



시험에 나오는것만 공부한다!

시나공시리즈

2020년 실기 시험 대비 정보처리기사 실기 기출문제 02회



정보처리기사 실기 시험은 한국산업인력공단에서 문제를 공개하지 않아 문제 복원에 많은 어려움이 있습니다. 다음에 제시된 문제는 시험을 치른 학생들의 기억을 토대로 복원한 것이므로, 일부 내용이나 문제별 배점이 실제 시험과 다를 수 있음을 알립니다.

저작권 안내

이 자료는 시나공 카페 회원을 대상으로 하는 자료로서 개인적인 용도로만 사용할 수 있습니다. 허락 없이 복제하거나 다른 매체에 옮겨 실을 수 없으며, 상업적 용도로 사용할 수 없습니다.

*** 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. 답안지 작성은 반드시 검정색 필기구만 사용하여야 합니다. (그 외 연필류, 유색 필기구 등을 사용한 답안은 채점하지 않으며 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 굿지 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 <문 제> 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 메모지사용 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)

```
#include <stdio.h>
main() {
    int i, j;
    for (i = 2; i <= 4; i++) {
        for (j = 5; j <= 7; j++) {
            }
        }
    printf("%d × %d = %2d", j, i, i * j);
}
```

답 :

문제 2 다음 기능 명세서를 보고 결함의 판단 기준에 해당하는 것을 모두 고르시오. (5점)

- ① 기능 명세서에 가능하다고 명시된 기능이 수행되지 않는 경우
- ② 기능 명세서에 불가능하다고 명시된 기능이 수행되지 않는 경우
- ③ 기능 명세서에 명시되어 있지 않지만, 기능이 수행되지 않는 경우
- ④ 기능 명세서에 명시되어 있지 않지만, 수행해야만 하는 기능이 수행되지 않는 경우
- ⑤ 테스트 시각에서 보았을 때, 문제가 있다고 판단되는 경우

답 :

문제 3 “파일 탐색기” 기능을 수행하기 위한 Windows 10의 단축키를 쓰시오. (5점)

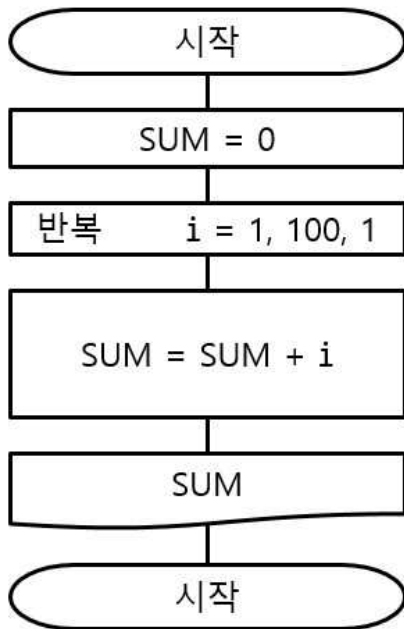
답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 4 다음 <순서도>와 <코드>는 1~100까지 합을 구하는 프로그램이다. 프로그램을 분석하여 괄호 안에 들어갈 알맞은 답을 쓰시오. (5점)

<순서도>



<코드>

```

public class Test {
    public static void main(String[] args) {
        int sum = 0;
        for ( (      ) )
            sum = sum + i;
        System.out.printf("%d", sum);
    }
}
  
```

답 :

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

```

#include <stdio.h>
hrd(num) {
    if (num <= 0)
        return;
    printf("%d ", num);
    hrd(num - 1);
}
main() {
    hrd(5);
    return 0;
}
  
```

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 6 다음 <사원> 릴레이션의 카디널리티(Cardinality) 개수를 구하시오. (5점)

<사원>

사번	성명	부서	직급	수당
11	김예소	기획부	대리	30
32	고강민	인사부	과장	40
41	이향기	재무부	부장	50

답 :

문제 7 소프트웨어가 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하는 테스트로, 동치 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등의 테스트가 여기에 속하며, 테스트 과정의 후반부에 사용자의 요구사항 명세를 보며 구현된 기능들을 검사하는 테스트를 가리키는 용어를 쓰시오. (5점)

