) {
        Cals a = new Test();
        a.get(10);
    }
}
```

<보기>

- ㉠ new
- ㉡ abstract
- ㉢ super
- ㉣ extends
- ㉤ implements

**답 :**

4640420



**문제 20** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
class ClassA {
    int a = 1;
    int b = 1;
}

class ClassB extends ClassA {
    void testcase() {
        System.out.println(this.a * this.b);
    }
    void testcase(int i) {
        System.out.println(this.a - this.b);
    }
    void testcase(char i) {
        System.out.println(this.a / this.b);
    }
    void testcase(float i) {
        System.out.println(this.a + this.b);
    }
}

public class Main
{
    public static void main(String[] args) {
        int a = 10;
        int b = 3;

        ClassB c = new ClassB();
        c.testcase(a/b);
    }
}
```

**답 :**

## [문제 1]

① ㉠ ② ㉡

## [문제 2]

PRIMARY KEY

## 해설

```
CREATE TABLE patient (
  id CHAR(5) PRIMARY KEY,
  name CHAR(10),
  sex CHAR(1),
  phone CHAR(20),
  CONSTRAINT sex_ck
    CHECK (sex='f' or sex='m'),
  CONSTRAINT id_fk FOREIGN KEY(id)
    REFERENCES doctor(doc_id)
);
```

생성한 테이블 이름은 <patient>이다.  
 'id' 속성은 문자 5자리이며, 기본키이다.  
 'name' 속성은 문자 10자리이다.  
 'sex' 속성은 문자 1자리이다.  
 'phone' 속성은 문자 20자리이다.  
 'sex' 속성은 'f' 또는 'm'만 입력되어야 하며,  
 이 제약 조건의 이름은 'sex\_ck'이다.  
 'id' 속성은 <doctor> 테이블의 기본키인 'doc\_id' 속성을  
 참조하는 외래키이며, 이 제약 조건의 이름은 'id\_fk'이다.

## [문제 3]

㉡ → ㉠ → ㉢ → ㉣

## [문제 4]

제품코드

## 해설

외래키는 다른 릴레이션의 기본키를 참조하는 속성으로, 여기서는 <제품> 테이블의 기본키인 '제품코드'를 참조하는 <판매> 테이블의 '제품코드'가 외래키입니다.

## [문제 5]

㉠, ㉢, ㉣

## [문제 6]

※ 다음 중 하나를 쓰면 됩니다.

도메인, Domain

## [문제 7]

① as ② where

## 해설

```
create view sale
as select ID, NAME, price
from Product
where price < 100000;
```

생성한 뷰의 이름은 <sale>이다.  
 'ID', 'NAME', 'price' 속성을 가져온다.  
 <Product> 테이블에서 속성을 가져온다.  
 'price'의 값이 100000 미만인 자료만을 대상으로 한다.

## [문제 8]

on

## 해설

update 판매 a inner join 목표 b

on a.ID = b.ID

set a.수량 = b.수량

where b.매출액 &gt; 5000;

&lt;a&gt;라는 별칭의 &lt;판매&gt; 테이블과 &lt;b&gt;라는 별칭의 &lt;목표&gt; 테이블을 이용하여 갱신한다.

&lt;판매&gt; 테이블의 'ID'와 &lt;목표&gt; 테이블의 'ID'를 기준으로 서로 JOIN한다.

&lt;판매&gt; 테이블의 '수량'을 &lt;목표&gt; 테이블의 '수량'으로 변경한다.

&lt;목표&gt; 테이블의 '매출액'이 5000을 초과하는 자료만을 대상으로 한다.

## [문제 9]

60//61//80

## 해설

```
#include <stdio.h>
int main()
{
  ① int calc, x = 40, y = 60, z = 80;
  ② calc = x < y ? y++ : --z;
  ③ printf("%d//%d//%d", calc, y, z);
  ④ return 0;
}
```

① 정수형 변수 calc, x, y, z를 선언하고 x는 40, y는 60, z는 80으로 초기화한다.

② x가 y보다 작으면 y를 calc에 저장한 후 y의 값을 1 증가시키고, x가 y보다 크거나 같으면 z를 1 감소시킨 값을 calc에 저장한다. x의 값 40은 y의 값 60보다 작으므로 calc에 60을 저장한 후 y의 값은 1 증가되어 61이 된다.

③ calc, y, z의 값 사이에 //을 포함하여 출력한다.

결과 60//61//80

④ main() 함수에서의 return 0;은 프로그램의 종료를 의미한다.

## [문제 10]

C1

## 해설

```
#include <stdio.h>
int main()
{
  ① char a[] = {'1', 'B', 'C', 'D', 'E'};
  ② char *p;
  ③ p = &a[2];
  ④ printf("%c%c", *p, *(p-2));
  ⑤ return 0;
}
```

❶ 5개의 요소를 갖는 문자형 배열 `a`를 선언한다. 개수를 지정하지 않았으므로, 초기값으로 지정된 수만큼 배열의 요소가 만들어진다.

[0] [1] [2] [3] [4]

a

1

B

C

D

E

❷ 문자형 포인터 변수 `p`를 선언한다.

❸ `p`에 `a[2]`의 주소를 저장한다. 다음의 그림에서 `a`가 할당된 공간의 주소는 임의로 정한 것이며, 이해를 돕기 위해 10진수로 표현했다.

주소

메모리

a 1000

1Byte	1Byte	1Byte	1Byte	1Byte
a[0]	a[1]	a[2]	a[3]	a[4]
1	B	C	D	E
1000	1001	1002	1003	1004
p-2	p-1	p	p+1	p+2

❹ `p`가 가리키는 곳의 값과 `p-2`가 가리키는 곳의 값을 문자로 출력한다.

결과

C1

❺ `main()` 함수에서의 `return 0;`은 프로그램의 종료를 의미한다.

[문제 11]

REVOKE

[문제 12]

DHCP

[문제 13]

※ 다음 중 하나를 쓰면 됩니다.

프로토콜, Protocol

[문제 14]

① rm    ② chmod

[문제 15]

㉠ → ㉡ → ㉢ → ㉣

[문제 16]

㉡



## [문제 17]

result : 17

## 해설

```

#include <stdio.h>
int main()
{
    ❶ int map[5][5] = {
        1, 5, 6, 7, 8,
        2, 4, 6, 4, 9,
        1, 5, 7, 4, 2,
        2, 3, 4, 5, 5,
        5, 2, 4, 1, 1 };
    ❷ int i = 0, j = 0;
    ❸ int res = map[i][j];
    ❹ while(1)
    {
        ❺ if (i==4 && j==4) break;
        ❻ else if (i==4) j++;
        ❼ else if (j==4) i++;
        ❽ else if (map[i+1][j] >= map[i][j+1]) j++;
        ❾ else
            i++;
        ❿ res += map[i][j];
    }
    ⓫ printf("result : %d", res);
    ⓬ return 0;
}

```

❶ 5행 5열의 요소를 갖는 정수형 2차원 배열 `map`을 선언하고 초기화한다.

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

- ❷ 정수형 변수 `i`와 `j`를 선언하고 각각 0으로 초기화한다.
- ❸ 정수형 변수 `res`를 선언하고 `map[i][j]`의 값인 1로 초기화한다.
- ❹ `while`의 조건이 1, 즉 참이므로 ❺~❿를 무한 반복한다.
- ❺ `i`와 `j`가 모두 4이면, `while`문을 벗어나 ⓫번으로 이동한다.
- ❻ 그렇지 않고 `i`가 4이면 `j`의 값에 1을 누적시킨다.
- ❼ 그렇지 않고 `j`가 4이면 `i`의 값에 1을 누적시킨다.
- ❽ 그렇지 않고 `map[i+1][j]`가 `map[i][j+1]`보다 크거나 같으면 `j`의 값에 1을 누적시킨다.
- ❾ ❺~❽ 조건에 해당하지 않으면 `i`의 값에 1을 누적시킨다.
- ❿ `res = res + map[i][j];`와 동일하다. `res`에 `map[i][j]`의 값을 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	j	map[i+1][j]	map[i][j+1]	map[i][j]	res
0	0	[1][0]=2	[0][1]=5	[0][0]=1	1
1				[1][0]=2	3
2		[2][0]=1	[1][1]=4	[2][0]=1	4
		[3][0]=2	[2][1]=5		
3				[3][0]=2	6
	1	[4][0]=5	[3][1]=3	[3][1]=3	9
4		[4][1]=2	[3][2]=4	[4][1]=2	11
	2			[4][2]=4	15
	3			[4][3]=1	16
	4			[4][4]=1	17

⑪ result : 를 출력한 후 res의 값 17을 정수로 출력한다.

결과 result : 17

⑫ main() 함수에서의 return 0;은 프로그램의 종료를 의미한다.

[문제 18]

N

해설

```
import java.lang.*;           Math 클래스를 사용하기 위해 java.lang 패키지를 선언한다.
public class Main
{
    public static void main(String[] args) {
        ① switch((int)Math.signum(-100)) {
        ②     case -1:
        ③         System.out.print("N");
        ④         break;
            case 0:
                System.out.print("P");
                break;
            case 1:
                System.out.print("E");
                break;
            default:
                System.out.print("Z");
        }
    } ⑤
}
```

① '(int)Math.signum(-100)'의 결과에 해당하는 숫자를 찾아간다. signum(-100)에서 인수가 음수이므로 -1.0을 반환하고 반환된 값은 (int)-1.0에 의해 정수인 -1로 변환된다. -1에 해당하는 ②번으로 이동한다.

• Math : 수학 관련 메소드들이 포함되어 있는 클래스

• signum() : 인수가 양수이면 1.0, 음수이면 -1.0을 반환함

② '(int)Math.signum(-100)'이 -1인 경우 찾아오는 곳이다.

③ N을 출력한다.

결과 N

④ switch문을 벗어나 ⑤번으로 이동한 후 프로그램을 종료한다.

## [문제 19]

ⓐ

## 해설

```

① interface Cals {
②     public void get(int v);
}

③ class Test implements Cals {
④     ③ public void get(int v) {
        ④     System.out.print(v*v);
    }
}

public class Main {
    public static void main(String[] args) {
①         Cals a = new Test();
②         a.get(10);
    } ⑤
}

```

① 인터페이스 Cals를 선언한다.

② 반환값이 없는 get( )을 선언한다. 인터페이스에 선언된 메소드는 선언만 있고 내부에 실행 코드가 없는 추상 메소드이므로, 이후 상속 관계가 설정된 자식 클래스에서 재정의한 후 사용한다.

③ Cals 인터페이스를 상속받는 클래스 Test를 정의한다.

④ 반환값이 없는 get( )을 정의한다. get( ) 메소드는 ③에서 선언된 추상 메소드(②)를 재정의하는 것이다.

※ 추상 메소드는 부모 클래스가 자식 클래스에게 주는 의무와 같습니다. 부모 클래스와 상속 관계에 있다면 반드시 부모 클래스의 추상 메소드를 재정의해야 합니다. 그렇지 않으면 프로그램에 오류가 발생합니다.

모든 Java 프로그램은 반드시 main( ) 메소드에서 시작한다.

① Test 클래스의 생성자를 이용하여 Cals의 객체 변수 a를 선언한다. 부모 클래스의 객체 변수를 선언하면서 자식 클래스의 생성자를 사용하면 형 변환이 발생한다. 이렇게 형 변환이 발생했을 때 부모 클래스와 자식 클래스에 동일한 속성이나 메소드가 있으면 부모 클래스의 속성이나 메소드가 자식 클래스의 속성이나 메소드로 재정의된다.

② 10을 인수로 하여 a의 get( ) 메소드를 호출한다. ③번으로 이동한다.

※ get( ) 메소드는 a 객체의 자료형이 Cals이므로 Cals 클래스의 get( )이라고 생각할 수 있지만 ①번에서 클래스 형 변환이 발생하였고, get( ) 메소드가 자식 클래스에서 재정의되었으므로 Test 클래스의 get( ) 메소드가 수행됩니다.

③ 반환값이 없는 get( ) 메소드의 시작점이다. ②번에서 전달받은 10을 v가 받는다.

④ v\*v의 결과값인 100을 출력한 후 메소드가 종료되면, 호출했던 ②번의 다음 줄인 ⑤번으로 이동하여 프로그램을 종료한다.

결과 100

## [문제 20]

0

## 해설

```

A class ClassA {
    int a = 1;
    int b = 1;
}

B class ClassB extends ClassA {
C void testcase() {
    System.out.println(this.a * this.b);
}
D 5 void testcase(int i) {
    6 System.out.println(this.a - this.b);
}
E void testcase(char i) {
    System.out.println(this.a / this.b);
}
F void testcase(float i) {
    System.out.println(this.a + this.b);
}
}
public class Main
{
    public static void main(String[] args) {
        1 int a = 10;
        2 int b = 3;

        3 ClassB c = new ClassB();
        4 c.testcase(a/b);
    } 7
}

```

A ClassA 클래스를 정의한다.

B ClassB 클래스를 정의하고 부모 클래스로 ClassA를 지정하면서 ClassA에 속한 변수를 상속받는다.

C 반환값 없는 메소드 testcase()를 정의한다.

D 반환값 없는 메소드 testcase(int i)를 정의한다. 메소드 이름이 C, E, F와 같지만 '인수를 받는 자료형'이 다르므로 서로 다른 메소드이다. 이렇게 이름은 같지만 자료형을 달리하여 여러 기능을 정의하는 것을 오버로딩(Overloading)이라고 한다.

모든 Java 프로그램의 실행은 반드시 main() 메소드에서 시작한다.

1 정수형 변수 a를 선언하고 10으로 초기화한다.

2 정수형 변수 b를 선언하고 3으로 초기화한다.

3 클래스 ClassB의 객체 변수 c를 선언한다.

4 a/b의 결과인 3을 인수로 하여 c의 testcase() 메소드를 호출한다. 5번으로 이동한다.

※ 1, 2번에서 a와 b를 정수형으로 선언했으므로 a/b의 결과는 3.3333333이 아닌 3입니다.

※ a/b의 결과값이 정수형이므로 C~F 중 4번에서 호출되는 메소드는 정수형 자료형을 인수로 하는 testcase(int i) 메소드입니다.

5 반환값 없는 testcase(int i) 메소드의 시작점이다. 4번에서 전달받은 3을 i가 받는다.

⑥ 'this.a - this.b'의 값, 즉 ClassA에서 선언한 a에서 b를 뺀 값 0을 출력한다. 메소드가 종료되면, 호출했던 ④번의 다음 줄인 ⑦번으로 이동하여 프로그램을 종료한다.

※ **this** : 현재 실행중인 메소드가 속한 클래스를 가리키는 예약어로, 현재 실행중인 testcase(int i) 메소드가 속한 ClassB와 ClassB가 상속받은 ClassA 클래스 모두를 가리킵니다.

결과 0



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익의 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 굵지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

4640301



**문제 1** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main()
{
    char a[] = "A B c D e F ! ";
    delBl(a);
    printf("%s", a);
    return 0;
}

void delBl(char a[])
{
    int len = strlen(a);
    char* str = (char*)malloc(sizeof(char) * len);
    int i, k = 0;

    for(i = 0; i < len; i++)
    {
        if (a[i] != ' ')
            str[k++] = a[i];
    }
    str[k] = '\0';
    strcpy(a, str);
    free(str);
}
```

**답 :**

**문제 2** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

- 로컬 서버에서 프로그램을 이용하여 메일을 액세스하기 위한 표준 프로토콜이다.
- 다중 로그인을 지원하기 때문에 다양한 위치에서 메일을 검색하고 관리할 수 있다.
- 메일 서버에서 메일 제목이나 보낸 사람을 확인한 후 선택적으로 메일을 다운로드할 수 있다.

4640302



답 :

**문제 3** Windows 10을 사용하는 컴퓨터에서 컴퓨터를 잠그거나 사용자를 전환하기 위해 사용하는 단축 키를 쓰시오. (5점)

4640303



답 :

**문제 4** OSI 7계층 중 개방 시스템들 간의 네트워크 연결 관리, 데이터의 교환 및 중계를 담당하며, ARP, IPX, IP 프로토콜과 관련된 계층은 무엇인지 쓰시오. (5점)

4640304



답 :



4640305



**문제 5** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
public class Test {
    public static void main(String[] args) {
        String str1 = "HELlloWorLD!";
        String str2 = "heLLowORld!";

        if (str1.equals(str2))
            System.out.print(str1.toUpperCase());
        else if (str1.equalsIgnoreCase(str2))
            System.out.print(str1.toLowerCase());
        else
            System.out.print(str2);
    }
}
```

**답 :**

4640306



**문제 6** <학생> 테이블에는 '학번', '이름', '성적', '학과코드' 필드가 있고 <학과> 테이블에는 '학과코드', '학과명' 필드가 있으며, <학과등급> 테이블에는 '최저'와 '최고' 필드가 있을 때 '성적' 필드의 값이 <학과등급> 테이블의 '최저' 필드의 값 이상이고 '최고' 필드의 값 이하인 자료만을 추출하려고 한다. 다음 <SQL문>의 괄호에 알맞은 연산자를 쓰시오. (5점)

<SQL문>

```
Select 학번, 이름, 학과명
From 학생, 학과, 학과등급
Where 학생.학과코드 = 학과.학과코드 AND 학생.성적 (    ) 학과등급.최저 AND 학과등급.최고;
```

**답 :**

4640307

**문제 7** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

- UNIX 시스템의 구성 요소 중 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.
- 명령을 해석하여 처리할 수 있도록 커널로 전달해주는 명령 인터프리터로, 단말장치를 통하여 사용자로부터 명령어를 입력받는다.

**답 :**

4640308

**문제 8** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>
int main()
{
    int a = 15;
    printf("%o", a);
    return 0;
}
```

**답 :**

4640309

**문제 9** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>

int printBin(int a)
{
    if (a == 0 | a == 1) printf("%d", a);
    else
    {
        printBin(a/2);
        printf("%d", a%2);
    }
}
```

```
int main()
{
    int x = 11;
    printBin(x);
    return 0;
}
```

답 :

**문제 10** <학생> 테이블에서 '학과' 별로 '용돈'의 평균을 구한 후 '평균'이라는 이름으로 표시하되, 학과를 기준으로 오름차순 정렬하려고 한다. 다음 <SQL문>의 괄호에 알맞은 예약어를 쓰시오. (5점)

4640310



<SQL문>

Select 학과, (            ) (용돈) As 평균 From 학생 Group By 학과 Order By 학과 asc;

답 :

**문제 11** 문자로 된 도메인 네임을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템을 무엇이라고 하는지 쓰시오. (5점)

4640311



답 :



**문제 12** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
public class Test
{
    public static void main(String[] args) {
        String str = "1,2,3,4,,,5,6,7,,8,9";
        String[] splittest = str.split(",");

        for(int i = 0; i<splittest.length; i++)
        {
            System.out.print(splittest[i]);
            if ((i+1) % 3 == 0)
                System.out.println();
        }
    }
}
```

**답 :**



**문제 13** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

- 후보키 중에서 특별히 선정된 키로 중복된 값을 가질 수 없다.
- 유일성과 최소성을 가지며 튜플을 식별하기 위해 반드시 필요한 키다.
- NULL 값을 가질 수 없다.

**답 :**

**문제 14** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

4640314



```
public class Test
{
    public static void main(String[] args) {
        String str = "HELLO!@#WORLD/-";
        String res = str.replaceAll("[^ㄱ-ㅎㅌ-ㅣ가-힣a-zA-Z0-9,.]", "");
        System.out.print(res);
    }
}
```

**답 :**

**문제 15** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

4640315



A, B, C 3개의 속성을 가진 릴레이션 R에서 어떤 복합 속성(A, C)에 대응하는 B 값의 집합이 A 값에만 종속되고 C 값에는 무관한 경우로,  $A \twoheadrightarrow B$ 로 표기한다.

**답 :**

**문제 16** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

4640316



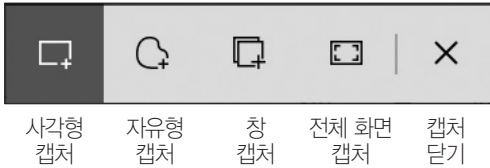
- 소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 기능 테스트라고도 한다.
- 프로그램의 구조를 고려하지 않기 때문에 테스트 케이스는 프로그램 또는 모듈의 요구나 명세를 기초로 결정한다.
- 소프트웨어 인터페이스에서 실시되는 테스트이다.
- 부정확하거나 누락된 기능, 인터페이스 오류, 자료 구조나 외부 데이터베이스 접근에 따른 오류, 행위나 성능 오류, 초기화와 종료 오류 등을 발견하기 위해 사용되며, 테스트 과정의 후반부에 적용된다.

**답 :**

4440317



**문제 17** Windows 10을 사용하는 컴퓨터에서 화면 상단에 다음과 같이 캡처 도구와 유사한 스크린샷 바를 표시하는 단축키를 쓰시오. (5점)



**답 :**

4440318



**문제 18** <학생> 테이블을 대상으로 '학과코드'의 값이 "ELE"인 경우 "전자공학과"라는 별칭으로 개수를 1씩 증가시키고, '학과코드'의 값이 "COM"인 경우 "컴퓨터공학과"라는 별칭으로 개수를 1씩 증가시킨 후 <출력형태>와 같이 출력하려고 한다. 다음 <SQL문>의 괄호에 공통으로 들어갈 알맞은 명령을 쓰시오. (5점)

<출력형태>

전자공학과	컴퓨터공학과
2	3

<SQL문>

```
select count( ( ) when 학과코드 = "ELE" then 1 end) As "전자공학과",  
count( ( ) when 학과코드 = "COM" then 1 end) As "컴퓨터공학과" from 학생;
```

**답 :**

**문제 19** 다음에서 설명하는 스키마의 종류를 쓰시오. (5점)

4640319



물리적 저장장치의 입장에서 본 데이터베이스 구조로서, 실제로 데이터베이스에 저장될 레코드의 형식을 정의하고 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 등을 나타낸다.

**답 :****문제 20** 다음 〈처리 조건〉을 준수하여 색인을 생성하는 〈SQL문〉의 괄호에 들어갈 알맞은 명령어를 쓰시오. (5점)

4640320



〈처리 조건〉

- 기본 테이블 T의 열(P, Q, R)에 관한 조합으로 X 색인을 생성한다.
- 색인 내용은 P(오름차순), Q(내림차순), R(오름차순)이다.
- SQL 작성 시 UNIQUE와 CLUSTER는 생략 가능하다.

〈SQL문〉

CREATE (            ) X ON T(P, Q DESC, R);

**답 :**

[문제 1]

ABcDeF!

## 해설

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main()
{
    ① char a[] = "A B c D e F ! ";
    ② delB1(a);
    ⑬ printf("%s", a);
    ⑭ return 0;
}

③ void delB1(char a[])
{
    ④ int len = strlen(a);
    ⑤ char* str = (char*)malloc(sizeof(char) * len);
    ⑥ int i, k = 0;

    ⑦ for(i = 0; i < len; i++)
    {
        ⑧ if (a[i] != ' ')
        ⑨ str[k++] = a[i];
    }
    ⑩ str[k] = '\0';
    ⑪ strcpy(a, str);
    ⑫ free(str);
}

```

모든 C 프로그램은 반드시 main() 함수에서 시작한다.

① 16개의 요소를 갖는 문자형 배열 a를 선언한다. 개수를 지정하지 않았으므로, 초기값으로 지정된 수만큼 배열의 요소가 만들어진다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
a	A		B		c			D		e			F		!	

② 배열 a의 시작 주소를 인수로 하여 delB1() 함수를 호출한다.

③ delB1 함수의 시작점이다. ②번에서 전달받은 주소를 배열 a가 받는다.

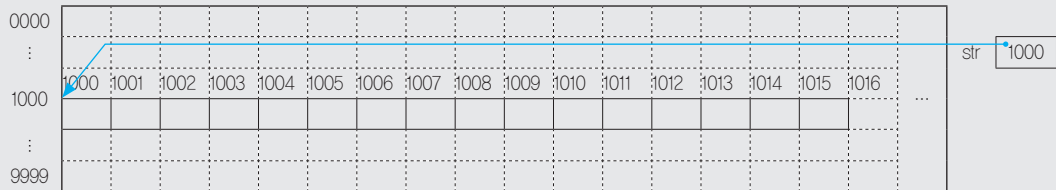
④ 정수형 변수 len을 선언하고, a 배열에 저장된 문자열, 즉 "A B c D e F !"의 길이인 16을 저장한다.

• strlen(): 문자열의 길이를 구하는 함수이다.

⑤ malloc 함수가 메모리에서 char 자료형의 크기 \* 16, 즉 16Byte의 빈 영역을 찾아 할당한다. 그 영역의 시작 주소를 포인터 변수 str에 저장한다. 이제 str은 할당된 공간의 시작 주소를 가리킨다.

\* malloc 함수가 동적으로 할당하는 것이므로 여기서 지정한 주소 1000은 임의로 정한 것이며, 이해를 돕기 위해 주소를 10진수로 표현했습니다.





- ⑥ 정수형 변수 `i`와 `k`를 선언하고 `k`를 0으로 초기화한다.  
 ⑦ 반복 변수 `i`가 0부터 1씩 증가하면서 16보다 작은 동안 ⑧~⑨번을 반복 수행한다.  
 ⑧ `a[i]`의 값이 ''(공백)이 아니면 ⑨번 문장을 수행한다.  
 ⑨ `a[i]`의 값을 `str[k]`에 저장한다. `k++`은 후치 증가 연산자이므로 `str[k]`에 값이 저장된 후 1 증가한다.  
 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	a[i]	k	a 배열 / str 배열
0	A	0	<div> <div>a</div> <div>str</div> </div>
1		1	
2	B	2	
3			
4	c	3	
5			
6			
7	D	4	
8			
9	e	5	
10			
11			
12	F	6	
13			
14	!	7	
15			
16			

- ⑩ `str[k]`에 '\0'(널 문자)를 저장한다. 널 문자('\0')는 문자열의 끝을 알리는 용도이다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
str	A	B	c	D	e	F	!	\0								

- ⑪ 문자열 배열 `str`을 문자열 배열 `a`에 복사한다.

• `strcpy(a, b)`: 문자 배열 `a`에 문자 배열 `b`의 값을 복사한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
a	A	B	c	D	e	F	!	\0								

- ⑫ `str`이 가리키고 있는 메모리 공간을 해제한다. 즉 `str`이 가리키고 있는 1000 번지부터 할당된 16Byte 공간을 해제한다.

• `free()`: `malloc()` 함수에 의해 동적으로 할당된 메모리를 해제한다.

- ⑬ 배열 `a`에 저장된 문자열을 출력한다.

결과 **ABcDeF!**

- ⑭ `main()` 함수에서의 `return 0`;은 프로그램의 종료를 의미한다.

[문제 2]

※ 다음 중 하나를 쓰면 됩니다.

IMAP, Internet Message Access Protocol

[문제 3]

Winkey + L

[문제 4]

※ 다음 중 하나를 쓰면 됩니다.

네트워크 계층, Network Layer

[문제 5]

helloworld!

해설

```
public class Test {
    public static void main(String[] args) {
        ❶ String str1 = "HELloWorLD!";
        ❷ String str2 = "heLLOwORld!";

        ❸ if (str1.equals(str2))
            System.out.print(str1.toUpperCase());
        ❹ else if (str1.equalsIgnoreCase(str2))
        ❺ System.out.print(str1.toLowerCase());
        else
            System.out.print(str2);
    }
}
```

- ❶ 문자열 변수 str1을 선언하고 HELloWorLD!로 초기화한다.
- ❷ 문자열 변수 str2를 선언하고 heLLOwORld!로 초기화한다.
- ❸ str1과 str2를 대소문자를 구분하여 비교한 후 같으면 다음 문장을 수행하고 그렇지 않으면 ❹번을 실행한다. str1과 str2는 같지 않으므로 ❹번을 실행한다.
  - **A.equals(B)**: 문자열 A와 B를 대소문자를 구분하여 비교한 후 같으면 참, 다르면 거짓을 반환한다.
- ❹ str1과 str2를 대소문자 구분없이 비교하여 같으면 ❺번을 수행한다. str1과 str2는 같으므로 ❺번을 실행한다.
  - **A.equalsIgnoreCase(B)**: 대소문자 구분없이 문자열 A와 B를 비교한 후 같으면 참, 다르면 거짓을 반환한다.
- ❺ str1을 모두 소문자로 변환한 후 출력한다.
  - **toLowerCase()**: 문자열을 모두 소문자로 변환한다.

결과 helloworld!

## [문제 6]

between

※ **답안 작성 시 주의 사항** : SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```

Select 학번, 이름, 학과명
From 학생, 학과, 학과등급
Where 학생.학과코드 = 학과.학과코드
      AND 학생.성적 Between 학과등급.최저 AND 학과등급.최고;

```

‘학번’, ‘이름’, ‘학과명’을 표시한다.  
 〈학생〉, 〈학과〉, 〈학과등급〉 테이블을 대상으로 검색한다.  
 〈학생〉 테이블의 ‘학과코드’와 〈학과〉 테이블의 ‘학과코드’가 같고  
 〈학생〉 테이블의 ‘성적’이 〈학과등급〉 테이블의 ‘최저’와 ‘최고’ 사이인 학생만을 대상으로 한다.

## [문제 7]

※ 다음 중 하나를 쓰면 됩니다.

셸, Shell

## [문제 8]

17

## 해설

```

#include <stdio.h>
int main()
{
  ❶ int a = 15;
  ❷ printf("%o", a);
  ❸ return 0;
}

```

❶ 정수형 변수 a를 선언하고 15로 초기화한다.

❷ a의 값을 8진수로 출력한다.

10진수 15를 8진수로 변환하면 다음과 같다.

```

8 | 15
  | 1 ... 7

```

결과 17

❸ main() 함수에서의 **return 0**은 프로그램의 종료를 의미한다.

## [문제 9]

1011

## 해설

```
int main()
{
  ❶ int x = 11;
  ❷ printBin(x);
  return 0;
}
```

모든 C언어 프로그램은 반드시 main() 함수에서 시작한다.

- ❶ 정수형 변수 x를 선언하고 11로 초기화한다.
- ❷ x의 값 11을 인수로 하여 printBin(11) 함수를 호출한다.

```
3 int printBin(int a)      //a는 11이다.
{
  4   if (a == 0 | a == 1) printf("%d", a);
    else
    {
      5     printBin(a/2);
        printf("%d", a%2);
    }
}
```

- ❸ 정수를 반환하는 printBin() 함수의 시작점이다. ❷번에서 전달받은 11을 a가 받는다.
- ❹ a가 0 또는 1이면 a의 값을 정수로 출력하고 그렇지 않으면 ❺번을 실행한다. a가 11이므로 ❺번을 실행한다.
- ❺ a/2의 값인 5를 인수로 하여 printBin(5) 함수를 호출한다.

```
6 int printBin(int a)      //a는 5다.
{
  7   if (a == 0 | a == 1) printf("%d", a);
    else
    {
      8     printBin(a/2);
        printf("%d", a%2);
    }
}
```

- ❻ 정수를 반환하는 printBin() 함수의 시작점이다. ❸번에서 전달받은 5를 a가 받는다.
- ❼ a가 5이므로 ❸번을 실행한다.
- ❽ a/2의 값인 2를 인수로 하여 printBin(2) 함수를 호출한다.

```

9 int printBin(int a)      //a는 2다.
{
10     if (a == 0 | a == 1) printf("%d", a);
    else
    {
11         printBin(a/2);
        printf("%d", a%2);
    }
}

```

9 정수를 반환하는 printBin() 함수의 시작점이다. 9번에서 전달받은 2를 a가 받는다.

10 a가 2이므로 10번을 실행한다.

11 a/2의 값인 1을 인수로 하여 printBin(1) 함수를 호출한다.

```

12 int printBin(int a)      //a는 1이다.
{
13     if (a == 0 | a == 1) printf("%d", a);
    else
    {
        printBin(a/2);
        printf("%d", a%2);
    }
}

```

12 정수를 반환하는 printBin() 함수의 시작점이다. 11번에서 전달받은 1을 a가 받는다.

13 a가 1이므로 1을 출력한 후 제어를 printBin(1)을 호출했던 11번 아래 문장인 14번으로 옮긴다.

결과 1

```

9 int printBin(int a)      //a는 2다.
{
10     if (a == 0 | a == 1) printf("%d", a);
    else
    {
11         printBin(a/2);
14         printf("%d", a%2);
    }
}

```

14 printBin(1)을 호출할 때 a는 2였으므로 a%2의 값인 0을 출력한 후 제어를 printBin(2)를 호출했던 9번 아래 문장인 9번으로 옮긴다.

결과 10

```
⑥ int printBin(int a)      //a는 5다.
{
  ⑦   if (a == 0 | a == 1) printf("%d", a);
      else
      {
        ⑧   printBin(a/2);
        ⑬   printf("%d", a%2);
      }
}
```

⑬ printBin(2)를 호출할 때 a는 5였으므로 a%2의 값인 1을 출력한 후 제어를 printBin(5)를 호출했던 ⑬번 아래 문장인 ⑬번으로 옮긴다.

결과 101

```
③ int printBin(int a)      //a는 110이다.
{
  ④   if (a == 0 | a == 1) printf("%d", a);
      else
      {
        ⑤   printBin(a/2);
        ⑬   printf("%d", a%2);
      }
}
```

⑬ printBin(5)를 호출할 때 a는 110이었으므로 a%2의 값인 1을 출력한 후 제어를 printBin(11)을 호출했던 ④번 아래 문장인 ⑦번으로 옮긴다.

결과 1011

```
int main()
{
  ①   int x = 11;
  ②   printBin(x);
  ⑦   return 0;
}
```

⑦ main() 함수에서의 **return 0;**은 프로그램의 종료를 의미한다.

### [문제 10]

avg

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

#### 해설

```
Select 학과, avg(용돈) As 평균
From 학생
Group By 학과
Order By 학과 asc;
```

'학과', '용돈'의 평균을 표시하되, '용돈'의 평균은 '평균'으로 표시한다.  
<학생> 테이블에서 검색한다.  
'학과'를 기준으로 그룹을 지정한다.  
'학과'를 기준으로 오름차순 정렬한다.

## [문제 11]

※ 다음 중 하나를 쓰면 됩니다.

DNS, Domain Name System

## [문제 12]

123

4

567

89

## 해설

```
public class Test
{
    public static void main(String[] args) {
        ❶ String str = "1,2,3,4,,,5,6,7,,8,9";
        ❷ String[] splittest = str.split(",");

        ❸ for(int i = 0; i<splittest.length; i++)
        {
            ❹ System.out.print(splittest[i]);
            ❺ if ((i+1) % 3 == 0)
            ❻ System.out.println();
        }
    }
}
```

❶ 문자열 변수 `str`을 선언하고 1,2,3,4,,,5,6,7,,8,9로 초기화한다.

❷ 문자열 배열 `splittest`를 선언하고 `str`의 값을 ","로 구분하여 분리한 후 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
splittest	1	2	3	4			5	6	7		8	9

• **split(구분자)**: 지정된 구분자로 문자열을 구분하여 분리한다.

❸ 반복 변수 `i`가 0부터 1씩 증가하면서 `splittest` 배열의 길이인 12보다 작은 동안 ❹~❻번을 반복 수행한다.

❹ `splittest[i]`의 값을 출력한다.

❺ `(i+1)`을 3으로 나눈 나머지가 0이면 다음 문장을 수행한다.

❻ 커서를 다음 줄의 처음으로 이동한다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	splittest[i]	(i+1) % 3	출력
0	1	1	1
1	2	2	12
2	3	0	123 ← 다음 줄로 커서 이동
3	4	1	4
4		2	4
5		0	4 ← 다음 줄로 커서 이동
6	5	1	5
7	6	2	56
8	7	0	567 ← 다음 줄로 커서 이동

9		1	
10	8	2	8
11	9	0	89 ← 다음 줄로 커서 이동
12			

[문제 13]

※ 다음 중 하나를 쓰면 됩니다.

기본키, primary key

[문제 14]

HELLO\*\*\*WORLD\*\*

해설

```
public class Test
{
    public static void main(String[] args) {
        ① String str = "HELLO!@#WORLD/-";
        ② String res = str.replaceAll("[^ㄱ-ㅎㅌ-ㅣ가-힣a-zA-Z0-9,\\.]", "*");
        ③ System.out.print(res);
    }
}
```

- ① 문자열 변수 str을 선언하고, HELLO!@#WORLD/-로 초기화한다.  
② 문자열 변수 res를 선언하고, str의 !@#/- 문자를 \*로 치환한 값으로 초기화한다.

res      HELLO\*\*\*WORLD\*\*

- String.replaceAll(변환 대상, 변환할 문자): String 객체의 replaceAll() 메소드는 정규식으로 작성된 변환 대상을 변환할 문자로 치환하는 메소드이다.
- [^ㄱ-ㅎㅌ-ㅣ가-힣a-zA-Z0-9,\\.]: 대괄호[]는 범위를, ^는 제외를, ㄱ-ㅎ은 한글 자음을, ㅌ-ㅣ는 한글 모음을, 가-힣은 한글 '가'부터 '힣'까지의 글자를, a-z와 A-Z는 영문 소문자와 대문자를, 0-9는 숫자를 의미한다. 즉 해당 정규식은 한글 자음과 모음, '가'부터 '힣'까지의 글자, 영문 대·소문자, 숫자(0-9), 쉼표(.), 마침표(.)를 제외한 모든 텍스트가 변환 대상이라는 의미이고, 이 변환 대상은 두 번째 인수로 입력된 별표(\*)로 치환된다.
- HELLO!@#WORLD/-에서 한글, 영문, 숫자와 쉼표, 마침표 기호가 아닌 것은 !@#/-이므로 이 문자들을 \*로 치환한 HELLO\*\*\*WORLD\*\*을 반환한다.

- ③ res의 값을 출력한다.

결과      HELLO\*\*\*WORLD\*\*

[문제 15]

※ 다음 중 하나를 쓰면 됩니다.

다치 종속, 다중 값 종속, Multi Valued Dependency

[문제 16]

※ 다음 중 하나를 쓰면 됩니다.

블랙박스 테스트, Black Box Test

[문제 17]

Winkey + Shift + S



## [문제 18]

case

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```

① select count( case when 학과코드 = "ELE" then 1 end) As "전자공학과",
②      count( case when 학과코드 = "COM" then 1 end) As "컴퓨터공학과"
③ from 학생;

```

① '학과코드'의 값이 "ELE"일 때 반환된 1의 개수를 계산하여 표시하되, "전자공학과"라는 이름으로 표시한다.

- case when 조건 then 반환값 end : 조건이 맞으면 반환값을 반환한다.
- '학과코드' 필드의 값이 다음과 같은 경우 count 함수 적용 예

학과코드	반환값
ELE	1
ELE	1
COM	
COM	
COM	
count( ) 적용	2

② '학과코드'의 값이 "COM"일 때 반환된 1의 개수를 계산하여 표시하되, "컴퓨터공학과"라는 이름으로 표시한다.

- '학과코드' 필드의 값이 다음과 같은 경우 count 함수 적용 예

학과코드	반환값
ELE	
ELE	
COM	1
COM	1
COM	1
count( ) 적용	3

③ <학생> 테이블을 대상으로 한다.