답 :

문제 8 클라이언트와 서버 사이에서 요청과 응답을 주고받기 위해 사용되는 프로토콜로, 1989년 영국의 컴퓨터 과학자 버너스리(Berners-Lee)가 WWW(World Wide Web)를 고안하면서 설계하였다. 주로 HTML(Hyper Text Markup Language)로 이루어진 하이퍼텍스트 문서를 전송하기 위해 사용되는 이 통신규약의 이름을 쓰시오. (5점)

답 :

문제 9 다음은 TCP/IP 계층 프로토콜이다. 괄호에 들어갈 알맞은 답을 쓰시오. (5점)

응용 계층
() 계층
인터넷 계층
네트워크 인터페이스 계층

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 10 테이블의 구조를 변경하는 DDL(Data Define Language) 명령어를 쓰시오. (5점)

답 :

문제 11 다음은 SELECT문의 실행 작동 순서를 나열한 것이다. 괄호(①~③)에 알맞은 명령을 보기에서 찾아 쓰시오. (5점)

FROM → (①) → (②) → (③) → SELECT → ORDER BY

<보기>

HAVING, GROUP BY, WHERE

답

- ① :
- ② :
- ③ :

문제 12 <도서>, <도서가격> 테이블을 참고하여 다음 SQL문의 실행 결과를 쓰시오. (5점)

SELECT 가격 FROM 도서가격
WHERE 책번호 = (SELECT 책번호 FROM 도서 WHERE 책명 = '데이터베이스');

<도서>

책번호	책명
1111	데이터베이스
2222	운영체제
3333	자료구조

<도서가격>

책번호	가격
1111	18000
2222	20000
3333	10000
4444	15000

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 13 XML의 대안으로 등장한 자바 스크립트(JavaScript) 기반의 기술로, 속성-값 쌍(Attribute-Value Pairs) 형태로 이루어진 구조적인 데이터의 교환을 위해 사람이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 웹 브라우저와 웹 서버 간의 비동기 통신, 웹 서버 간의 데이터 교환 등에 주로 사용되는 이 기술의 이름을 쓰시오. (5점)

답 :

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

```
#include <stdio.h>
int hrdcompare(num1, num2);
main() {
    printf("%d", hrdcompare(10, 23) + hrdcompare(35, 19));
}
hrdcompare(int num1, int num2) {
    if (num1 > num2)
        return num1;
    else
        return num2;
}
```

답 :

문제 15 다음 결함 내용 중 Fixed의 의미를 쓰시오. (5점)

No	내용	상태
1	로그인 시 비밀번호에 특수문자가 섞인 경우 인증이 거부되는 현상	Fixed
2	회원가입 시 중복인증 여부가 체크되지 않는 현상	Assigned
3	페이지 이동 시 순간적으로 BLANK 페이지가 노출되는 현상	Open
4	모바일에서 팝업 창 로드 시 이미지가 깨지는 현상	Assigned
5	링크에 마우스 오버 시 마우스 포인터가 바뀌지 않는 현상	Open

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 16 <HRD> 테이블의 '주민번호' 속성에 "900351"을 입력할 때 오류가 발생하여 <HRD> 테이블의 '주민번호' 속성의 크기를 6으로 변경하는 SQL문을 작성하였다. 괄호에 알맞은 명령을 쓰시오. (5점)

<테이블 생성 SQL문>

```
CREATE TABLE HRD
  주민번호 NUMBER(5);
```

<데이터 입력 SQL문> - 오류 발생

```
INSERT INTO HRD (주민번호) VALUES ('900351');
```

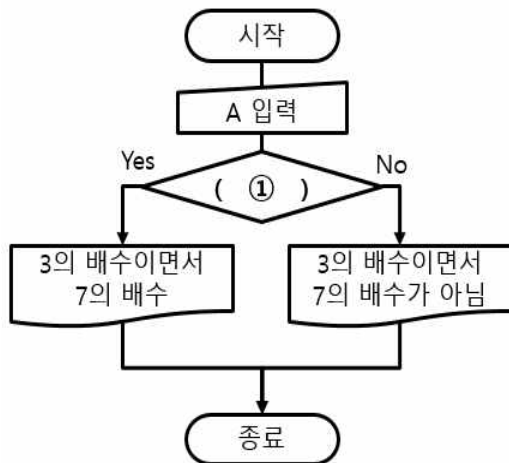
<테이블 속성 변경 SQL문>

```
ALTER TABLE HRD (      ) 주민번호 NUMBER(6);
```