## [문제 19]

내부 스키마

## [문제 20]

INDEX

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```

CREATE INDEX X
ON T(P, Q DESC, R);

```

인덱스(색인) <X>를 생성한다.

<T> 테이블의 'P', 'Q', 'R' 속성을 이용하여 생성한다. 'P'와 'R'은 오름차순, 'Q'는 내림차순으로 정렬한다.

※ 정렬 여부가 생략된 경우 오름차순으로 정렬됩니다.



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익의 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 굵지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

**문제 1** IPv6의 주소 표현 방식 중 유니캐스트 주소 공간을 표현하는 방식으로 옳은 것을 <보기>에서 찾아 기호(㉠~㉤)로 쓰시오. (5점)



<보기>

㉠ ::128      ㉡ ::/96      ㉢ ::ffff:0:0/96      ㉣ fc00::/7      ㉤ ff00::/8

**답 :**

**문제 2** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)



- 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위이다.
- 시스템의 성능이 향상되고 관리 효율성은 증가하지만 데이터의 일관성 및 정합성이 저하될 수 있다.

**답 :**

**문제 3** Windows 10을 사용하는 컴퓨터에서 클립보드의 내용을 확인하기 위해 사용하는 단축키를 쓰시오. (5점)



**답 :** Winkey + (      )

**문제 4** UDP(User Datagram Protocol)가 사용하는 데이터 전송 단위가 무엇인지 쓰시오. (5점)



**답 :**



**문제 5** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
import java.util.Arrays;
import java.util.StringTokenizer;

public class Test {
    public static int[] Add(int[] originalArr, int val) {
        int[] newArray = Arrays.copyOf(originalArr, originalArr.length+1);
        newArray[newArray.length-1] = val;
        return newArray;
    }

    public static void main(String[] args) {
        String test1 = "15, -41, 12, 42, -12, 2, 4";
        String test2 = test1.replaceAll("[^0-9,-]", ",");
        StringTokenizer strtoken = new StringTokenizer(test2, ",");
        int[] originalArr = new int[strtoken.countTokens()];
        int index = 0;

        while(strtoken.hasMoreTokens()) {
            originalArr[index] = Integer.parseInt(strtoken.nextToken());
            index++;
        }

        int[] newArray = Add(originalArr, 99);
        int maxNumber = Integer.MIN_VALUE;

        for(int number : newArray) {
            if(number > maxNumber) {
                maxNumber = number;
            }
        }

        System.out.print(maxNumber);
    }
}
```

**답 :**

**문제 6** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

4640206



상호 배제에 의해 나타나는 문제점으로, 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상을 의미한다.

**답 :****문제 7** 리눅스 명령어 중 다음 설명에 해당하는 알맞은 명령어를 쓰시오. (5점)

4640207



- 파일의 내용을 앞에서부터 10행까지만 표시한다.
- -n 옵션을 사용하면 지정한 숫자의 행까지만 표시한다.
- -c 옵션을 사용하면 지정한 Byte만큼만 표시한다.

**답 :****문제 8** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

4640208



```
#include <stdio.h>
int main()
{
    int n1 = 172;
    int n2 = 387;

    while( n1 != n2 )
    {
        if ( n1 > n2 ) n1-=n2;
        else n2-=n1;
    }

    printf("%d", n1);

    return 0;
}
```

**답 :**



**문제 9** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#include <stdio.h>
#include <stdbool.h>

bool numTest(int n) {
    int i = 2;
    if ( n < 2 ) return false;
    else if ( n == 2 ) return true;

    while(1)
    {
        if ( n % i == 0 ) return false;
        else if ( i * i > n ) break;
        i++;
    }
    return true;
}

int main()
{
    int j = 0;
    int res = 0;
    for (j = 1; j <= 20; j++)
    {
        if ( numTest(j) ) res+=j;
    }
    printf("%d", res);

    return 0;
}
```

**답 :**

**문제 10** 다음에 제시된 관계대수를 가장 간략한 SQL문으로 변환하시오. (5점)

4640210



$$\sigma_{\text{mode}=101}(\text{stu})$$

**답 :**

**문제 11** 다음 설명에 해당하는 알맞은 용어를 쓰시오. (5점)

4640211



데이터의 검색이나 그룹화 등의 작업 속도를 향상시키기 위해 데이터가 정렬되도록 설정하는 기능으로, 검색을 자주하는 필드에 대해 설정하는 것이 바람직하다.

**답 :**

**문제 12** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

4640212



```
import java.math.*;

public class Test
{
    public static void main(String[] args) {
        BigInteger n = new BigInteger("12345");
        BigInteger m = new BigInteger("54321");
        System.out.print(n.compareTo(m));
    }
}
```

**답 :**

4640213



**문제 13** 다음 SQL문에 대한 <보기>의 설명 중 옳은 문장의 기호(㉠~㉣)를 쓰시오. (단 SQL문에서 ‘employ’는 테이블명이고, ‘position’은 ‘직위’ 필드임) (5점)

<SQL문>

```
CREATE VIEW vi_table  
AS  
Select * from employ where position like "%부장%"
```

<보기>

- ㉠ 직위가 “부장”이 아닌 자료를 대상으로 뷰를 생성한다.
- ㉡ 직위가 “부장”으로 시작하는 자료를 대상으로 뷰를 생성한다.
- ㉢ 직위가 “부장”을 포함하는 자료를 대상으로 뷰를 생성한다.
- ㉣ 직위가 “부장”으로 끝나는 자료를 대상으로 뷰를 생성한다.

**답 :**

4640214



**문제 14** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
public class Test  
{  
    public static void main(String[] args) {  
        System.out.print( -5 >> 31 );  
    }  
}
```

**답 :**

4640215



**문제 15** 트랜잭션의 특성 4가지를 모두 쓰시오. (5점)

**답 :**



**문제 16** 다음 설명에 해당하는 검증 기준을 쓰시오. (5점)

4640216



- 화이트박스 테스트의 검증 기준 중 소스 코드의 모든 구문이 한 번 이상 수행되도록 테스트 케이스를 설계하는 방법이다.
- 다른 커버리지에 비해 가장 약하다.

**답 :**

**문제 17** IPv6 주소의 길이는 몇 비트인지 쓰시오. (5점)

4640217



**답 :**

**문제 18** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

4640218



```
#include <stdio.h>
int main()
{
    double num = 0.01;
    double res = 0;
    int cnt = 0;
    while(cnt < 100)
    {
        res += num;
        cnt++;
    }
    printf(res == 1 ? "true" : "false");

    return 0;
}
```

**답 :**

한국산업인력공단에서 시험 문제를 공개하지 않아 수험생의 기억을 토대로 대부분의 문제를 재구성하였으나, [문제 19]와 [문제 20]은 수험생의 기억을 토대로 재구성하기에 어려움이 있었습니다. 이점 양해 바랍니다.

## [문제 1]

㉔

## 해설

IPv6 유니캐스트의 주소 구분

- Global Unicast Address → 2000::/3
- Unique Local Unicast Address → FC00::/7
- Link Local Unicast Address → FE80::/64

## [문제 2]

※ 다음 중 하나를 쓰면 됩니다.

반정규화, 역정규화, Denormalization

## [문제 3]

V

## [문제 4]

※ 다음 중 하나를 쓰면 됩니다.

데이터그램, Datagram

## [문제 5]

99

## 해설

```

import java.util.Arrays;           Arrays 클래스가 정의되어 있는 라이브러리이다.
import java.util.StringTokenizer;  StringTokenizer 클래스가 정의되어 있는 라이브러리이다.

public class Test {
    ⑩ public static int[] Add(int[] originalArr, int val) {
    ⑪     int[] newArray = Arrays.copyOf(originalArr, originalArr.length+1);
    ⑫     newArray[newArray.length-1] = val;
    ⑬     return newArray;
    }

    public static void main(String[] args) {
    ①     String test1 = "15, -41, 12, 42, -12, 2, 4";
    ②     String test2 = test1.replaceAll("[^0-9,-]", ",");
    ③     StringTokenizer strtoken = new StringTokenizer(test2, ",");
    ④     int[] originalArr = new int[strtoken.countTokens()];
    ⑤     int index = 0;

    ⑥     while(strtoken.hasMoreTokens()) {
    ⑦         originalArr[index] = Integer.parseInt(strtoken.nextToken());
    ⑧         index++;
    }
    }

```

```

914    int[] newArray = Add(originalArr, 99);
15    int maxNumber = Integer.MIN_VALUE;

16    for(int number : newArray) {
17        if(number > maxNumber) {
18            maxNumber = number;
19        }
20    }

19    System.out.print(maxNumber);

    }
}

```

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

- 문자열 변수 test1을 선언하고, 15, -41, 12, 42, -12, 2, 4로 초기화한다.
- 문자열 변수 test2를 선언하고, test1의 공백을 쉼표로 치환한 값으로 초기화한다.

test2    15,-41,12,42,-12,2,4

- String.replaceAll(변환 대상, 변환할 문자)**: String 객체의 replaceAll() 메소드는 정규식으로 작성된 변환 대상을 변환할 문자로 치환하는 메소드이다.
- [^0-9,-]**: 대괄호[]는 범위를, ^는 제외를, 0-9는 숫자를 의미한다. 즉 해당 정규식은 숫자(0-9)와 쉼표(.), 마이너스 기호(-)를 제외한 모든 텍스트가 변환 대상이라는 의미이고, 이 변환 대상은 두 번째 인수로 입력된 쉼표(,)로 치환된다.
- test1의 15, -41, 12, 42, -12, 2, 4에서 숫자와 쉼표, 마이너스 기호가 아닌 것은 공백뿐이므로 모든 공백을 쉼표로 치환한 15,-41,12,42,-12,2,4를 반환한다.

- StringTokenizer 클래스의 객체 변수 strtoken을 선언하고, test2에 저장된 문자열을 쉼표(,)로 분리하여 저장한다.

- StringTokenizer(문자열, 구분자)**: StringTokenizer 클래스는 생성자 호출 시 인수로 입력된 문자열을 지정한 구분자로 분리한 후 분리된 문자열을 하나의 토큰(Token)으로 취급하여 처리하는 클래스이다(빈 문자열은 토큰으로 인식하지 않는다).

토큰1   토큰2   토큰3   토큰4   토큰5   토큰6   토큰7

strtoken	15	-41	12	42	-12	2	4
----------	----	-----	----	----	-----	---	---

- strtoken의 토큰 개수인 7에 해당하는 정수형 배열 originalArr을 선언한다.

- StringTokenizer.countTokens()**: StringTokenizer 객체의 토큰 개수를 반환하는 메소드이다.

[0]   [1]   [2]   [3]   [4]   [5]   [6]

originalArr	0	0	0	0	0	0	0
-------------	---	---	---	---	---	---	---

※ Java는 배열 선언 시 초기값을 지정하지 않으면 배열의 모든 요소에 자동으로 0이 저장됩니다.

- 정수형 변수 index를 선언하고 0으로 초기화한다.

- strtoken에 반환할 토큰이 있으면 7, 8번을 반복 수행하고, 더 이상 반환할 토큰이 없으면 9번으로 이동한다.

- StringTokenizer.hasMoreTokens()**: StringTokenizer 객체에 반환할 토큰이 있으면 1(참)을, 토큰이 없으면 0(거짓)을 반환하는 메소드이다.

- 토큰을 정수로 변환하여 originalArr[index]에 저장한다.

- StringTokenizer.nextToken()**: StringTokenizer 객체에서 차례대로 토큰을 가져오는 메소드이다.

- Integer.parseInt()**: 숫자로 된 문자열을 정수로 변환하는 메소드이다.

- index = index + 1과 동일하다. index의 값을 1씩 누적시킨다. 6~8번을 반복 수행하면 originalArr은 다음의 값을 갖는다.

[0]   [1]   [2]   [3]   [4]   [5]   [6]

originalArr	15	-41	12	42	-12	2	4
-------------	----	-----	----	----	-----	---	---

- 정수형 변수 newArray를 선언한 후 originalArr의 시작 주소와 99를 인수로 Add() 메소드를 호출하고 돌려받은 값을 newArray에 저장한다.

- 정수형 배열을 반환하는 Add() 메소드의 시작점이다. 9번에서 전달받은 originalArr의 시작 주소와 99를 originalArr과 val이 받는다.

[0]   [1]   [2]   [3]   [4]   [5]   [6]

Add.originalArr	15	-41	12	42	-12	2	4
-----------------	----	-----	----	----	-----	---	---

Add.val    99

※ main() 메소드와 Add() 메소드에서 같은 이름을 가진 변수 및 객체들의 구분을 위해서 Add() 메소드의 변수 및 객체명에만 메소드명을 표기했습니다.

⑪ originalArr에서 originalArr의 길이에 1을 더한 8만큼의 요소를 복사하여 새로운 정수형 배열 newArray를 선언한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Add.newArray	15	-41	12	42	-12	2	4	0

• **length**: length는 배열 클래스의 속성으로, 배열 요소의 개수가 저장되어 있다.

• **Arrays.copyOf(배열, 요소의 개수)**: 인수로 입력된 배열을 요소의 개수만큼 복사하여 반환하는 메소드이다. 복사할 요소의 개수가 배열의 전체 요소의 개수보다 크면 초과된 요소는 0으로 초기화되어 복사된다.

⑫ newArray.length는 80이므로 newArray[7]에 val의 값 99를 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Add.newArray	15	-41	12	42	-12	2	4	99

⑬ Add.newArray의 주소를 함수를 호출했던 ⑩번으로 반환한다.

⑭ ⑬번에서 돌려받은 주소를 newArray에 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
newArray	15	-41	12	42	-12	2	4	99

⑮ 정수형 변수 maxNumber를 선언하고  $-2^{31}$ (-2,147,483,648)로 초기화한다.

• **Integer.MIN\_VALUE**: 정수 자료가 가지는 최소값인  $-2^{31}$ (-2,147,483,648)이 저장되어 있는 Integer 클래스의 속성이다.

⑯ newArray의 각 요소를 정수형 변수 number에 저장하면서 ⑰, ⑱번을 반복 수행한다. newArray는 8개의 요소를 가지므로 ⑰, ⑱번은 총 8회 수행된다.

⑰ number가 maxNumber보다 크면 ⑲번으로 이동하고 아니면 반복문의 처음인 ⑱번으로 이동한다.

⑲ maxNumber에 number의 값을 저장한다. ⑲~⑲번의 반복문 실행에 따른 변수들의 변화는 다음과 같다.

maxNumber	number	number>maxNumber
-2,147,483,648	15	Yes
15	-41	No
42	12	No
99	42	Yes
	-12	No
	2	No
	4	No
	99	Yes

⑳ maxNumber의 값 99를 출력한다.

결과 99

[문제 6]

※ 다음 중 하나를 쓰면 됩니다.

교착 상태, DeadLock

[문제 7]

head

## [문제 8]

43

## 해설

```
#include <stdio.h>
int main()
{
  ❶ int n1 = 172;
  ❷ int n2 = 387;

  ❸ while( n1 != n2 )
  {
    ❹ if ( n1 > n2 ) n1 -= n2;
    ❺ else n2 -= n1;
  }

  ❻ printf("%d", n1);

  ❼ return 0;
}
```

❶ 정수형 변수 n1을 선언하고 172로 초기화한다.

❷ 정수형 변수 n2를 선언하고 387로 초기화한다.

❸ n1과 n2가 같지 않은 동안 ❹, ❺번을 반복 수행한다.

❹ n1이 n2보다 크면 'n1 -= n2;', 즉 'n1 = n1 - n2;'를 수행하고, 아니면 ❺번으로 이동한다.

❺ ❹번 조건이 거짓일 경우 실행할 문장으로, n1이 n2보다 작거나 같으면 'n2 -= n1;', 즉 'n2 = n2 - n1;'을 수행한다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

n1	n2	n1\ n2
172	387	No
	215	No
	43	Yes
129		Yes
86		Yes
43		

❻ n1의 값 43을 정수로 출력한다.

결과 43

❼ 프로그램을 종료한다.

## [문제 9]

77

## 해설

```

#include <stdio.h>
#include <stdbool.h>

5  bool numTest(int n) {
6      int i = 2;
7      if ( n < 2 ) return false;
9      else if ( n == 2 ) return true;

10     while(1)
        {
11         if ( n % i == 0 ) return false;
12         else if ( i * i > n ) break;
14         i++;
        }
13     return true;
    }

    int main()
    {
1  int j = 0;
2  int res = 0;
3  for ( j = 1; j <= 20; j++)
    {
4 8      if ( numTest(j) ) res += j;
    }
15     printf("%d", res);

16     return 0;
    }

```

모든 C언어 프로그램은 반드시 main() 함수에서 시작한다.

- ① 정수형 변수 j를 선언하고 0으로 초기화한다.
- ② 정수형 변수 res를 선언하고 0으로 초기화한다.
- ③ 반복 변수 j가 1부터 1씩 증가하면서 20보다 작거나 같은 동안 ④번을 반복 수행한다.

## 첫 번째 반복

- ④ j의 값 1을 인수로 numTest() 함수를 호출한 결과가 참(true)이면 res에 j의 값을 누적시킨다.
- ⑤ 논리값을 반환하는 numTest() 함수의 시작점이다. ④번에서 전달받은 1을 n이 받는다.
- ⑥ 정수형 변수 i를 선언하고 2로 초기화한다.
- ⑦ n은 1이므로 2보다 작다. 조건을 만족하므로 논리값 false(거짓)를 반환하면서 ⑧번으로 이동한다.
- ⑧ ⑦번에서 돌려받은 값이 false(거짓)이므로 'res += j;'를 수행하지 않고 ③번으로 돌아가 j의 값을 1 증가시킨다.

## 두 번째 반복

- ④ j의 값 2를 인수로 numTest() 함수를 호출한 결과가 참(true)이면 res에 j의 값을 누적시킨다.
- ⑤ 논리값을 반환하는 numTest() 함수의 시작점이다. ④번에서 전달받은 2를 n이 받는다.

- ⑥ 정수형 변수 `i`를 선언하고 2로 초기화한다.
- ⑦ `n`은 20이므로 2보다 작지 않다. 조건을 만족하지 않으므로 ⑨번으로 이동한다.
- ⑨ `n`은 20이므로 조건을 만족한다. 논리값 `true`(참)을 반환하면서 함수를 호출했던 ⑥번으로 이동한다.
- ⑧ ⑨번에서 돌려받은 값이 `true`(참)이므로 `res`에 `j`를 누적한 후 ⑩번으로 돌아가 `j`의 값을 1 증가시킨다.  
`res`에는 2가 저장된다.

#### 세 번째 반복

- ④ `j`의 값 3을 인수로 `numTest()` 함수를 호출한 결과가 참(`true`)이면 `res`에 `j`의 값을 누적시킨다.
- ⑤ 논리값을 반환하는 `numTest()` 함수의 시작점이다. ④번에서 전달받은 3을 `n`이 받는다.
- ⑥ 정수형 변수 `i`를 선언하고 2로 초기화한다.
- ⑦ `n`은 30이므로 ⑦번과 ⑨번 조건을 만족하지 않아 ⑩번으로 이동한다.
- ⑩ 조건이 참(1)이므로 `break` 또는 `return`을 만나 반복문을 탈출할 때까지 ⑪~⑭번을 무한 반복한다.
- ⑪ `n`의 값 3을 `i`의 값 2로 나눈 나머지가 10이므로 조건을 만족하지 않아 ⑫번으로 이동한다.
- ⑫ `i`의 값 2를 두 번 곱한 값 4는 `n`의 값 3보다 크므로 `break` 명령을 수행한다. 반복문을 탈출하여 ⑬번으로 이동한다.
- ⑬ 논리값 `true`(참)을 반환하면서 함수를 호출했던 ⑧번으로 이동한다.
- ⑧ ⑬번에서 돌려받은 값이 `true`(참)이므로 `res`에 `j`를 누적한 후 ⑩번으로 돌아가 `j`의 값을 1 증가시킨다.  
`res`에는 5가 저장된다.

#### 네 번째 반복

- ④ `j`의 값 4를 인수로 `numTest()` 함수를 호출한 결과가 참(`true`)이면 `res`에 `j`의 값을 누적시킨다.
- ⑤ 논리값을 반환하는 `numTest()` 함수의 시작점이다. ④번에서 전달받은 4를 `n`이 받는다.
- ⑥ 정수형 변수 `i`를 선언하고 2로 초기화한다.
- ⑦ `n`은 40이므로 ⑦번과 ⑨번 조건을 만족하지 않아 ⑩번으로 이동한다.
- ⑩ 조건이 참(1)이므로 `break` 또는 `return`을 만나 반복문을 탈출할 때까지 ⑪~⑭번을 무한 반복한다.
- ⑪ `n`의 값 4를 `i`의 값 2로 나눈 나머지가 00이므로 논리값 `false`(거짓)를 반환하면서 함수를 호출했던 ⑧번으로 이동한다.
- ⑧ ⑪번에서 돌려받은 값이 `false`(거짓)이므로 '`res += j`;'를 수행하지 않고 ⑩번으로 돌아가 `j`의 값을 1 증가시킨다.

#### 다섯 번째 반복

- ④ `j`의 값 5를 인수로 `numTest()` 함수를 호출한 결과가 참(`true`)이면 `res`에 `j`의 값을 누적시킨다.
- ⑤ 논리값을 반환하는 `numTest()` 함수의 시작점이다. ④번에서 전달받은 5를 `n`이 받는다.
- ⑥ 정수형 변수 `i`를 선언하고 2로 초기화한다.
- ⑦ `n`은 50이므로 ⑦번과 ⑨번 조건을 만족하지 않아 ⑩번으로 이동한다.
- ⑩ 조건이 참(1)이므로 `break` 또는 `return`을 만나 반복문을 탈출할 때까지 ⑪~⑭번을 무한 반복한다.
- ⑪ `n`의 값 5를 `i`의 값 2로 나눈 나머지가 10이므로 조건을 만족하지 않아 ⑫번으로 이동한다.
- ⑫ `i`의 값 2를 두 번 곱한 값 4는 `n`의 값 5보다 작으므로 ⑭번으로 이동한다.
- ⑭ '`i = i + 1`;'과 동일하다. `i`의 값을 1 증가시킨 후 ⑪번으로 이동한다. `i`에는 3이 저장된다.
- ⑪ `n`의 값 5를 `i`의 값 3으로 나눈 나머지가 20이므로 조건을 만족하지 않아 ⑫번으로 이동한다.
- ⑫ `i`의 값 3을 두 번 곱한 값 9는 `n`의 값 5보다 크므로 `break` 명령을 수행한다. 반복문을 탈출하여 ⑬번으로 이동한다.
- ⑬ 논리값 `true`(참)을 반환하면서 함수를 호출했던 ⑧번으로 이동한다.
- ⑧ ⑬번에서 돌려받은 값이 `true`(참)이므로 `res`에 `j`의 값을 누적시킨 후 ⑩번으로 돌아가 `j`의 값을 1 증가시킨다.  
`res`에는 100이 저장된다.

위의 과정을 통해 다음 사항들을 알 수 있다.

- `numTest()` 함수는 인수가 2보다 작으면 `false`를, 2면 `true`를 반환하고 인수가 3 이상인 경우 인수를 2부터 1씩 증가하면서 나눴을 때 나머지가 00이 아니면 `true`를 반환하는 것으로 보아 소수를 찾는 함수임을 알 수 있다.
- ⑤번에서 `numTest(j)`가 `true`(참)라는 것은 소수임을 의미하므로, `res`에는 1부터 20까지의 숫자 중에서 소수에 해당하는 값이 누적된다.

• 자세한 값의 변화는 다음 표를 통해 확인하자.

main( ) 함수		numTest( ) 함수			
j	res	n	i	n % i	반환값
0	0	1	2		false
1					
2	2	2	2		true
3	5	3	2	1	true
4		4	2	0	false
5	10	5	2	1	true
			3	2	
6		6	2	0	false
7	17	7	2	1	true
			3		
:	:	:	:	:	:
11	28	11	2	1	true
			3	2	
			4	3	
:	:	:	:	:	:
13	41	13	2	1	true
			3	1	
			4	1	
:	:	:	:	:	:
17	58	17	2	1	true
			3	2	
			4	1	
			5	2	
:	:	:	:	:	:
19	77	19	2	1	true
			3	1	
			4	3	
			5	4	
20		20	2	0	false
21					

㉓ res의 값 77을 정수로 출력한다.

결과 77

㉔ 프로그램을 종료한다.

[문제 10]

select \* from stu where mcode=101;

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대·소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

해설

- 표시할 필드에 대한 조건이 없으므로 모든 필드를 대상으로 검색하면 됩니다. select \*입니다.
- σ<sub>mcode=101</sub> : 'mcode'가 101인 자료만을 대상으로 검색하므로 where mcode = 101입니다.
- (stu) : <stu> 테이블의 자료를 검색하므로 from stu입니다.



## [문제 11]

※ 다음 중 하나를 쓰면 됩니다.

인덱스, Index

## [문제 12]

-1

## 해설

```
import java.math.*;           BigInteger 클래스가 정의되어 있는 라이브러리이다.

public class Test
{
    public static void main(String[] args) {
        ❶    BigInteger n = new BigInteger("12345");
        ❷    BigInteger m = new BigInteger("54321");
        ❸    System.out.print(n.compareTo(m));
    }
}
```

❶ BigInteger 클래스의 객체 변수 n을 선언하고, 문자열 12345를 정수로 변환하여 n에 저장한다.

• **BigInteger**: BigInteger 클래스는 long 형을 벗어나는 큰 정수를 다룰 수 있게 하는 클래스로, 생성자 호출 시 숫자로 된 문자열을 인수로 입력 받아 정수로 저장한다.

❷ BigInteger 클래스의 객체 변수 m을 선언하고, 문자열 54321을 정수로 변환하여 m에 저장한다.

❸ n보다 m이 크므로 -1을 출력한다.

• **n.compareTo(m)**: n과 m을 비교하여 같으면 0, m이 크면 -1, n이 크면 1을 반환하는 메소드이다.

결과 -1

## [문제 13]

㉠

## 해설

CREATE VIEW vi_table	생성한 뷰의 이름은 <vi_table>이다.
AS	
Select *	테이블의 모든 속성을 가져온다.
from employ	<employ> 테이블에서 속성을 가져와 뷰를 생성한다.
where position like "%부장%"	'position', 즉 '직위' 필드가 "부장"을 포함하는 자료만을 뷰로 생성한다.

[문제 14]

-1

해설

```
public class Test
{
    public static void main(String[] args) {
        ❶ System.out.print( -5 >> 31 );
    }
}
```

❶ >>는 오른쪽 시프트 연산자이므로, -5를 오른쪽으로 31비트 이동시킨 값을 출력한다.

- -5를 4Byte 2진수로 표현하면 다음과 같다.

	32	31	30	29	...	9	8	7	6	5	4	3	2	1
-5	1	1	1	1	...	1	1	1	1	1	1	0	1	1
					...		$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
부호 비트					...		128	64	32	16	8	4	2	1

- 부호를 제외한 전체 비트를 오른쪽으로 31비트 이동시킨다. 음수이므로 패딩 비트(빈자리)에는 1이 채워지므로, 모든 비트가 1로 채워지게 된다.

	32	31	30	29	...	9	8	7	6	5	4	3	2	1
-1	1	1	1	1	...	1	1	1	1	1	1	1	1	1
					...		$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
부호 비트					...		128	64	32	16	8	4	2	1

패딩 비트

- 이동된 값을 10진수로 변환하면 -1이다.

결과 -1

[문제 15]

※ 한글이나 영문 중 하나를 쓰면 됩니다.

- 원자성, 일관성, 독립성, 영속성
- Atomicity, Consistency, Isolation, Durability

[문제 16]

※ 다음 중 하나를 쓰면 됩니다.

문장 검증 기준, Statement Coverage

[문제 17]

128

## [문제 18]

false

## 해설

```
#include <stdio.h>
int main()
{
  ❶ double num = 0.01;
  ❷ double res = 0;
  ❸ int cnt = 0;
  ❹ while(cnt < 100)
  {
    ❺ res += num;
    ❻ cnt++;
  }
  ❼ printf(res == 1 ? "true" : "false");

  ❽ return 0;
}
```

- ❶ 실수형 변수 num을 선언하고 0.01로 초기화한다.
- ❷ 실수형 변수 res를 선언하고 0으로 초기화한다.
- ❸ 정수형 변수 cnt를 선언하고 0으로 초기화한다.
- ❹ cnt가 100보다 작은 동안 ❺, ❻번을 반복 수행한다.
- ❺ 'res = res + num;'과 동일하다. res에 num을 누적시킨다.
- ❻ 'cnt = cnt + 1;'과 동일하다. cnt의 값을 1 증가시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

num	res	cnt
0.01	0	0
	0.01	1
	0.02	2
	0.03	3
	⋮	⋮
	0.99	99
	1.00	100

- ❼ res의 값이 1이면 true를 출력하고, 아니면 false를 출력한다. res의 값이 1이지만 double 형이고 비교되는 1은 int형이므로 자료형이 서로 달라 false가 출력된다.

결과    **false**



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

**문제 1** 다음은 <student> 테이블의 'S\_NAME' 속성에 고유한 특성을 갖는 'idx\_index'라는 이름의 인덱스를 정의하는 SQL문이다. 괄호에 들어갈 알맞은 명령어를 쓰시오. (5점)

4640101



<student>

S_ID	S_NO	S_NAME	M_ID
1	2019AA01	김호박	2
2	2020BB02	박나물	2
3	2021CC03	최가지	4
4	2022DD04	이무침	4

<SQL>

CREATE (            ) INDEX idx\_index ON student(S\_NAME);

답 :

**문제 2** DDL(Data Define Language)에 속하는 명령어를 <보기>에서 골라 기호(㉠~㉤)로 쓰시오. (5점)

4640102



<보기>

㉠ SELECT            ㉡ INSERT            ㉢ CREATE            ㉣ UPDATE

답 :

**문제 3** 키보드로 명령어를 직접 입력하지 않고, 마우스로 아이콘이나 메뉴를 선택하여 모든 작업을 수행하는 방식의 사용자 인터페이스를 무엇이라고 하는지 영문 약어로 쓰시오. (5점)

4640103



답 :



**문제 4** 다음 <회원>, <테이프>, <대여> 테이블을 참고하여 <SQL>의 실행 결과를 쓰시오. (5점)

<SQL>

```
SELECT 회원.성명, 회원.전화번호
FROM 회원, 대여
WHERE 회원.회원번호 = 대여.회원번호 AND 대여.테이프번호 = 'T3';
```

<회원>

회원번호	성명	전화번호
S1	이동국	111-1111
S2	홍길동	222-2222
S3	이동국	333-3333
S4	박찬성	444-4444
S5	성춘향	555-5555

<테이프>

테이프번호	제목
T1	금강산도 식후경
T2	니사랑 내사랑
T3	오늘도 저 끝까지
T4	파라노말 디렉티브
T5	앱솔루트 커맨드

<대여>

회원번호	성명	테이프번호
S1	이동국	T2
S1	이동국	T3
S2	홍길동	T4
S2	홍길동	T5
S3	이동국	T1
S3	이동국	T3
S4	박찬성	T3
S4	박찬성	T4
S5	성춘향	T5
S5	성춘향	T6

**답 :**

4640105



**문제 5** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
public class Test {
    public static void main(String[] args) {
        Otest ot = new Otest();
        ot.cat();
        ot.cat("4");
    }
}

class Otest {
    void cat() {
        System.out.print("1234");
    }
    void cat(int c) {
        System.out.print(++c);
    }
    void cat(String c) {
        System.out.print("문자");
    }
}
```

**답 :**

4640106



**문제 6** TCP/IP의 네트워크 관리 프로토콜로, 라우터나 허브 등 네트워크 기기의 네트워크 정보를 네트워크 관리 시스템에 보내는데 사용되는 표준 통신 규약을 <보기>에서 찾아 기호(㉠~㉤)로 쓰시오. (5점)

<보기>

㉠ TCP

㉡ HTTP

㉢ SGMP

㉤ SNMP

㉥ UDP

**답 :**

4440107



**문제 7** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
#include <stdio.h>
#define N 100
main() {
    int i = 1;
    int cnt = 0;

    while (i <= N) {
        if ((i % 3) == 0 && (i % 7) == 0) {
            cnt++;
            printf("%d*%d*", cnt, i);
        }
        i++;
    }
}
```

**답 :**

4440108



**문제 8** Windows 10/11에서 윈도우 화면을 잠그도록 하는 단축키를 쓰시오. (5점)

**답 :** Winkey + (       )

4440109



**문제 9** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
public class Test {
    public static void main(String[] args) {
        int a = 1;
        int b = a++ > 1 ? a+2 : a+3;
        System.out.print(b);
    }
}
```

**답 :**



**문제 10** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 실행 플랫폼은 데이터 모델 LLP 64 또는 LP 64 기준임) (5점)

4640110



```
#include <stdio.h>

main() {
    int a = sizeof(int) + sizeof(char);
    printf("%d", a);
}
```

**답 :**

**문제 11** OSI 7계층 중 다음에서 설명하는 특징을 갖는 계층이 무엇인지 쓰시오. (5점)

4640111



- 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 시스템 간 연결 설정과 유지 및 종료를 담당한다.
- 송신 측과 수신 측의 속도 차이 해결을 위한 흐름 제어 기능을 한다.
- 프레임의 시작과 끝을 구분하기 위한 프레임의 동기화 기능을 한다.
- 오류의 검출과 회복을 위한 오류 제어 기능을 한다.
- 프레임의 순서적 전송을 위한 순서 제어 기능을 한다.

**답 :**

**문제 12** 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic) 값들의 집합으로, 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는데에도 이용하는 것은 무엇인지 쓰시오. (5점)

4640112



**답 :**

**문제 13** 다음에서 설명하는 특징을 갖는 스키마는 무엇인지 쓰시오. (5점)

- 물리적 저장장치의 입장에서 본 데이터베이스 구조이다.
- 실제로 저장될 레코드의 형식, 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 등을 나타낸다.



**답 :**

**문제 14** UNIX 명령어 중 화면에 파일 내용을 출력하는 명령어를 <보기>에서 찾아 기호(㉠~㉥)로 쓰시오. (5점)

<보기>

㉠ ls                      ㉡ fcsk                      ㉢ pwd                      ㉣ cat                      ㉤ chmod



**답 :**

**문제 15** RSH, RCP, RLOGIN, REXEC 및 TELNET, FTP 서비스 등을 대체하기 위한 네트워크 보안 도구 중 하나로, 기본적으로 22번 포트를 사용하며 다른 네트워크상의 컴퓨터에 원격 접속하거나 파일을 복사할 수 있게 해주는 응용 프로그램 또는 프로토콜을 가리키는 용어를 쓰시오. (5점)



**답 :**

**문제 16** 데이터베이스에서 다른 테이블의 기본키를 참조하는 속성 또는 속성들의 집합을 무엇이라고 하는지 쓰시오. (5점)



**답 :**

**문제 17** 다음은 언어 번역 과정이다. ㉠에 들어갈 알맞은 용어를 쓰시오. (5점)

원시 프로그램 → ( ㉠ ) → 목적 프로그램 → 링커 → 실행 모듈



**답 :**

**문제 18** 다음 C 언어로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

4640118



```
#include <stdio.h>
#include <math.h>

int main() {
    int ans = pow(2, ceil(M_PI));
    printf("%d", ans);
}
```

답 :

**문제 19** 다음 Java로 구현한 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

4640119



```
public class Test {
    public static void main(String[] args) {
        String weeks[] = {"월요일", "화요일", "수요일", "목요일", "금요일",
                           "토요일", "일요일"};

        int cnt = 1;
        for ( int i = 1; i < 7; i++) {
            if (i%3 == 0)
                break;
            cnt++;
        }
        System.out.print("오늘은 " + weeks[cnt]);
    }
}
```

답 :



**문제 20** 다음은 SELECT문의 실행 순서를 나열한 것이다. <보기>에 제시된 명령어를 실행 순서에 맞게 기호(㉠~㉣)로 나열하시오. (5점)

FROM → (        ) → (        ) → (        ) → SELECT → ORDER BY

<보기>

㉠ GROUP BY      ㉡ HAVING      ㉢ WHERE

**답 :**

## [문제 1]

UNIQUE

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
CREATE UNIQUE INDEX idx_index
ON student (S_NAME);
```

중복 값이 없는 고유한 특성을 갖는 'idx\_index'라는 이름의 인덱스를 생성한다.

〈student〉 테이블의 'S\_NAME' 속성을 대상으로 한다.

## [문제 2]

㉔

## [문제 3]

GUI

## [문제 4]

이동국 111-1111

이동국 333-3333

박찬성 444-4444

※ 답안 작성 시 주의 사항 : 출력 형태에 대한 명확한 조건이 제시되어 있지 않은 문제입니다. 어떤 방식으로 SQL문의 실행 결과를 가져 오냐에 따라 출력 형태가 다양하기 때문에 정답에서 제시한 결과만 정확하게 포함되어 있다면 정답이라고 보아도 좋습니다.

## 해설

```
SELECT 회원.성명, 회원.전화번호
FROM 회원, 대여
WHERE 회원.회원번호 = 대여.회원번호
AND 대여.테이프번호 = 'T3';
```

〈회원〉 테이블의 '성명', '전화번호'를 표시한다.

〈회원〉, 〈대여〉 테이블을 대상으로 검색한다.

〈회원〉 테이블의 '회원번호'와 〈대여〉 테이블의 '회원번호'가 같고,

〈대여〉 테이블의 '테이프번호'가 "T3"인 튜플만을 대상으로 한다.

[문제 5]

1234문자

※ **답안 작성 시 주의 사항**: 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값 사이에 줄을 나눠 1234로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 처리됩니다.

해설

```
public class Test {
    public static void main(String[] args) {
        ❶ Otest ot = new Otest();
        ❷ ot.cat();
        ❸ ot.cat("4");
    } ❹
}
class Otest {
    ❺ void cat() {
        ❻ System.out.print("1234");
    }
    ❼ void cat(int c) {
        System.out.print(++c);
    }
    ❽ void cat(string c) {
        ❹ System.out.print("문자");
    }
}
```

Ⓐ 반환값 없는 메소드 cat()를 정의한다.

Ⓑ 반환값 없는 메소드 cat(int c)를 정의한다.

Ⓒ 반환값 없는 메소드 cat(string c)를 정의한다.

※ Ⓐ, Ⓑ, Ⓒ가 메소드 이름은 같지만 '인수를 받는 자료형'이 다르므로 서로 다른 메소드입니다. 즉 cat()와 cat(int c), cat(string c)는 다른 메소드라는 것입니다. 이렇게 이름은 같지만 인수를 받는 자료형을 달리하여 여러 기능을 정의하는 것을 오버로딩(Overloading)이라고 합니다.

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

❶ Otest 클래스의 객체 변수 ot를 선언한다.

❷ ot.cat() 메소드를 호출한다. cat()에 인수가 없으므로 ❸번으로 이동한다.

❸ 반환값이 없는 cat() 메소드의 시작점이다.

❹ 1234를 출력한다.