답 :

문제 17 다음 <순서도>와 <코드>는 정수 값을 입력받아 값이 3의 배수이면서 7의 배수인지 판단하는 프로그램이다. 프로그램을 분석하여 괄호(①, ②)에 해당하는 각 답을 쓰시오. (5점)

<순서도>



<코드>

```
#include <stdio.h>
main() {
    int a;
    printf("정수 입력 : ");
    scanf("%d", &a);
    if ( ② )
        printf("3의 배수이면서 7의 배수임\n");
    else
        printf("3의 배수이면서 7의 배수가 아님\n");
}
```

답

- ① :
- ② :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

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

```
#include <stdio.h>
main() {
    int x = 4;
    printf("%d", x++);
}
```

답 :

문제 19 다음은 <topic> 테이블과 <auth> 테이블을 결합하여 검색하는 <SQL>문이다. 괄호에 알맞은 명령을 쓰시오. (5점)

<topic>

id	title1	title2	auth_id
1	sql	sql is	1
2	oracle	oracle is	1
3	sql-server	sql-server is	3
4	progress	progress is	2
5	mongoDB	mongoDB is	1

<auth>

id	name	job
1	kim	develope
2	shin	database administrator
3	park	scientist, develope

<SQL>

SELECT * FROM topic LEFT () auth ON topic.auth_id = auth.id;

<결과>

id	title1	title2	auth_id	id	name	job
1	sql	sql is	1	1	kim	develope
2	oracle	oracle is	1	1	kim	develope
3	sql-server	sql-server is	3	3	park	scientist, develope
4	progress	progress is	2	2	shin	database administrator
5	mongoDB	mongoDB is	1	1	kim	develope

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 20 CUI(Command User Interface)는 명령어를 직접 입력해서 명령을 수행하는 방식으로 DOS에서 주로 사용한다. 반면 마우스로 아이콘이나 메뉴를 선택하여 명령을 수행하는 방식으로 주로 Windows에서 사용하는 인터페이스 방식을 가리키는 용어를 쓰시오. (5점)

답 :

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

연 습 란

기출문제 정답

[문제 1]

$8 \times 5 = 40$

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 서식지정자를 무시하고 8, 5, 40 또는 8 5 40으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[해설]

```
#include <stdio.h>
main() {
    int i, j;
    ❶ for (i = 2; i <= 4; i++) {
    ❷     for (j = 5; j <= 7; j++) {
    ❸     }
    ❹ }
    ❺ printf("%d × %d = %2d", j, i, i * j);
}
```

- ❶ 반복 변수 i 가 2에서 시작하여 1씩 증가하면서 4보다 작거나 같은 동안 ❷번 문장을 반복하여 수행한다. 결국 ❷번을 3번 반복하여 수행한다.
- ❷ 반복 변수 j 가 5에서 시작하여 1씩 증가하면서 7보다 작거나 같은 동안 반복하는데 실행할 문장 없으므로 for문만 3번 반복하여 수행한다.
- ❸ ❷번 반복문의 끝이다.
- ❹ ❶번 반복문의 끝이다.
- ❺ " $j \times i = i \times j$ "의 형태로 변수의 값을 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	j
2	5
	6
	7
	8
3	5
	6
	7
	8
4	5
	6
	7
	8
5	

[문제 2]

❶, ❹, ❺

[문제 3]

Windows 로고 키 + E

[문제 4]

```
int i = 1; i <= 100; i++
```

※ 답안 작성 시 주의 사항

증가식(i++)은 i가 1씩 증가하는 어떠한 식도 정답이 될 수 있습니다. 예를 들어 `i = i + 1`로 작성하거나, 대입 연산자를 사용하여 `i += 1`, 또는 전치 증가 연산자를 사용하여 `++i`로 작성해도 답이 됩니다.