결과 1234

❺ 숫자를 큰따옴표로 묶으면 문자로 인식한다. 즉 문자 4를 인수로 하여 ot.cat() 메소드를 호출한다. 전달되는 인수의 자료형이 문자형이므로 ❽번으로 이동한다.

❽ 반환값이 없는 cat(string c) 메소드의 시작점이다.

❹ 문자를 출력한다.

결과 1234문자

## [문제 6]

㉠

## [문제 7]

1\*21\*2\*42\*3\*63\*4\*84\*

## 해설

```
#include <stdio.h>
#define N 100          숫자 100을 N으로 정의한다. 프로그램 안에서 N은 100과 동일하게 쓰인다.

main() {
    ❶ int i = 1;
    ❷ int cnt = 0;

    ❸ while ( i <= N ) {
    ❹     if ( (i%3) == 0 && (i%7) == 0 ) {
    ❺         cnt++;
    ❻         printf("%d*%d*", cnt, i);
    ❼     }
    ❽     i++;
    }
}
```

- ❶ 정수형 변수 *i*를 선언하고 1로 초기화한다.
- ❷ 정수형 변수 *cnt*를 선언하고 0으로 초기화한다.
- ❸ *i*가 *N*, 즉 100보다 작거나 같은 동안 ❹~❽번을 반복 수행한다.
- ❹ *i*를 3과 7로 나눈 나머지가 모두 0이면 ❺ ❻번을 수행하고, 아니면 ❼번으로 이동한다.
- ❺ 'cnt = cnt + 1;'과 동일하다. cnt의 값에 1을 누적시킨다.
- ❻ cnt의 값을 정수로 출력하고 \*을 출력한 다음 *i*의 값을 정수로 출력하고 \*을 출력한다.
- ❼ 'i = i + 1;'과 동일하다. *i*의 값에 1을 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	i%3	i%7	cnt	출력
1	1	1	0	
2	2	2		
3	0	3		
4	1	4		
5	2	5		
6	0	6		
7	1	0		
8	2	1	1	1*21*
9	0	2		
⋮	⋮	⋮		
21	0	0		
22	1	1		
⋮	⋮	⋮		

42	0	0	2	1*21*2*42*
43	1	1		
⋮	⋮	⋮		
63	0	0	3	1*21*2*42*3*63*
64	1	1		
⋮	⋮	⋮		
84	0	0	4	1*21*2*42*3*63*4*84*
85	1	1		
⋮	⋮	⋮		
100	1	2		

[문제 8]

L

[문제 9]

5

해설

```
public class Test {
    public static void main(String[] args) {
        ❶ int a = 1;
        ❷ int b = a++ > 1 ? a+2 : a+3;
        ❸ System.out.print(b);
    }
}
```

❶ 정수형 변수 a를 선언하고 1로 초기화한다.

❷ int b = a++ > 1 ? a+2 : a+3;

Ⓐ

Ⓑ

• Ⓐ: a의 값 1은 1보다 크지 않으므로 정수형 변수 b에 a+3을 저장한다.

• Ⓑ: a++은 후치 증가 연산자이므로, a++ > 1을 수행한 후 a의 값을 1 증가시켜 2가 된다. 그러므로 b에는 5가 저장된다.

❸ b의 값 5를 출력한다.

결과 5



## [문제 10]

5

## 해설

```
#include <stdio.h>

main() {
  ❶ int a = sizeof(int) + sizeof(char);
  ❷ printf("%d", a);
}
```

❶ int a = sizeof(int) + sizeof(char);

Ⓐ

Ⓑ

Ⓒ

- Ⓐ : 데이터 모델 LLP 64 또는 LP 64 기준에서는 int(정수형)의 크기가 4Byte이다.
- Ⓑ : 데이터 모델 LLP 64 또는 LP 64 기준에서는 char(문자형)의 크기가 1Byte이다.
- Ⓒ : 정수형 변수 a에 Byte 단위로 산출한 int(정수형)의 크기와 char(문자형)의 크기를 더한 값인 5를 저장한다.

❷ a의 값 5를 정수로 출력한다.

결과

5

## [문제 11]

※ 다음 중 하나를 쓰면 됩니다.

데이터 링크 계층, Data Link Layer

## [문제 12]

도메인, Domain

## [문제 13]

내부 스키마

## [문제 14]

㉠

## [문제 15]

※ 다음 중 하나를 쓰면 됩니다.

SSH, Secure Shell

## [문제 16]

※ 다음 중 하나를 쓰면 됩니다.

외래키, Foreign Key

## [문제 17]

컴파일러

[문제 18]

16

해설

```
#include <stdio.h>
#include <math.h>

int main() {
  ❶ int ans = pow(2, ceil(M_PI));
  ❷ printf("%d", ans);
}
```

- ❶ 정수형 변수 ans를 선언하고 16으로 초기화한다.

**int ans = pow(2, ceil(M\_PI));**



- Ⓐ 원주율을 정의한 상수로, 3.14159265358979323846임
- Ⓑ **ceil(3.1415...)**: 소수점을 올림하여 정수로 표현함 → 4
- Ⓒ **pow(2, 4)**: 2의 4승 → 16

- ❷ ans의 값 16을 정수로 출력한다.

결과 16

[문제 19]

오늘은 목요일

해설

```
public class Test {
  public static void main(String[] args) {
    ❶ String weeks[] = {"월요일", "화요일", "수요일", "목요일", "금요일", "토요일", "일요일"};
    ❷ int cnt = 1;
    ❸ for ( int i = 1; i < 7; i++) {
    ❹     if (i%3 == 0)
    ❺         break;
    ❻         cnt++;
    }
    ❼ System.out.print("오늘은 " + weeks[cnt]);
  }
}
```

- ❶ 7개의 요소를 갖는 문자열 배열 weeks를 선언하고 초기화한다. 개수를 지정하지 않으면, 초기값으로 지정된 수만큼 배열의 요소가 만들어진다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
weeks	월요일	화요일	수요일	목요일	금요일	토요일	일요일

- ❷ 정수형 변수 cnt를 선언하고 1로 초기화한다.
- ❸ 반복 변수 i가 1부터 1씩 증가하면서 7보다 작은 동안 ❹~❻번을 반복 수행한다.
- ❹ i를 3으로 나눈 나머지가 0이면 ❺번을 수행하고 아니면 ❻번으로 이동한다.
- ❺ for문을 벗어나 ❼번으로 이동한다.

⑥ cnt의 값을 1씩 누적시킨다.

⑦ 오늘은을 출력하고 한 칸을 띄운 후 weeks[cnt]의 값을 출력한다.

반복문 수행에 따른 변수들의 변화는 다음과 같다.

cnt	i	i%3	출력
1	1	1	오늘은 목요일
2	2	2	
3	3	0	

[문제 20]

㉠, ㉡, ㉢



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

**문제 1** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

1641201



```
public class Test {
    static int cycle(int[] arr, int i) {
        if (i <= 0) return 0;
        return arr[i] % 3 + cycle(arr, i - 1);
    }
    public static void main(String[] args) {
        int[] arr = { 5, 4, 3, 2, 1 };
        System.out.print(cycle(arr, 4));
    }
}
```

**답 :**

**문제 2** IPv6의 특징에 대한 다음 설명에서 괄호에 들어갈 알맞은 답을 쓰시오. (5점)

1641202



IPv6는 현재 사용하고 있는 IP 주소 체계인 IPv4의 주소 부족 문제를 해결하기 위해 개발되었다. IPv6는 ( ) 비트씩 ( )부분, 총 128비트로 구성되어 있다. 각 부분을 16진수로 표현하고, 콜론(:)으로 구분한다.

**답 :**



**문제 3** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
#include <stdio.h>
#include <string.h>
main( ) {
    char a[11] = "KOREA";
    char b[] = "12345";

    strcat(a, b);
    puts(a);
    printf("%s", b);
    return 0;
}
```

**답 :**



**문제 4** OSI 7계층 중 다음에서 설명하는 특징을 갖는 계층이 무엇인지 쓰시오. (5점)

- 송·수신 측 간의 관련성을 유지하고 대화 제어를 담당한다.
- 대화(회화) 구성 및 동기 제어, 데이터 교환 관리 기능을 한다.
- 동기점은 오류가 있는 데이터의 회복을 위해 사용하는 것으로, 종류에는 소동기점과 대동기점이 있다.

**답 :**

**문제 5** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

1641205



```
public class Test {
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer( );
        sb.append("KOREA");
        sb.insert(3, "HRD");
        System.out.print(sb.toString( ));
    }
}
```

**답 :**

**문제 6** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

1641206



```
#include <stdio.h>
main( ) {
    printf("%d", -5 & 7);
    return 0;
}
```

**답 :**



**문제 7** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
public class Test {
    public static void main(String[] args) {
        int arr1[] = {3, 5, 9, 2, 5, 4};
        atest at = new atest();
        int arr2[] = at.rw(arr1);
        at.pa(arr2);
    }
}

class atest {
    int[] rw(int[] a) {
        int len = a.length;
        int b[] = new int[len];
        for(int i = 0; i < len; i++)
            b[i] = a[len - i - 1];
        return b;
    }
    void pa(int[] arr) {
        for(int i = 0; i < arr.length; i++) {
            System.out.print(arr[i]);
        }
    }
}
```

**답 :**





**문제 8** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
#include <stdio.h>
#define ArrSize 5

void into(int arr[], int gn, int ao);
void PrintArr(int arr[]);

void into(int arr[], int gn, int ao) {
    arr[ao] = gn;
}

void PrintArr(int arr[]) {
    int i;
    for (i = 0; i < ArrSize; i++) {
        printf("%d", arr[i]);
    }
    printf("\n");
}

int main( ) {
    int arr[ArrSize];
    int i, j;
    for (i = 1; i < 6; i++) {
        for (j = 0; j < ArrSize; j++) {
            into(arr, (j + 1) * i % 5, j);
        }
        PrintArr(arr);
    }
    return 0;
}
```

**답 :**

**문제 9** 다음 설명에 해당하는 무결성의 명칭을 쓰시오. (5점)

외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다. 즉 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정이다.



1641209

**답 :**

**문제 10** 넷스케이프 네비게이터 웹 브라우저를 위해 개발된 프로토콜로, SSL 프로토콜을 기반으로 하여 실행된다. 웹 브라우저와 서버 간의 안전한 통신을 위해 3-way Handshake 방식이 사용되는 이 프로토콜의 명칭을 쓰시오. (5점)



1641210

**답 :**

**문제 11** UNIX 시스템의 구성 요소 중 다음 설명에 해당하는 용어의 명칭을 한 단어로 쓰시오. (5점)

UNIX의 가장 핵심적인 부분으로 컴퓨터가 부팅될 때 주기억장치에 적재된 후 상주하면서 실행된다. 하드웨어를 보호하고, 프로그램과 하드웨어 간의 인터페이스 역할을 담당하며, 프로세스(CPU 스케줄링) 관리, 기억장치 관리, 파일 관리, 입·출력 관리, 데이터 전송 및 변환 등 여러 가지 기능을 수행한다.



1641211

**답 :**

**문제 12** Windows 10을 사용하는 컴퓨터에서 모니터 두 대를 연결하여 사용 시 다음과 같은 메뉴가 표시되도록 하는 단축키를 쓰시오. (5점)



1641212

- |          |            |
|----------|------------|
| • PC 화면만 | • 복제       |
| • 확장     | • 두 번째 화면만 |

**답 :**

**문제 13** 다음은 <회원> 테이블에 '주소' 필드를 추가하는 SQL문이다. 괄호에 들어갈 알맞은 명령어를 한 단어로 쓰시오. (5점)

1641213



<SQL문>

ALTER TABLE 회원 (                    ) 주소 CHAR(30);

**답 :**

**문제 14** Windows 10을 사용하는 컴퓨터에서 윈도우 탐색기가 실행되도록 하는 단축키를 쓰시오. (5점)

1641214



**답 :**

**문제 15** DML 중 데이터베이스의 데이터를 변경할 때 사용하는 명령어의 명칭을 한 단어로 쓰시오. (5점)

1641215



**답 :**

**문제 16** 다음은 <cus> 테이블의 'name' 필드와 <emp> 테이블의 'name' 필드를 하나의 테이블로 통합하여 검색하되, 같은 레코드도 중복하여 출력하는 SQL문이다. 괄호에 들어갈 알맞은 명령어를 한 단어로 쓰시오. (5점)

1641216



Select name From cus UNION (                    ) Select name From emp;

**답 :**

1641217



**문제 17** 다음은 <회원> 테이블에서 '전화번호' 필드가 비어있는 회원의 '이름'을 검색하는 SQL문이다. 괄호에 들어갈 알맞은 명령어를 쓰시오. (5점)

<회원>

회원번호	이름	전화번호
S1	김종일	11-1234
S2	김선길	22-2345
S3	김용갑	33-3456
S4	김원중	
S5	김예중	55-5678

<SQL>

Select 이름 From 회원 Where 전화번호 (            );

**답 :**

1641218



**문제 18** 후보키 중에서 선정된 기본키를 제외한 나머지 후보키를 의미하는 명칭을 한 단어로 쓰시오. (5점)

**답 :**

1641219



**문제 19** 다음에 제시된 보기는 소프트웨어 개발 과정에서 진행되는 소프트웨어 테스트들을 나열한 것이다. 보기에 제시된 테스트 진행 과정을 순서대로 나열하시오. (5점)

<보기>

• 통합 검사          • 인수 검사          • 시스템 검사          • 단위 검사

**답 :** (            ) → (            ) → (            ) → (            )

1641220



**문제 20** 정규화 과정 중 릴레이션 R의 모든 결정자(Determinant)가 후보키이면 그 릴레이션 R은 어떤 정규형에 속하는지 쓰시오. (5점)

**답 :**

## [문제 1]

4

## 해설

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

```
public static void main(String[] args) {
  ❶ int[] arr = { 5, 4, 3, 2, 1 };
  ❷ System.out.print(cycle(arr, 4));
}
```

❶ 정수형 배열 arr을 선언하고 초기화한다. 배열의 크기를 생각하면, 초기값의 수로 배열의 크기가 지정되므로 5개의 요소를 갖는 배열이 선언된다.

	[0]	[1]	[2]	[3]	[4]
arr	5	4	3	2	1

❷ 배열 arr의 시작 주소와 4를 인수로 하여 cycle() 메소드를 호출한 다음 돌려받은 값을 출력한다.

```
  ❸ static int cycle(int[] arr, int i) {
  ❹ if (i <= 0) return 0;
  ❺ return arr[i] % 3 + cycle(arr, i - 1);
}
```

❸ 정수를 반환하는 cycle() 메소드의 시작점이다. ❷번에서 전달받은 배열의 주소는 배열 arr이 받고 4는 i가 받는다.

※ 배열의 이름은 배열의 시작 주소를 가리키므로, 인수로 전달하는 경우 메소드에서 변경된 값이 main()의 배열에도 적용된다는 점을 염두에 두세요.

❹ i가 0보다 작거나 같으면 메소드를 호출했던 ❷번으로 이동하여 0을 반환한다. i의 값 4는 0보다 작거나 같지 않으므로 ❻번을 수행한다.

❺ arr[i] % 3 + cycle(arr, i - 1)을 계산한 후 메소드를 호출했던 ❷번으로 결과를 반환한다. ❺번을 수행하기 위해 cycle() 메소드를 호출하는데, 호출할 때 전달되는 값은 cycle(arr, i-1)이므로 cycle(arr, 3)인 상태로 호출된다.

```
  ❻ static int cycle(int[] arr, int i) {
  ❼ static int cycle(int[] arr, int i) {
  ❽ return arr[i] % 3 + cycle(arr, i - 1);
}
```

cycle() 메소드가 호출될 때 arr의 주소와 3을 전달받았으므로 i는 3이다. ❼번의 조건을 만족하지 않으므로 ❽번을 수행한다. ❽번을 수행하기 위해 cycle() 메소드를 호출하는데, 호출할 때 전달되는 값은 cycle(arr, i-1)이므로 cycle(arr, 2)인 상태로 호출된다.

```
  ❾ static int cycle(int[] arr, int i) {
  ❿ if (i <= 0) return 0;
  ⓫ return arr[i] % 3 + cycle(arr, i - 1);
}
```

cycle() 메소드가 호출될 때 arr의 주소와 2를 전달받았으므로 i는 2이다. ❿번의 조건을 만족하지 않으므로 ⓫번을 수행한다. ⓫번을 수행하기 위해 cycle() 메소드를 호출하는데, 호출할 때 전달되는 값은 cycle(arr, i-1)이므로 cycle(arr, 1)인 상태로 호출된다.

```
  ⓬ static int cycle(int[] arr, int i) {
  ⓭ if (i <= 0) return 0;
  ⓮ return arr[i] % 3 + cycle(arr, i - 1);
}
```

cycle() 메소드가 호출될 때 arr의 주소와 1을 전달받았으므로 i는 1이다. ⓭번의 조건을 만족하지 않으므로 ⓮번을 수행한다. ⓮번을 수행하기 위해 cycle() 메소드를 호출하는데, 호출할 때 전달되는 값은 cycle(arr, i-1)이므로 cycle(arr, 0)인 상태로 호출된다.

```

15 static int cycle(int[] arr, int i) {
16     if (i <= 0) return 0;
        return arr[i] % 3 + cycle(arr, i - 1);
    }

```

cycle() 메소드가 호출될 때 arr의 주소와 0을 전달받았으므로 i는 0이다. 16번의 조건을 만족하므로 함수의 실행을 종료하고 0을 반환하면서 제어를 cycle(arr, 0) 메소드를 호출했던 14번으로 옮긴다.

```

12 static int cycle(int[] arr, int i) {
13     if (i <= 0) return 0;
14 17 return arr[i] % 3 + cycle(arr, i - 1);
    }

```

17 16번에서 0을 반환하였으므로 1을 반환하면서 제어를 cycle(arr, 1) 메소드를 호출했던 11번으로 옮긴다.  
return arr[i] % 3 + cycle(arr, i - 1);

㉑                      ㉒

- ㉑ : arr[1] % 3 → 4 % 3 = 1 ('cycle(arr, i-1)'을 호출할 때 i는 1이었으므로)
- ㉒ : 0 (16번에서 0을 반환하였으므로)

∴ ㉑+㉒는 1이므로 함수를 호출했던 11번으로 1이 반환된다.

```

9 static int cycle(int[] arr, int i) {
10     if (i <= 0) return 0;
11 18 return arr[i] % 3 + cycle(arr, i - 1);
    }

```

18 17번에서 1을 반환하였으므로 1을 반환하면서 제어를 cycle(arr, 2) 메소드를 호출했던 8번으로 옮긴다.  
return arr[i] % 3 + cycle(arr, i - 1);

㉑                      ㉒

- ㉑ : arr[2] % 3 → 3 % 3 = 0 ('cycle(arr, i-1)'을 호출할 때 i는 2였으므로)
- ㉒ : 1 (17번에서 1을 반환하였으므로)

∴ ㉑+㉒는 1이므로 함수를 호출했던 8번으로 1이 반환된다.

```

6 static int cycle(int[] arr, int i) {
7     if (i <= 0) return 0;
8 19 return arr[i] % 3 + cycle(arr, i - 1);
    }

```

19 18번에서 1을 반환하였으므로 3을 반환하면서 제어를 cycle(arr, 3) 메소드를 호출했던 5번으로 옮긴다.  
return arr[i] % 3 + cycle(arr, i - 1);

㉑                      ㉒

- ㉑ : arr[3] % 3 → 2 % 3 = 2 ('cycle(arr, i-1)'을 호출할 때 i는 3이었으므로)
- ㉒ : 1 (18번에서 1을 반환하였으므로)

∴ ㉑+㉒는 3이므로 함수를 호출했던 5번으로 3이 반환된다.

```

3 static int cycle(int[] arr, int i) {
4     if (i <= 0) return 0;
5 20 return arr[i] % 3 + cycle(arr, i - 1);
    }

```

20 19번에서 3을 반환하였으므로 4를 반환하면서 제어를 cycle(arr, 4) 메소드를 호출했던 2번으로 옮긴다.  
return arr[i] % 3 + cycle(arr, i - 1);

㉑                      ㉒

- ㉑ : arr[4] % 3 → 1 % 3 = 1 ('cycle(arr, i-1)'을 호출할 때 i는 4였으므로)

- ⑥: 3 (②번에서 3을 반환하였으므로)  
 ∴ ③+⑥는 40이므로 함수를 호출했던 ②번으로 4가 반환된다.

```
public static void main(String[] args) {
    ① int[] arr = { 5, 4, 3, 2, 1 };
    ② ② System.out.print(cycle(arr, 4));
}
```

- ② ②번에서 4를 반환하였으므로 돌려받은 값 4를 출력하고 프로그램을 종료한다.

결과 4

## [문제 2]

16, 8

## [문제 3]

KOREA12345

12345

- ※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이를 쉼표로 구분하여 KOREA12345, 12345나 공백으로만 구분하여 KOREA12345 12345와 같이 한 줄로 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

```
#include <stdio.h>
#include <string.h>
main( ) {
    ① char a[11] = "KOREA";
    ② char b[] = "12345";

    ③ strcat(a, b);
    ④ puts(a);
    ⑤ printf("%s", b);
    ⑥ return 0;
}
```

- ① 11개의 요소를 갖는 문자형 배열 a를 선언하고 초기화한다. 문자열의 경우 문자열의 끝을 의미하는 널 문자('\0')가 추가로 저장되며, 출력 시 널 문자는 표시되지 않는다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
a	K	O	R	E	A	\0					

- ② 문자형 배열 b를 선언하고 초기화한다. 저장되는 값은 숫자의 형태이지만 큰따옴표로 묶여있으므로 문자열로 저장된다. 또한 배열의 크기를 생략하면, 초기값으로 지정된 개수만큼 배열이 선언되므로 6개의 요소를 갖는 배열로 선언된다.

	[0]	[1]	[2]	[3]	[4]	[5]
b	1	2	3	4	5	\0

- ③ 배열 a의 뒤쪽에 배열 b를 이어 붙여 하나의 문자열로 저장한다. 배열 a의 널 문자('\0')가 제거되고 그 위치부터 배열 b의 값이 이어 붙는다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
a	K	O	R	E	A	1	2	3	4	5	\0

- **strcat(문자열A, 문자열B)** : 문자열A의 뒤에 문자열B를 연결하여 붙이는 함수

④ 배열 a에 저장된 문자열 전체를 출력한 후 커서를 다음 줄의 처음으로 옮긴다.

결과 KOREA12345

- `puts()`: 인수로 주어진 문자열을 화면에 출력한 후 커서를 자동으로 다음 줄 처음으로 이동시키는 함수

⑤ 배열 b에 저장된 값을 문자열로 출력한다.

결과 KOREA12345  
12345

⑥ `main()` 함수에서의 `return 0;`은 프로그램의 종료를 의미한다.

[문제 4]

※ 다음 중 하나를 쓰면 됩니다.

세션 계층, Session Layer

[문제 5]

KORHRDEA

※ **답안 작성 시 주의 사항** : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값을 소문자로 korhrdea와 같이 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
public class Test {  
    public static void main(String[] args) {  
        ①    StringBuffer sb = new StringBuffer( );  
        ②    sb.append("KOREA");  
        ③    sb.insert(3, "HRD");  
        ④    System.out.print(sb.toString( ));  
    }  
}
```

① `StringBuffer` 자료형의 객체 변수 `sb`를 선언한다.

- **StringBuffer 자료형**: 문자열을 추가하거나 변경할 수 있는 자료형
  - `append(문자열)`: 문자열을 추가하는 메소드
  - `insert(위치, 문자열)`: 지정한 위치에 문자열을 삽입하는 메소드
  - `substring(시작위치, 끝위치)`: 시작위치부터 끝위치까지의 문자열을 추출하는 메소드

② `sb`에 "KOREA" 문자열을 추가한다.

	[0]	[1]	[2]	[3]	[4]
sb	K	O	R	E	A

③ `sb`의 3 위치, 즉 `sb[3]`에 "HRD" 문자열을 삽입한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
sb	K	O	R	H	R	D	E	A

④ `sb`를 `String` 자료형으로 변환한 후 변환된 값을 출력한다.

- `toString`: `String` 자료형으로 변환하는 메소드

결과 KORHRDEA



## [문제 6]

3

## 해설

```
#include <stdio.h>
main( ) {
  ❶ printf("%d", -5 & 7);
  ❷ return 0;
}
```

- ❶ -5와 7을 &(비트 and) 연산한 결과를 정수형으로 출력한다.

C언어에서 정수형은 4바이트(32비트)이므로 각각의 값을 4바이트 이진수로 변환한 다음 비트별로 연산한다.

```
-5 = 1111 1111 1111 1111 1111 1111 1111 1011
 7 = 0000 0000 0000 0000 0000 0000 0000 0111
& = 0000 0000 0000 0000 0000 0000 0000 0011
∴ 0000 0000 0000 0000 0000 0000 0000 0011은 십진수로 3이다.
```

결과 **3**

- ❷ 프로그램을 종료한다.

## [문제 7]

452953

## 해설

```
public class Test {
    public static void main(String[] args) {
        ❶ int arr1[] = {3, 5, 9, 2, 5, 4};
        ❷ atest at = new atest();
        ❸❩ int arr2[] = at.rw(arr1);
        ❹ at.pa(arr2);
    } ❺
}

class atest {
    ❻ int[] rw(int[] a) {
        ❼ int len = a.length;
        ❶ int b[] = new int[len];
        ❷ for(int i = 0; i < len; i++)
        ❸ b[i] = a[len - i - 1];
        ❹ return b;
    }
    ❺ void pa(int[] arr) {
        ❻ for(int i = 0; i < arr.length; i++) {
        ❼ System.out.print(arr[i]);
        ❶ }
    }
}
```

모든 Java 프로그램은 반드시 `main()` 메소드에서 시작한다.

- ❶ 정수형 배열 `arr1`을 선언하고 초기화한다. 배열의 크기를 생각하면, 초기값의 수로 배열의 크기가 지정되므로 6개의 요소를 갖는 배열이 선언된다.

	[0]	[1]	[2]	[3]	[4]	[5]
arr1	3	5	9	2	5	4

- ❷ 클래스 `atest`의 객체 변수 `at`를 선언한다.

- ❸ 정수형 배열 `arr2`를 선언하고, 배열 `arr1`의 시작 주소를 인수로 하여 `at`의 `rw()` 메소드를 호출한 후 반환받은 결과로 `arr2`를 초기화한다. `at`는 `atest` 클래스의 객체 변수이므로 `atest`의 `rw()` 메소드인 ❹번이 호출된다.

- ❹ 정수형 배열을 반환하는 `rw()` 메소드의 시작점이다. ❸번에서 전달받은 주소는 배열 `a`가 받는다.

	[0]	[1]	[2]	[3]	[4]	[5]
a	3	5	9	2	5	4

※ 배열의 이름은 배열의 시작 주소를 가리키므로, 인수로 전달하는 경우 메소드에서 변경된 값이 `main()`의 배열에도 적용된다는 점을 염두에 두세요.

- ❺ 정수형 변수 `len`을 선언하고 배열 `a`의 길이인 6으로 초기화한다.

• `length`: 배열 요소의 개수를 저장하고 있는 속성

- ❻ `len`이 6이므로 6개의 요소를 갖는 정수형 배열 `b`를 선언한다.

	[0]	[1]	[2]	[3]	[4]	[5]
b						

- ❼ 반복 변수 `i`가 0부터 1씩 증가하면서 `len`의 값인 6보다 작은 동안 ❸번을 반복 수행한다.

- ❽ `a[len-i-1]`의 값을 `b[i]`에 저장한다. `a` 배열에 저장된 값들이 `b` 배열에 뒤에서부터 차례대로 저장된다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

b[i]	len	i	배열 a	a[len-i-1]
[0] [1] [2] [3] [4] [5]			[0] [1] [2] [3] [4] [5]	
4	6	0	3 5 9 2 5 4	a[5] = 4
4 5		1	3 5 9 2 5 4	a[4] = 5
4 5 2		2	3 5 9 2 5 4	a[3] = 2
4 5 2 9		3	3 5 9 2 5 4	a[2] = 9
4 5 2 9 5		4	3 5 9 2 5 4	a[1] = 5
4 5 2 9 5 3		5	3 5 9 2 5 4	a[0] = 3
		6		

- ❹ 배열 `b`의 시작 주소를 `rw()` 메소드를 호출했던 ❶번으로 반환한다.

- ❶ ❸번에서 반환받은 배열 `b`의 시작 주소로 배열 `arr2`를 초기화한다.

	[0]	[1]	[2]	[3]	[4]	[5]
arr2	4	5	2	9	5	3

- ❷ 배열 `arr2`의 시작 주소를 인수로 하여 `at`의 `pa()` 메소드를 호출한다. `at`는 `atest` 클래스의 객체 변수이므로 `atest`의 `pa()` 메소드인 ❷번이 호출된다.

- ❸ `pa()` 메소드의 시작점이다. ❷번에서 전달받은 배열의 주소는 배열 `arr`이 받는다.

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

- ❹ 반복 변수 `i`가 0부터 1씩 증가하면서 6보다 작은 동안 ❸번을 반복 수행한다.

- ❺ `arr[i]`의 값을 출력한다. 반복문이 종료되어 메소드가 끝나면 메소드를 호출했던 ❶번의 다음 줄인 ❺번으로 이동하여 프로그램을 종료한다.

결과 452953

## [문제 8]

12340

24130

31420

43210

00000

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이를 쉼표로 구분하여 12340, 24130, 31420, 43210, 00000이나 공백으로만 구분하여 12340 24130 31420 43210 00000과 같이 한 줄로 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

```
#include <stdio.h>
#define ArrSize 5          숫자 5를 ArrSize로 정의한다. 프로그램 안에서 ArrSize는 숫자 5와 동일하게 쓰인다.

void into(int arr[], int gn, int ao);    사용할 into() 함수를 선언한다.
void PrintArr(int arr[]);              사용할 PrintArr() 함수를 선언한다.

⑥ void into(int arr[], int gn, int ao) {
⑦     arr[ao] = gn;
    }

⑨ void PrintArr(int arr[]) {
⑩     int i;
⑪     for (i = 0; i < ArrSize; i++) {
⑫         printf("%d", arr[i]);
    }
⑬     printf("\n");
    }

int main( ) {
①     int arr[ArrSize];
②     int i, j;
③     for (i = 1; i < 6; i++) {
④         for (j = 0; j < ArrSize; j++) {
⑤             into(arr, (j + 1) * i % 5, j);
        }
⑧         PrintArr(arr);
    }
⑭     return 0;
}
```

모든 C 프로그램은 반드시 main() 함수에서 시작한다.

① ArrSize의 값인 5, 즉 5개의 요소를 갖는 정수형 배열 arr을 선언한다.

	[0]	[1]	[2]	[3]	[4]
arr					

② 정수형 변수 i와 j를 선언한다.

③ 반복 변수 i가 1부터 1씩 증가하면서 6보다 작은 동안 ④~⑧번을 반복 수행한다. ⑧번은 함수 호출문이므로 함수에서 수행되는 코드를 포함하면 반복문이 수행하는 반복문의 범위는 ④~⑧번이 된다.

④ 반복 변수 j가 0부터 1씩 증가하면서 ArrSize의 값인 5보다 작은 동안 ⑤번을 반복 수행한다.

- ⑤ 배열 arr의 시작 주소와  $(j + 1) * i \% 5$ , j의 값을 인수로 하여 into() 함수를 호출한다.
- ⑥ into 함수의 시작점이다. ⑤번에서 전달받은 값들을 배열 arr, gn, ao가 순서대로 받는다. into 함수의 수행 결과는  $arr[j] = (j + i) * i \% 5$ 와 같다.  
※ 배열의 이름은 배열의 시작 주소를 가리키므로, 인수로 전달하는 경우 함수에서 변경된 값이 main() 함수의 배열에도 적용된다는 점을 염두에 두세요.
- ⑦ arr[ao]에 gn의 값을 저장한다. 메소드가 종료되었으므로 메소드를 호출했던 ⑤번으로 이동하여 반복문을 계속 수행한다.
- ⑧ 배열 arr의 시작 주소를 인수로 하여 PrintArr() 함수를 호출한다.
- ⑨ PrintArr 함수의 시작점이다. ⑧번에서 전달받은 주소를 배열 arr0이 받는다.
- ⑩ 정수형 변수 i를 선언한다.
- ⑪ 반복 변수 i가 0부터 1씩 증가하면서 ArrSize의 값인 5보다 작은 동안 ⑩번을 반복 수행한다.
- ⑫ arr[i]의 값을 정수형으로 출력한다.
- ⑬ 한 줄을 띄운다.
- ⑭ 프로그램을 종료한다.
- ※ 반복문의 실행에 따른 변수들의 변화는 다음과 같다.

main ( )			into ( )					PrintArr ( )		
i	j	$(j+1)*i\%5$	gn	ao	arr [0] [1] [2] [3] [4]					i    반복문 출력    전체 출력
1	0	(1)*1 % 5 = 1	1	0	1 2 3 4 0					0    1    12340
	1	(2)*1 % 5 = 2	2	1						1    12
	2	(3)*1 % 5 = 3	3	2						2    123
	3	(4)*1 % 5 = 4	4	3						3    1234
	4	(5)*1 % 5 = 0	0	4						4    12340
	5									5
2	0	(1)*2 % 5 = 2	2	0	2 4 1 3 0					0    2    12340
	1	(2)*2 % 5 = 4	4	1						1    24    24130
	2	(3)*2 % 5 = 1	1	2						2    241
	3	(4)*2 % 5 = 3	3	3						3    2413
	4	(5)*2 % 5 = 0	0	4						4    24130
	5									5
3	0	(1)*3 % 5 = 3	3	0	3 1 4 2 0					0    3    12340
	1	(2)*3 % 5 = 1	1	1						1    31    24130
	2	(3)*3 % 5 = 4	4	2						2    314    31420
	3	(4)*3 % 5 = 2	2	3						3    3142
	4	(5)*3 % 5 = 0	0	4						4    31420
	5									5
4	0	(1)*4 % 5 = 4	4	4	4 3 2 1 0					0    4    12340
	1	(2)*4 % 5 = 3	3	1						1    43    24130
	2	(3)*4 % 5 = 2	2	2						2    432    31420
	3	(4)*4 % 5 = 1	1	3						3    4321    43210
	4	(5)*4 % 5 = 0	0	4						4    43210
	5									5
5	0	(1)*5 % 5 = 0	0	0	0 0 0 0 0					0    0    12340
	1	(2)*5 % 5 = 0	0	1						1    00    24130
	2	(3)*5 % 5 = 0	0	2						2    000    31420
	3	(4)*5 % 5 = 0	0	3						3    0000    43210
	4	(5)*5 % 5 = 0	0	4						4    00000    00000
	5									5
6										

## [문제 9]

참조 무결성

## [문제 10]

※ 다음 중 하나를 쓰면 됩니다.

HTTPS, HyperText Transfer Protocol Secure

## [문제 11]

※ 다음 중 하나를 쓰면 됩니다.

커널, Kernel

## [문제 12]

Windows key + P

## [문제 13]

ADD

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

ALTER TABLE 회원  
ADD 주소 CHAR(30);

수정할 테이블의 이름은 <회원>이다.  
문자 30자리인 '주소' 속성을 추가한다.

## [문제 14]

Windows Key + E

## [문제 15]

UPDATE

## [문제 16]

ALL

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

Select name From cus  
UNION ALL  
Select name From emp;

<cus> 테이블의 'name' 속성을 표시한다.  
두 Select문의 조회 결과를 통합하되 중복된 행도 그대로 출력한다.  
<emp> 테이블의 'name' 속성을 표시한다.

## [문제 17]

is NULL

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

해설

Select 이름	'이름' 속성을 표시한다.
From 회원	〈회원〉 테이블을 검색한다.
Where 전화번호 is Null;	'전화번호'가 비어 있는 자료만을 대상으로 검색한다.

〈결과〉

이름
김원중

[문제 18]

※ 다음 중 하나를 쓰면 됩니다.

대체키, Alternate Key

[문제 19]

단위 검사 → 통합 검사 → 시스템 검사 → 인수 검사

[문제 20]

※ 다음 중 하나를 쓰면 됩니다.

BCNF, Boyce-Codd Normal Form



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음

1641101



**문제 1** 계정 ID가 “김공단”인 사용자에게 <emp> 테이블에 대한 삭제 권한을 부여하고, 부여한 삭제 권한을 다른 계정에 부여할 수 있는 권한까지 부여하고자 한다. (5점)

GRANT DELETE ON emp TO 김공단 WITH (        ) OPTION;

**답 :**

1641102



**문제 2** 트랜잭션의 특징 중 다음 괄호에 들어갈 알맞은 특징을 쓰시오. (5점)

트랜잭션의 특징 중 (        )은/는 성공적으로 완료(COMMIT)된 트랜잭션의 결과가 영구적으로 반영되어야 한다는 성질을 말한다.

**답 :**

1641103



**문제 3** 다음에서 설명하는 것이 무엇인지 쓰시오. (단 상용 소프트웨어의 명칭은 제외) (5점)

사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해 주고, 데이터베이스를 관리해 주는 소프트웨어이다.

**답 :**

1641104



**문제 4** 다음 중 화이트박스 테스트의 종류가 아닌 것을 모두 골라 기호(㉠~㉣)로 쓰시오. (5점)

- |            |          |             |
|------------|----------|-------------|
| ㉠ 기초 경로 검사 | ㉡ 한계값 분석 | ㉢ 데이터 흐름 검사 |
| ㉣ 조건 검사    | ㉤ 비교 검사  |             |

**답 :**



**문제 5** 다음은 Windows 10 Home과 Windows 10 Pro의 차이점을 설명한 것이다. 괄호에 공통으로 들어갈 적합한 용어를 쓰시오. (5점)

1641105



Windows 10의 Home과 Pro의 가장 큰 차이점은 보안 기능이다. 장치 암호화, 방화벽 및 네트워크 보호, 보안 루팅 등의 기능은 두 버전 모두 공통적으로 제공되고 있으나, (     ), WIP(Windows Information Protection)는 Pro에서만 제공되고 있다. (     )은(는) 128비트 키의 CBC 모드에서 AES 암호화 알고리즘을 사용하여 데이터를 보호하는 기능으로, 장치를 분실하거나 도난당하더라도 다른 사람이 장치의 데이터에 접근할 수 없도록 한다.

**답 :**

**문제 6** 다음에서 설명하는 프로토콜이 무엇인지 <보기>에서 찾아 기호(㉠~㉤)로 쓰시오. (5점)

1641106



OSI 7계층의 전송 계층에 해당하는 프로토콜로, 패킷의 다중화, 순서 제어, 오류 제어, 흐름 제어 기능을 제공한다. 신뢰성 있는 연결형 서비스를 제공하며, 신뢰성 있는 연결을 위해 송신지와 수신지 간의 통신에 앞서 단계에 걸친 확인 작업인 3 Way Handshake를 수행한다.