[해설]

```
public class Test {  
    public static void main(String[] args) {  
        ❶ int sum = 0;  
        ❷ for (int i = 1; i <= 100; i++)  
        ❸     sum = sum + i;  
        ❹ System.out.printf("%d", sum);  
    }  
}
```

❶ 정수형 변수 sum을 선언하고 0으로 초기화한다.

❷ 반복 변수 i가 1에서 시작하여 1씩 증가하면서 100보다 작거나 같은 동안 ❸번 문장을 반복 수행한다.

❸ sum에 i의 값을 누적한다.

❹ sum의 값을 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	sum
1	0
2	1
3	3
4	6
5	10
6	15
7	21
⋮	28
96	⋮
97	4656
98	4753
99	4851
100	4950
101	5050

[문제 5]

5 4 3 2 1

※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 코마를 넣어 5, 4, 3, 2, 1로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

[해설]

```
main( ) {  
  ❶ hrd(5);  
    return 0;  
}
```

❶ 5를 인수로 하여 hrd 함수를 호출한다.

①회

```
hrd(num) {  
  ❷ if (num <= 0)  
    return;  
  ❸ printf( "%d " , num);  
  ❹ hrd(num-1);  
}
```

hrd 함수가 호출될 때 5를 전달받았으므로 num은 5이다. ❷의 조건을 만족하지 않으므로 ❸을 수행한다.

❸ num의 값 5를 출력하고 한 칸을 띄운다.

결과 **5**

❹를 수행하기 위해 hrd 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(4)인 상태로 호출된다.

②회

```
hrd(num) {  
  ❺ if (num <= 0)  
    return;  
  ❻ printf( "%d " , num);  
  ❼ hrd(num-1);  
}
```

hrd 함수가 호출될 때 4를 전달받았으므로 num은 4이다. ❺의 조건을 만족하지 않으므로 ❻을 수행한다.

❻ num의 값 4를 출력하고 한 칸을 띄운다.

결과 **5 4**

❼을 수행하기 위해 hrd 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(3)인 상태로 호출된다.

③회

```
hrd(num) {  
  ❽ if (num <= 0)  
    return;  
  ❾ printf( "%d " , num);  
  ❿ hrd(num-1);  
}
```

hrd 함수가 호출될 때 3을 전달받았으므로 num은 3이다. ❽의 조건을 만족하지 않으므로 ❾를 수행한다.

❾ num의 값 3을 출력하고 한 칸을 띄운다.

결과 **5 4 3**

❿를 수행하기 위해 hrd 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(2)인 상태로 호출된다.

④회

```
hrd(num) {  
  ⑪ if (num <= 0)  
    return;  
  ⑫ printf( "%d " , num);  
  ⑬ hrd(num-1);  
}
```

hrd 함수가 호출될 때 2를 전달받았으므로 num은 2이다. ⑪의 조건을 만족하지 않으므로 ⑫를 수행한다.

⑫ num의 값 2를 출력하고 한 칸을 띄운다.

결과 **5 4 3 2**

⑬을 수행하기 위해 hrd 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(1)인 상태로 호출된다.

⑤회

```
hrd(num) {  
  ⑭ if (num <= 0)  
    return;  
  ⑮ printf( "%d " , num);  
  ⑯ hrd(num-1);  
}
```

hrd 함수가 호출될 때 1을 전달받았으므로 num은 1이다. ⑭의 조건을 만족하지 않으므로 ⑮를 수행한다.

⑮ num의 값 1을 출력하고 한 칸을 띄운다.

결과 **5 4 3 2 1**

⑯을 수행하기 위해 hrd 함수를 호출하는데, 호출할 때 전달되는 값은 num-1이므로 hrd(0)인 상태로 호출된다.

⑥회

```
hrd(num) {  
  ⑰ if (num <= 0)  
    ⑱ return;  
  printf( "%d " , num);  
  hrd(num-1);  
}
```

hrd 함수가 호출될 때 0을 전달받았으므로 num은 0이다. ⑰의 조건을 만족하므로 ⑱을 수행한다.