<보기>

- |       |          |        |
|-------|----------|--------|
| ㉠ TCP | ㉡ IP     | ㉢ HTTP |
| ㉣ FTP | ㉤ TELNET |        |

**답 :**



**문제 7** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
#include <stdio.h>
main( ) {
    int a[5] = {5, 4, 3, 2, 1};
    int i = 4, sum = 0;
    do {
        a[i] = a[i] % 3;
        sum = sum + a[i];
        i--;
    } while (i > 0);
    printf("%d", sum);
}
```

**답 :**

**문제 8** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

1641108



```
public class hrdkorea {
    public static void main(String[ ] args) {
        int[ ] arr1 = {3, 5, 9, 2, 1, 5};
        atest at = new atest();
        at.pa(arr1, 0);
    }
}

class atest {
    void pa(int[ ] arr, int c) {
        int i;
        if (c > 1) {
            for (i = 0; i < arr.length; i++) {
                if ( (i + 1) % c == 0 ) {
                    System.out.printf("%d", arr[i]);
                }
            }
        }
        else {
            for(i = 0; i < arr.length; i++) {
                System.out.printf("%d", arr[i]);
            }
        }
    }
}
```

**답 :**

**문제 9** 윈도우 운영체제의 도스 프롬프트 모드에서 사용하는 것으로, UNIX의 ls와 동일하게 현재 디렉터리의 파일 목록을 표시하는 명령어를 쓰시오. (5점)

1641109



**답 :**

**문제 10** 다음 설명의 괄호(①, ②)에 들어갈 알맞은 키를 쓰시오. (5점)

1641110



Windows에서는 바탕 화면을 여러 개 만들어 바탕 화면별로 각각의 앱을 실행해 놓고 바탕 화면을 전환해 가면서 작업할 수 있는 가상 데스크톱 기능을 지원한다. 키보드의 [Windows Key]+[Tab]을 누르면 현재 설정된 가상 데스크톱을 확인할 수 있다. 새로운 가상 데스크톱을 생성하려면 키보드의 [Ctrl]+[Windows Key]+( ① )를 눌러야 하고, 현재 작업 중인 가상 데스크톱을 제거하려면 키보드의 [Ctrl]+[Windows Key]+( ② )를 눌러야 한다.

**답 :**

**문제 11** 다음은 <학생> 테이블에 가변적인 5자리 문자가 저장되는 속성을 추가하는 명령문이다. 괄호에 들어갈 알맞은 명령을 쓰시오. (5점)

1641111



ALTER TABLE 학생 ADD 주소 (                    );

**답 :**

**문제 12** 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블을 무엇이라고 하는지 쓰시오. (5점)

1641112



**답 :**

**문제 13** 다음은 <emp> 테이블을 사용하여 고유한 특성을 갖는 'empidx'라는 이름의 인덱스를 정의하되, 'empid' 속성에 대해 오름차순으로 정렬하는 SQL문이다. 괄호에 들어갈 알맞은 명령어를 쓰시오. (5점)

1641113



<SQL문>

CREATE (                    ) INDEX empidx ON emp(empid ASC);

**답 :**

**문제 14** 다음 JAVA로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

1641114



```
public class hrdkorea {
    public static void main(String[ ] args) {
        int cnt = 0;
        int sum = 0;
        for (int i = 0; i <= 7; i++)
        {
            if (i % 2 == 1)
            {
                cnt = cnt + 1;
                sum = sum + i;
            }
        }
        System.out.printf(cnt + ", " + sum);
    }
}
```

**답 :**

**문제 15** 다음에서 설명하는 유틸리티가 무엇인지 쓰시오. (5점)

1641115



ICMP(Inter Control Message Protocol)를 사용하여 특정 호스트로의 연결 가능 여부를 확인하는 유틸리티이다. ICMP 에코 요청 패킷을 대상 호스트로 전송한 후 ICMP 에코 응답을 받으면 해당 호스트는 연결 가능한 호스트임이 확인된다.

**답 :**



**문제 16** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

```
#include <stdio.h>
main( )
{
    int a[5] = {3, 2, 5, 1, 4};
    int temp, i;
    for (i = 0; i < 4; i++)
    {
        temp = a[i];
        a[i] = a[i + 1];
        a[i + 1] = temp;
    }
    for (i = 0; i < 5; i++)
    {
        printf("%d", a[i]);
    }
}
```

**답 :**



**문제 17** OSI 7계층 중 네트워크 계층에서 사용되는 프로토콜을 모두 골라 기호(㉠~㉤)로 쓰시오. (5점)

- |        |        |       |
|--------|--------|-------|
| ㉠ ICMP | ㉡ IP   | ㉢ TCP |
| ㉣ UDP  | ㉤ HTTP |       |

**답 :**

**문제 18** 다음 JAVA로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

1641118



```
public class hrdkorea {
    public static void main(String[] args) {
        String str = "HelloWorld!";
        String a1 = str.substring(5);
        System.out.println(a1.toUpperCase());
    }
}
```

답 :

**문제 19** SET Operation(집합 연산자) 중 여러 개의 SQL문의 결과에 대한 합집합으로, 결과에서 모든 중복된 행은 하나의 행으로 만드는 연산자를 쓰시오. (5점)

1641119



답 :

**문제 20** 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단 공백과 개행에 주의하여 답안을 작성하시오.) (5점)

1641120



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

답 :

## [문제 1]

GRANT

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
GRANT DELETE ON emp
TO 김공단
WITH GRANT OPTION;
```

〈emp〉 테이블에 대한 삭제(DELETE) 권한을 부여한다.  
계정 ID가 "김공단"인 사용자에게  
부여받은 권한을 다른 사용자에게 다시 부여할 수 있는 권한을 부여한다.

## [문제 2]

※ 다음 중 하나를 쓰면 됩니다.

영속성, Durability

## [문제 3]

※ 다음 중 하나를 쓰면 됩니다.

데이터베이스 관리 시스템, DBMS, DataBase Management System

## [문제 4]

㉠, ㉡

## [문제 5]

※ 다음 중 하나를 쓰면 됩니다.

비트로커, BitLocker

## [문제 6]

㉡

## [문제 7]

4

## 해설

```
#include <stdio.h>
main( ) {
  ① int a[5] = {5, 4, 3, 2, 1};
  ② int i = 4, sum = 0;
  ③ do {
  ④     a[i] = a[i] % 3;
  ⑤     sum = sum + a[i];
  ⑥     i--;
  ⑦ } while (i > 0);
  ⑧ printf("%d", sum);
}
```

① 5개의 요소를 갖는 정수형 배열 a를 선언하고 초기화한다.



	[0]	[1]	[2]	[3]	[4]
a	5	4	3	2	1

- ② 정수형 변수 i와 sum을 선언하고 i를 4로, sum을 0으로 초기화한다.  
 ③ do~while 반복문의 시작점이다. ④~⑥번 문장을 반복하여 수행한다.  
 ④ a[i]를 3으로 나눈 나머지를 a[i]에 저장한다.  
 ⑤ sum에 a[i]의 값을 누적시킨다.  
 ⑥ i = i - 1;과 동일하다. i의 값을 1씩 감소시킨다.  
 ⑦ i가 0보다 큰 동안 ④~⑥번 문장을 반복하여 수행한다.  
 반복문 실행에 따른 변수들의 변화는 다음과 같다.

a	i	a[i]%3	sum
[0] [1] [2] [3] [4]	4	1	0
5 4 3 2 1	3	2	1
1 0 2 1	2	0	3
	1	1	3
	0		4

- ⑧ sum의 값을 정수로 출력한다.

결과 4

### [문제 8]

359215

- ※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이를 쉼표(.)나 공백으로 구분하여 3,5,9,2,1,5와 같이 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

#### 해설

```
public class hrdkorea {
    public static void main(String[] args) {
        ① int[] arr1 = {3, 5, 9, 2, 1, 5};
        ② atest at = new atest();
        ③ at.pa(arr1, 0);
    } ⑩
}

class atest {
    ④ void pa(int[] arr, int c) {
        ⑤ int i;
        ⑥ if (c > 1) {
            for (i = 0; i < arr.length; i++) {
                if ( (i + 1) % c == 0 ) {
                    System.out.printf("%d", arr[i]);
                }
            }
        }
        else {
            ⑦ for (i = 0; i < arr.length; i++) {
                ⑧ System.out.printf("%d", arr[i]);
            }
        }
    }
}
```

```
    } ⑨  
}  
}
```

모든 Java 프로그램은 반드시 `main()` 메소드에서 시작한다.

① 6개의 요소를 갖는 정수형 배열 `arr1`을 선언하고 초기화한다. 개수를 지정하지 않으면, 초기값으로 지정된 수만큼 배열의 요소가 만들어진다.

	[0]	[1]	[2]	[3]	[4]	[5]
arr1	3	5	9	2	1	5

② 클래스 `atest`의 객체 변수 `at`를 선언한다.

③ 배열 `arr1`의 시작 주소와 0을 인수로 하여 `atest`의 `pa()` 메소드를 호출한다.

④ 반환값이 없는 `pa` 메소드의 시작점이다. ⑥번에서 전달받은 배열 `arr1`의 주소는 배열 `arr`이 받고 0은 `c`가 받는다.

※ 배열의 이름은 배열의 시작 주소를 가리키므로, 인수로 전달하는 경우 메소드에서 변경된 값이 `main()`의 배열에도 적용된다는 점을 염두에 두세요.

	[0]	[1]	[2]	[3]	[4]	[5]		c
arr	3	5	9	2	1	5		0

⑤ 정수형 변수 `i`를 선언한다.

⑥ `c`의 값이 1보다 크면 바로 다음 문장으로 이동하고, 그렇지 않으면 `else`의 다음 문장인 ⑦번으로 이동한다. `c`의 값 0은 1보다 작으므로 ⑦번으로 이동한다.

⑦ 반복 변수 `i`가 0부터 1씩 증가하면서 배열 `arr`의 길이인 6보다 작은 동안 ⑧번을 반복 수행한다.

⑧ `arr[i]`의 값을 정수로 출력한다. 반복문이 종료되면 제어기가 ⑨번으로 이동하고, 메소드가 종료되었으므로 메소드를 호출했던 ③번의 다음 줄인 ⑩번으로 이동하여 프로그램이 종료된다.

※ 반복문 실행에 따른 변수들의 변화는 다음과 같다.

c	i	arr[i]	출력
0	0	3	3
	1	5	35
	2	9	359
	3	2	3592
	4	1	35921
	5	5	359215
	6		

[문제 9]

DIR

[문제 10]

① D    ② F4

[문제 11]

VARCHAR(5)

※ **답안 작성 시 주의 사항** : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

해설

```
ALTER TABLE 학생  
ADD 주소 VARCHAR(5);
```

수정할 테이블 이름은 <학생>이다.  
가변길이 문자 5자리인 '주소' 속성을 추가한다.

## [문제 12]

※ 다음 중 하나를 쓰면 됩니다.

뷰, View

## [문제 13]

UNIQUE

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

## 해설

```
CREATE UNIQUE INDEX empidx
ON emp(empid ASC);
```

중복 값이 없는 고유한 특성을 갖는 'empidx'라는 이름의 인덱스를 생성한다.  
<emp> 테이블을 사용하되, 'empid' 속성을 기준으로 오름차순 정렬한다.

## [문제 14]

4, 16

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이를 쉼표(,) 없이 공백으로만 구분하여 4 16과 같이 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

```
public class hrdkorea {
    public static void main(String[] args) {
        ① int cnt = 0;
        ② int sum = 0;
        ③ for (int i = 0; i <= 7; i++)
        {
            ④ if (i % 2 == 1)
            {
                ⑤ cnt = cnt + 1;
                ⑥ sum = sum + i;
            }
        }
        ⑦ System.out.printf(cnt + ", " + sum);
    }
}
```

- ① 정수형 변수 cnt를 선언하고 0으로 초기화한다.
- ② 정수형 변수 sum을 선언하고 0으로 초기화한다.
- ③ 반복 변수 i가 0부터 1씩 증가하면서 7보다 작거나 같은 동안 ④~⑥번을 반복 수행한다.
- ④ i를 2로 나눈 나머지가 1이면 ⑤~⑥번을 수행하고, 그렇지 않으면 ③번으로 이동한다.
- ⑤ cnt의 값을 1씩 증가시킨다. 즉, cnt에 홀수의 개수를 증가시킨다.
- ⑥ sum에 i의 값을 누적시킨다. 즉, sum에 홀수를 누적시킨다.
- ⑦ cnt의 값을 출력하고 쉼표(,)와 빈 칸을 출력한 후 sum의 값을 출력한다.

※ 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	i%2	cnt	sum	출력
0	0	0	0	4, 16
1	1	1	1	
2	0			
3	1	2	4	
4	0			
5	1	3	9	
6	0			
7	1	4	16	
8				

[문제 15]

ping

[문제 16]

25143

※ **답안 작성 시 주의 사항** : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이를 쉼표(.)로 구분하여 2,5,1,4,3과 같이 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
#include <stdio.h>
main( )
{
  ❶ int a[5] = {3, 2, 5, 1, 4};
  ❷ int temp, i;
  ❸ for (i = 0; i < 4; i++)
  ❹ {
    ❺ temp = a[i];
    ❻ a[i] = a[i + 1];
    ❼ a[i + 1] = temp;
  }
  ❽ for (i = 0; i < 5; i++)
  {
    ❾ printf("%d", a[i]);
  }
}
```

❶ 5개의 요소를 갖는 정수형 배열 a를 선언하고 초기화한다.

	[0]	[1]	[2]	[3]	[4]
a	3	2	5	1	4

❷ 정수형 변수 temp와 i를 선언한다.

❸ 반복 변수 i가 0부터 1씩 증가하면서 4보다 작은 동안 ❺~❷번을 반복 수행한다.

❺~❷ temp 변수를 이용하여 a[i]와 a[i+1]의 값을 교환한다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	temp	a[i]	a[i+1]	a				
				[0]	[1]	[2]	[3]	[4]
0	3	2	3	3	2	5	4	4
1	3	5	3	2	3	3	3	3
2	3	1	3		5	1	4	
3	3	4	3					
4								

- ㉔ 반복 변수 i가 0부터 1씩 증가하면서 5보다 작은 동안 ㉓번을 반복 수행한다.  
 ㉓ a[i]의 값을 정수로 출력한다.

결과 25143

#### [문제 17]

㉑, ㉒

#### [문제 18]

WORLD!

- ※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값을 소문자로 world!와 같이 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

#### 해설

```
public class hrdkorea {
    public static void main(String[] args) {
        ㉑ String str = "HelloWorld!";
        ㉒ String a1 = str.substring(5);
        ㉓ System.out.println(a1.toUpperCase());
    }
}
```

- ㉑ 문자열 변수 str을 선언하고 "Hello World!"로 초기화한다.  
 ㉒ 문자열 변수 a1을 선언하고 str에서 위치값이 5 이상인 모든 문자열, 즉 "World!"로 초기화한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
str	H	e	l	l	o	W	o	r	l	d	!

- **substring(위치값)** : 문자열에서 지정한 위치값을 포함한 이후 모든 문자열을 반환함
- ㉓ 대문자로 변환한 a1의 값을 출력한다.
- **toUpperCase()** : 문자열을 대문자로 변환한다.

결과 WORLD!

#### [문제 19]

UNION

#### [문제 20]

916

- ※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이를 쉼표(,)로 구분하여 9,1,6과 같이 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```
#include <stdio.h>
main( )
{
  ❶ int a[10] = {3, 7, 9, 4, 5, 1, 8, 2, 6, 10};
  ❷ int i;
  ❸ for (i = 0; i < 10; i++)
  {
    ❹ if (i % 3 == 2)
    {
      ❺ printf("%d", a[i]);
    }
  }
}
```

❶ 10개의 요소를 갖는 정수형 배열 a를 선언하고 초기화한다.

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

❷ 정수형 변수 i를 선언한다.

❸ 반복 변수 i가 0부터 1씩 증가하면서 10보다 작은 동안 ❹~❺번을 반복 수행한다.

❹ i를 3으로 나눈 나머지가 2이면 ❺번을 수행하고 그렇지 않으면 반복문의 처음인 ❸번으로 이동한다.

❺ a[i]의 값을 정수로 출력한다.

※ 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	i%3	a[i]	출력
0	0	3	
1	1	7	
2	2	9	9
3	0	4	
4	1	5	
5	2	1	91
6	0	8	
7	1	2	
8	2	6	916
9	0	10	
10			



## 수험자 유의사항

1. 시험 문제지를 받는 즉시 응시하고자 하는 종목의 문제지가 맞는지를 확인하여야 합니다.
2. 시험 문제지 총면수 · 문제번호 순서 · 인쇄상태 등을 확인하고, 수험번호 및 성명을 답안지에 기재하여야 합니다.
3. 문제 및 답안(지), 채점기준은 일절 공개하지 않으며 자신이 작성한 답안, 문제 내용 등을 수험표 등에 이기( 옮겨 적는 행위 ) 등은 관련 법 등에 의거 불이익 조치 될 수 있으니 유의하시기 바랍니다.
4. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 필기구만 사용하되, 흑색을 제외한 유색 필기구 또는 연필류를 사용하였을 경우 그 문항은 0점 처리됩니다.
5. 답란(답안 기재란)에는 문제와 관련 없는 불필요한 낙서나 특이한 기록사항 등을 기재하여서는 안되며 부정의 목적으로 특이한 표식을 하였다고 판단될 경우에는 모든 문항이 0점 처리됩니다.
6. 답안을 정정할 때에는 반드시 정정부분을 두 줄(=)로 그어 표시하여야 하며, 두 줄로 긋지 않은 답안은 정정하지 않은 것으로 간주합니다. (수정테이프, 수정액 사용불가)
7. 답안의 한글 또는 영문의 오타자는 오답으로 처리됩니다. 단, 답안에서 영문의 대 · 소문자 구분, 띄어쓰기는 여부에 관계 없이 채점합니다.
8. 계산 또는 디버깅 등 계산 연습이 필요한 경우는 <문제> 아래의 연습란을 사용하시기 바라며, 연습란은 채점대상이 아닙니다.
9. 문제에서 요구한 가지 수(항수) 이상을 답란에 표기한 경우에는 답안기재 순으로 요구한 가지 수(항수)만 채점하고 한 항에 여러 가지를 기재하더라도 한 가지로 보며 그 중 정답과 오답이 함께 기재란에 있을 경우 오답으로 처리됩니다.
10. 한 문제에서 소문제로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문제로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.
11. 답안은 문제의 마지막에 있는 답란에 작성하여야 합니다.
12. 부정 또는 불공정한 방법(시험문제 내용과 관련된 메모지 사용 등)으로 시험을 치른 자는 부정행위자로 처리되어 당해 시험을 중지 또는 무효로 하고, 2년간 국가기술자격검정의 응시자격이 정지됩니다.
13. 시험위원이 시험 중 신분확인을 위하여 신분증과 수험표를 요구할 경우 반드시 제시하여야 합니다.
14. 시험 중에는 통신기기 및 전자기기(휴대용 전화기 등)를 지참하거나 사용할 수 없습니다.
15. 국가기술자격 시험문제는 일부 또는 전부가 저작권법상 보호되는 저작물이고, 저작권자는 한국산업인력공단입니다. 문제의 일부 또는 전부를 무단 복제, 배포, 출판, 전자출판 하는 등 저작권을 침해하는 일체의 행위를 금합니다.