'return;'이므로 함수의 실행을 종료하고 반환값 없이 제어를 ⑤회 hrd(0) 함수를 호출했던 곳으로 옮긴다.

⑤회

```
hrd(num) {  
  if (num <= 0)  
    return;  
  printf( "%d " , num);  
  hrd(num-1);  
} ⑲
```

⑲ 함수의 실행을 종료하고 반환값 없이 제어를 ④회 hrd(1) 함수를 호출했던 곳으로 옮긴다.

```

hrd(num) {
    if (num <= 0)
        return;
    printf( "%d " , num);
    hrd(num-1);
} ㉔

```

④회

㉔ 함수의 실행을 종료하고 반환값 없이 제어를 ③회 hrd(2) 함수를 호출했던 곳으로 옮긴다.

```

hrd(num) {
    if (num <= 0)
        return;
    printf( "%d " , num);
    hrd(num-1);
} ㉓

```

③회

㉓ 함수의 실행을 종료하고 반환값 없이 제어를 ②회 hrd(3) 함수를 호출했던 곳으로 옮긴다.

```

hrd(num) {
    if (num <= 0)
        return;
    printf( "%d " , num);
    hrd(num-1);
} ㉒

```

②회

함수의 실행을 종료하고 반환값 없이 제어를 ①회 hrd(4) 함수를 호출했던 곳으로 옮긴다.

```

hrd(num) {
    if (num <= 0)
        return;
    printf( "%d " , num);
    hrd(num-1);
} ㉑

```

①회

함수의 실행을 종료하고 반환 값없이 제어를 처음 hrd(5) 함수를 호출했던 곳으로 옮긴다.

```

main( ) {
    hrd(5);
    ㉑ return 0;
}

```

main 함수에서 'return 0;'는 프로그램 종료를 의미한다.

[문제 6]

3

[답안 작성 방법 안내]

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

[문제 7]

블랙박스 테스트(Black Box Test)

[문제 8]

HTTP(HyperText Transfer Protocol)

[문제 9]

전송(Transport)

[문제 10]

ALTER

[문제 11]

① WHERE ② GROUP BY ③ HAVING

[해설]

SELEC문의 실행 작동 순서는 FROM → WHERE → GROUP BY → HAVING → SELECT → DISTINCT → ORDER BY 순입니다.

[문제 12]

18000

[해설]

하위 질의는 조건절에 주어진 질의(❶)를 먼저 수행하여 그 검색 결과를 조건절의 피연산자로 사용한다.

❷ SELECT 가격 FROM 도서가격 WHERE 책번호 =
❶ (SELECT 책번호 FROM 도서 WHERE 책명 = ‘데이터베이스’);

❶

- SELECT 책번호 : ‘책번호’를 표시한다.
 - FROM 도서 : <도서> 테이블을 대상으로 검색한다.
 - WHERE 책명 = ‘데이터베이스’ : ‘책명’이 “데이터베이스”인 자료만을 대상으로 검색한다.
- ※ ‘책명’이 “데이터베이스”인 도서의 책번호 “1111”이 검색되므로 ❷의 질의문은 SELECT 가격 FROM 도서가격 WHERE 책번호 = ‘1111’이 됩니다.

❷

- SELECT 가격 : ‘가격’을 표시한다.
 - FROM 도서가격 : <도서가격> 테이블을 대상으로 검색한다.
 - WHERE 책번호 = ‘1111’ : ‘책번호’가 “1111”인 자료만을 대상으로 검색한다.
- ※ ‘책번호’가 “1111”인 도서의 가격은 18000입니다.

[문제 13]

JSON(JavaScript Object Notation)

[문제 14]

58

[해설]

```
main() {  
  ❶ printf("%d", hrdcompare(10, 23) + hrdcompare(35, 19));  
}
```

❶ 10과 23을 인수로 hrdcompare 함수를 호출한 다음 돌려받은 값과, 35와 19를 인수로 hrdcompare 함수를 호출한 다음 돌려받은 값을 더하여 화면에 출력한다.

```
  ❷ hrdcompare(int num1, int num2) {  
    ❸ if (num1 > num2)  
      return num1;  
    else  
    ❹ return num2;  
  }
```

❷ hrdcompare 함수의 시작점이다. ❶번에서 전달받은 값 10과 23을 각각 num1과 num2가 받는다.