※ 수험자 유의사항 미준수로 인한 채점상의 불이익은 수험자 본인에게 전적으로 책임이 있음



**문제 1** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
class TestClass {
    int a, b, c;
}

public class Main {
    public static void main(String[] args) {
        TestClass Myint = new TestClass();
        Myint.a = 8;
        Myint.b = 10;
        hrd(Myint);
        System.out.println("a = " + Myint.a);
        System.out.println("b = " + Myint.b);
        System.out.println("c = " + Myint.c);
    }
    static void hrd(TestClass Myint) {
        if (Myint.a++ >= Myint.b--)
            Myint.c = Myint.a - Myint.b;
        else
            Myint.c = Myint.a + Myint.b;
    }
}
```

**답 :**



**문제 2** UNIX 시스템의 구성 요소 중 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기로 시스템과 사용자 간의 인터페이스를 담당하는 역할을 하는 것은 무엇인지 <보기>에서 골라 기호(㉠~㉥)로 쓰시오. (5점)

<보기>

- |          |           |        |        |
|----------|-----------|--------|--------|
| ㉠ 커널     | ㉡ 응용 프로그램 | ㉢ 유틸리티 | ㉣ 운영체제 |
| ㉤ 언어 번역기 | ㉥ 셸       |        |        |

**답 :**



**문제 3** DBMS는 사용자와 데이터베이스 사이에 위치하여 데이터베이스를 관리하고, 사용자의 요구에 따라 정보를 생성해 주는 소프트웨어를 말한다. 이러한 DBMS를 관리하고 운영에 관한 모든 것을 책임지는 사람이나 그룹을 무엇이라고 하는지 영문 약어로 쓰시오. (5점)

1641003



**답 :**

**문제 4** 호스트의 논리적인 IP 주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC Address)로 바꾸는 역할을 하는 프로토콜의 명칭을 쓰시오. (5점)

1641004



**답 :**

**문제 5** OSI 7계층 중 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성을 정의한 계층을 쓰시오. (5점)

1641005



**답 :**

**문제 6** 데이터베이스 정의어(DDL; Data Define Language) 중 데이터베이스 개체를 삭제하는 명령어를 쓰시오. (5점)

1641006



**답 :**

**문제 7** 다음은 <emp> 테이블을 사용하여 고유한 특성을 갖는 'empidx'라는 이름의 인덱스를 정의하되, 'empid' 속성에 대해 내림차순으로 정렬하는 SQL문이다. 괄호에 들어갈 알맞은 명령어를 쓰시오. (5점)

1641007



<SQL문>

```
CREATE UNIQUE INDEX empidx (      ) emp(empid DESC);
```

**답 :**



**문제 8** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
public class hrdkorea {
    public static void main(String[] args) {
        int[] arr1 = {3, 5, 9, 2, 1, 5};
        atest at = new atest();
        at.pa(arr1, 2);
    }
}

class atest {
    void pa(int[] arr, int c) {
        int i;
        if (c > 1) {
            for (i = 0; i < arr.length; i++) {
                if ( (i + 1) % c == 0 ) {
                    System.out.printf("%d", arr[i]);
                }
            }
        }
        else {
            for(i = 0; i < arr.length; i++) {
                System.out.printf("%d", arr[i]);
            }
        }
    }
}
```

**답 :**

**문제 9** 다음 C 언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

1641009



```
#include <stdio.h>
main() {
    for (int i = 9; i > 0; i--) {
        switch(i % 2) {
            case 1:
                printf("%d", i);
                break;
            default:
                printf("*");
                break;
        }
    }
    return 0;
}
```

**답 :**

**문제 10** 테이프 아카이브에서 유래된 것으로, 유닉스와 리눅스에서 여러 파일을 묶거나 묶인 파일을 풀 때 사용하는 명령어가 무엇인지 <보기>에서 찾아 기호(㉠~㉤)로 쓰시오. (5점)

1641010



<보기>

- |        |         |      |
|--------|---------|------|
| ㉠ kill | ㉡ tar   | ㉢ vi |
| ㉣ vd   | ㉤ chmod |      |

**답 :**

1641011



**문제 11** 다음 C 언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
#define _USE_MATH_DEFINES
#include <stdio.h>
#include <math.h>
main() {
    printf("%.2f", M_PI);
    return 0;
}
```

**답 :**

1641012



**문제 12** 다음의 <학생> 테이블에 대해 차수(Degree)와 카디널리티(Cardinality)의 수를 쓰시오. (5점)

<학생> 테이블

학번	이름	학과	학년	성적
19001	홍길동	전산	4	90
20002	이정재	수학	3	85
19012	신보라	전산	4	95
22015	도요새	영어	1	76

**답**

- ① 차수(Degree)의 수 :
- ② 카디널리티(Cardinality)의 수 :

1641013



**문제 13** 유닉스(UNIX)의 명령어 중 현재 작업중인 디렉터리의 경로를 화면에 출력하는 명령어가 무엇인지 쓰시오. (5점)

**답 :**

**문제 14** 다음 C 언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

1641014



```
#include <stdio.h>
main() {
    int i, j;
    int sum = 0;
    for (i = 0; i <= 4; i++) {
        for (j = 0; j <= i; j++) {
            if (j % 2 == 0)
                sum += j;
            else
                sum++;
        }
    }
    printf("%d", sum);
}
```

**답 :**

**문제 15** OSI 7계층 중 응용 계층에 해당하는 프로토콜을 <보기>에서 찾아 쓰시오. (5점)

1641015



<보기>

HTTP, TCP, UDP, ICMP, FTP, IPsec

**답 :**

**문제 16** 다음이 설명하는 용어를 쓰시오. (5점)

1641016



- 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위를 의미한다.
- 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다.
- 원자성, 일관성, 독립성, 영속성의 특성을 갖는다.

**답 :**

1641017



**문제 17** 데이터베이스 언어 중 데이터의 보안, 무결성, 회복, 병행 제어 등을 정의하는 데 사용되는 것으로, 데이터베이스 사용자에게 권한을 부여하거나 취소하기 위해 사용하는 언어를 영문 약어로 쓰시오. (5점)

**답 :**

1641018



**문제 18** 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (5점)

```
public class JavaEX {
    public static void main(String[] args) {
        int n;
        n = 10;
        increase(n);
        System.out.printf("%d", n);
    }

    static void increase(int n) {
        n = n + 1;
    }
}
```

**답 :**

1641019



**문제 19** 다음 설명에서 괄호 안에 들어갈 알맞은 용어를 쓰시오. (5점)

소프트웨어 ( )은/는 소프트웨어 개발 과정에서 소프트웨어의 변경 사항을 관리하기 위해 개발된 일련의 활동이다. 대표적인 도구에는 Git, CSV, Subversion 등이 있다.

**답 :**



**문제 20** 사원(사원번호, 이름, 부서, 연락처) 테이블에서 '부서'가 "총무부"인 사원들의 '사원번호', '이름', '연락처'를 가져와 '사번', '성명', '연락처'로 표시하는 '총무부사원' 뷰를 정의하는 SQL문이다. 괄호에 들어갈 알맞은 명령을 쓰시오. (5점)

〈SQL문〉

```
CREATE VIEW 총무부사원 (사번, 성명, 연락처)
(      ) (사원번호, 이름, 연락처)
FROM 사원
WHERE 부서 = '총무부';
```

**답 :**

## [문제 1]

a = 9

b = 9

c = 18

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 줄바꿈 없이 a = 9, b = 9, c = 18 또는 a = 9 b = 9 c = 18로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

## 해설

```

class TestClass {                클래스 TestClass를 정의한다.
    int a, b, c;                  클래스 TestClass에는 정수형 변수 a, b, c가 선언되어 있다.
}

public class Main {
    public static void main(String[] args) {
        ❶ TestClass Myint = new TestClass();
        ❷ Myint.a = 8;
        ❸ Myint.b = 10;
        ❹ hrd(Myint);
        ❸ System.out.println("a = " + Myint.a);
        ❹ System.out.println("b = " + Myint.b);
        ❺ System.out.println("c = " + Myint.c);
    }
    ❻ static void hrd(TestClass Myint) {
        ❼ if (Myint.a++ >= Myint.b--)
            Myint.c = Myint.a - Myint.b;
        else
        ❼ Myint.c = Myint.a + Myint.b;
    }
}

```

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

❶ 클래스 TestClass의 객체 변수 Myint를 선언한다.

	int a	int b	int c
객체 변수 Myint			

❷ 객체 변수 Myint의 변수 a에 8을 저장한다.

❸ 객체 변수 Myint의 변수 b에 10을 저장한다.

	int a	int b	int c
객체 변수 Myint	8	10	

❹ 객체 변수 Myint의 시작 주소를 인수로 하여 hrd 메소드를 호출한다.

❺ 빈칸 값이 없는 hrd() 메소드의 시작점이다. ❶번에서 전달받은 객체 변수의 주소는 Myint가 받는다.

※ 객체 변수의 이름은 객체 변수의 시작 주소를 가리키므로, 인수로 전달하는 경우 메소드에서 변경된 값이 main()의 객체 변수에도 적용된다는 점을 염두에 두세요.

❻ Myint의 a가 Myint의 b보다 크거나 같으면 다음 문장을 수행하고, 그렇지 않으면 ❷번을 수행한다. Myint의 a의 값 8은 Myint의 b의 값 10보다 작으므로 ❷번으로 이동한다.

※ Myint.a++과 Myint.b--는 후치 연산자이므로 조건을 비교한 후 Myint의 a의 값은 10이 증가하여 9가 되고 Myint의 b의 값은 10이 감소하여 9가 된다.



	int a	int b	int c
객체 변수 Myint	9	9	

⑦ Myint의 c에 Myint의 a와 b를 더한 값을 저장한다. Myint의 c에는 18이 저장된다. 메소드가 종료되었으므로 메소드를 호출했던 ④번의 다음 줄인 ⑧번으로 이동한다.

	int a	int b	int c
객체 변수 Myint	9	9	18

⑧ a = 를 출력하고 Myint의 a의 값 9를 출력한 후 다음 줄의 처음으로 커서를 이동시킨다.

결과    a = 9

⑨ b = 를 출력하고 Myint의 b의 값 9를 출력한 후 다음 줄의 처음으로 커서를 이동시킨다.

결과    a = 9  
b = 9

⑩ c = 를 출력하고 Myint의 c의 값 18을 출력한 후 다음 줄의 처음으로 커서를 이동시킨다.

결과    a = 9  
b = 9  
c = 18

[문제 2]

Ⓣ

[문제 3]

DBA

[문제 4]

※ 다음 중 하나를 쓰면 됩니다.

ARP, Address Resolution Protocol

[문제 5]

물리 계층

[문제 6]

DROP

[문제 7]

ON

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

해설

CREATE UNIQUE INDEX empidx  
ON emp(empid DESC);

고유한 특성을 갖는 'empidx'라는 이름의 인덱스를 생성한다.  
(emp) 테이블을 사용하되, 'empid' 속성을 기준으로 내림차순 정렬한다.

[문제 8]

525

※ 답안 작성 시 주의 사항 : 프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이를 쉼표(.)나 공백으로 구분하여 5,2,5 또는 5 2 5와 같이 쓴 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

해설

```

public class hrdkorea {
    public static void main(String[] args) {
        ❶ int[] arr1 = {3, 5, 9, 2, 1, 5};
        ❷ atest at = new atest();
        ❸ at.pa(arr1, 2);
    } ❹
}

class atest {
    ❺ void pa(int[] arr, int c) {
        ❻ int i;
        ❼ if (c > 1) {
            ❽ for (i = 0; i < arr.length; i++) {
                ❾ if ( (i + 1) % c == 0 ) {
                    System.out.printf("%d", arr[i]);
                }
            }
        }
        else {
            for (i = 0; i < arr.length; i++) {
                System.out.printf("%d", arr[i]);
            }
        } ❿
    }
}

```

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

❶ 6개의 요소를 갖는 정수형 배열 arr1을 선언하고 초기화한다. 개수를 지정하지 않으면, 초기값으로 지정된 수만큼 배열의 요소가 만들어진다.

	[0]	[1]	[2]	[3]	[4]	[5]
arr1	3	5	9	2	1	5

❷ 클래스 atest의 객체 변수 at를 선언한다.

❸ 배열 arr1의 시작 주소와 2를 인수로 하여 atest의 pa() 메소드를 호출한다.

❹ 반환 값이 없는 pa 메소드의 시작점이다. ❸번에서 전달받은 배열 arr1의 주소는 배열 arr이 받고 2는 c가 받는다.

※ 배열의 이름은 배열의 시작 주소를 가리키므로, 인수로 전달하는 경우 메소드에서 변경된 값이 main()의 배열에도 적용된다는 점을 염두에 두세요.

	[0]	[1]	[2]	[3]	[4]	[5]		c
arr	3	5	9	2	1	5		2

❺ 정수형 변수 i를 선언한다.

❻ c의 값이 1보다 크면 ❽번으로 이동하고, 그렇지 않으면 else의 다음 문장으로 이동한다. c의 값 2는 1보다 크므로 ❽번으로 이동한다.

❼ 반복 변수 i가 0부터 1씩 증가하면서 배열 arr의 길이인 6보다 작은 동안 ❽~❾번을 반복 수행한다.

❽ i+1을 c로 나눈 나머지가 0이면, 즉 i+1이 짝수이면 arr1 배열의 i번째에 해당하는 값을 출력하기 위해 ❾번으로 이동하고, 그렇지 않으면 반복문의 처음인 ❽번으로 이동한다.

❾ arr[i]의 값을 정수로 출력한다. 반복문이 종료되면 제어기가 ❿번으로 이동하고, 메소드가 종료되었으므로 메소드를 호출했던 ❸번의 다음 줄인 ❹번으로 이동하여 프로그램이 종료된다.

※ 반복문 실행에 따른 변수들의 변화는 다음과 같다.

arr	c	i	출력
[0] [1] [2] [3] [4] [5]	2	0	
3 5 9 2 1 5		1	5
		2	

	3	52
	4	
	5	525
	6	

## [문제 9]

9\*7\*5\*3\*1

## 해설

```
#include <stdio.h>
main() {
    ① for (int i = 9; i > 0; i--) {
        ② switch(i % 2) {
            ③ case 1:
                ④ printf("%d", i);
                ⑤ break;
            ⑥ default:
                ⑦ printf("");
                ⑧ break;
        }
    }
    ⑨ return 0;
}
```

① 반복 변수 i가 9부터 1씩 감소하면서 0보다 큰 동안 ②~⑧번을 반복 수행한다.

② i를 2로 나눈 나머지에 해당하는 숫자를 찾아간다.

③ 'i%2'가 1일 경우 찾아오는 곳이다.

④ i의 값을 정수로 출력한다.

⑤ switch문을 벗어나 반복문의 처음인 ①번으로 이동한다.

⑥ case 1에 해당되지 않는 경우 찾아오는 곳이다.

⑦ \*을 출력한다.

⑧ switch문을 벗어나 반복문의 처음인 ①번으로 이동한다.

⑨ main() 함수에서의 **return 0;**은 프로그램의 종료를 의미한다.

※ 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	c	출력
9	1	9
8	0	9*
7	1	9*7
6	0	9*7*
5	1	9*7*5
4	0	9*7*5*
3	1	9*7*5*3
2	0	9*7*5*3*
1	1	9*7*5*3*1
0		

## [문제 10]

㉠

## [문제 11]

3.14

## 해설

```

#define _USE_MATH_DEFINES    <math.h>에 정의되어 있는 수학 상수를 사용하기 위한 정의문
#include <stdio.h>           데이터의 입·출력에 사용되는 기능을 제공하는 헤더 파일
#include <math.h>            수학 함수를 제공하는 헤더 파일
main() {
    ① printf("%.2f", M_PI);
    ② return 0;
}

```

- ① 상수 `M_PI`의 값을 출력하되, 정수 부분은 모두 출력하고 소수점 이하는 3자리에서 반올림하여 2자리까지만 출력한다.

결과 3.14

※ `M_PI`: 원주율( $\pi$ )의 값 3.1415926535... 을 저장하고 있는 상수로, `math.h` 헤더 파일에 정의되어 있음

- ② `main()` 함수에서의 **return 0;**은 프로그램의 종료를 의미한다.

## [문제 12]

① 5    ② 4

## [문제 13]

PWD

## [문제 14]

16

## 해설

```

#include <stdio.h>
main() {
    ① int i, j;
    ② int sum = 0;
    ③ for (i = 0; i <= 4; i++) {
    ④     for (j = 0; j <= i; j++) {
    ⑤         if (j % 2 == 0)
    ⑥             sum += j;
    ⑦         else
    ⑧             sum++;
    ⑨     }
    ⑩ }
    ⑪ printf("%d", sum);
}

```

- ① 정수형 변수 `i`와 `j`를 선언한다.

- ② 정수형 변수 `sum`을 선언하고 0으로 초기화한다.

- ③ 반복 변수  $i$ 가 0부터 1씩 증가하면서 4보다 작거나 같은 동안 ④~⑦번을 반복 수행한다.  
 ④ 반복 변수  $j$ 가 0부터 1씩 증가하면서  $i$ 보다 작거나 같은 동안 ⑤~⑦번을 반복 수행한다.  
 ⑤  $j$ 를 2로 나눈 나머지가 0이면 ⑥번을 수행하고, 아니면 ⑦번을 수행한다.  
 ⑥  $\text{sum} = \text{sum} + j$ ;와 동일하다.  $\text{sum}$ 에  $j$ 의 값을 누적시킨다.  
 ⑦  $\text{sum} = \text{sum} + 1$ ;과 동일하다.  $\text{sum}$ 의 값을 1씩 누적시킨다.  
 ⑧  $\text{sum}$ 의 값을 정수로 출력한다.
- ※ 반복문 실행에 따른 변수들의 변화는 다음과 같다.

$i$	$j$	$j \% 2$	$\text{sum}$
0	0	0	0
	1	0	0
1	0	0	0
	1	1	1
	2	0	1
2	0	0	1
	1	1	2
	2	0	4
	3	1	4
3	0	0	4
	1	1	5
	2	0	7
	3	1	8
4	0	0	8
	1	1	9
	2	0	11
	3	1	12
	4	0	16
5			

## [문제 15]

HTTP, FTP

## [문제 16]

※ 다음 중 하나를 쓰면 됩니다.

트랜잭션, Transaction

## [문제 17]

DCL

## [문제 18]

10

해설

```
public class JavaEX {  
    public static void main(String[] args) {  
        ❶ int n;  
        ❷ n = 10;  
        ❸ increase(n);  
        ❹ System.out.printf("%d", n);  
    }  
  
    ❺ static void increase(int n) {  
        ❻ n = n + 1;  
    } ❼  
}
```

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

- ❶ 정수형 변수 n을 선언한다.
- ❷ n에 10을 저장한다.
- ❸ n의 값 10을 인수로 하여 increase() 메소드를 호출한다.
- ❹ 반환 값이 없는 increase() 메소드의 시작점이다. ❸번에서 전달받은 10을 n이 받는다.
- ❺ n의 값을 1 증가시킨다.
- ❻ increase() 메소드가 종료된 후 메소드를 호출했던 ❸번의 다음 줄인 ❹번으로 이동한다.
- ❼ n의 값을 정수로 출력한다. increase() 메소드의 변수 n은 main() 메소드의 변수 n과 별개의 변수이므로 ❺번에서 증가한 값이 main() 메소드의 변수 n에 영향을 주지 못한다. 그러므로 여기에서는 ❷번에서 저장했던 10이 출력되게 된다.

결과 10

[문제 19]

※ 다음 중 하나를 쓰면 됩니다.

형상관리, SCM, Software Configuration Management

[문제 20]

AS SELECT

※ 답안 작성 시 주의 사항 : SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

해설

CREATE VIEW 총무부사원 (사번, 성명, 연락처)	생성한 뷰의 이름은 <총무부사원>으로, '사번', '성명', '연락처'로 속성을 표시한다.
AS SELECT (사번, 성명, 연락처)	'사원번호', '이름', '연락처' 속성을 가져온다.
FROM 사원	<사원> 테이블에서 속성을 가져온다.
WHERE 부서 = '총무부';	'부서'가 "총무부"인 자료만을 대상으로 한다.