❸ num1의 값 10이 num2의 값 23보다 크지 않으므로 ❹번으로 이동한다.

❹ num2의 값 23을 반환한다. ❶번의 hrdcompare(10, 23)의 값은 23이 된다.

```
  ❺ hrdcompare(int num1, int num2) {  
    ❻ if (num1 > num2)  
    ❼ return num1;  
    else  
      return num2;  
  }
```

❺ hrdcompare 함수의 시작점이다. ❶번에서 전달받은 값 35와 19를 각각 num1과 num2가 받는다.

❻ num1의 값 35가 num2의 값 19보다 크므로 ❼번으로 이동한다.

❼ num1의 값 35를 반환한다. ❶번의 hrdcompare(35, 19)의 값은 35가 된다.

```
main() {  
  ❽ printf("%d", hrdcompare(10, 23) + hrdcompare(35, 19));  
}
```

❽ hrdcompare(10, 23)의 값 23과 hrdcompare(35, 19)의 값 35를 더한 58을 화면에 출력한다.

[문제 15]

다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

Fixed는 개발자가 필요한 변경 작업을 수행하여 결합 수정 작업을 완료한 상태를 의미한다.

[문제 16]

ALTER 또는 MODIFY

[문제 17]

❶ $(A \text{ MOD } 3) + (A \text{ MOD } 7) = 0$ ❷ $a \% 3 + a \% 7 == 0$

※ 답안 작성 시 주의 사항

A를 3과 7로 나눈 나머지가 모두 0인 어떠한 식도 정답이 될 수 있습니다. 예를 들어 ❷번에서 논리 연산

자를 사용하여 `a%3 == 0 && a%7 == 0`으로 작성해도 답이 됩니다.

[해설]

```
#include <stdio.h>
main() {
  ❶ int a;
  ❷ printf("정수 입력 : ");
  ❸ scanf("%d", &a);
  ❹ if (a % 3 + a % 7 == 0)
  ❺     printf("3의 배수이면서 7의 배수임\n");
  else
  ❻     printf("3의 배수이면서 7의 배수가 아님\n");
}
```

- ❶ 정수형 변수 a를 선언한다.
- ❷ 화면에 정수 입력 : 를 출력한다.
- ❸ 정수를 입력받아 a에 저장한다.
- ❹ a를 3으로 나눈 나머지와 a를 7로 나눈 나머지를 합한 값이 0이면 ❺번을 실행하고, 아니면 ❻번을 실행한다.
- ❺ 화면에 3의 배수이면서 7의 배수임을 출력하고 커서를 다음 줄의 처음으로 옮긴다.
- ❻ 화면에 3의 배수이면서 7의 배수가 아님을 출력하고 커서를 다음 줄의 처음으로 옮긴다.

[문제 18]

4

[해설]

```
#include <stdio.h>
main() {
  ❶ int x = 4;
  ❷ printf("%d", x++);
}
```

- ❶ 정수형 변수 x를 선언하고 4로 초기화한다.
- ❷ x의 값을 화면에 출력한다. k는 후치 증가 연산이므로 현재 값 4를 출력한 후에 1이 증가하여 5가 된다.

[문제 19]

OUTER JOIN

[해설]

```
❶ SELECT *
❷ FROM topic LEFT OUTER JOIN auth
❸ ON topic.auth_id = auth.id;
```

- ❶ 모든 속성을 표시한다.
- ❷ LEFT OUTER JOIN이므로, 좌측의 <topic> 테이블이 기준이 되어 <topic> 테이블에 있는 튜플은 모두 표시하고 우측의 <auth> 테이블에서는 관련이 있는 튜플만 표시한다.
- ❸ <topic> 테이블의 'auth_id'와 <auth> 테이블의 'id'를 기준으로 서로 JOIN한다.

[문제 20]

GUI(Graphic User Interface)