



시험에 나오는것만 공부한다!

시나공시리즈

기출문제

2021년 2회 정보처리기사 실기



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. 수험자 인적사항 및 답안작성(계산식 포함)은 흑색 기구만 사용하되, 동일한 한 가지 색의 필기구만 사용하여야 하며 흑색을 제외한 유색 필기구 또는 연필류를 사용하거나 2가지 이상의 색을 혼합 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 굿지 않은 답안은 정정하지 않은 것으로 간주합니다.
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 네트워크 및 인터넷과 관련된 다음 설명에 해당하는 용어를 쓰시오. (5점)

- 재난 및 군사 현장과 같이 별도의 고정된 유선망을 구축할 수 없는 장소에서 모바일 호스트(Mobile Host)만을 이용하여 구성된 네트워크이다.
- 망을 구성한 후 단기간 사용되는 경우나 유선망을 구성하기 어려운 경우에 적합하다.
- 멀티 홉 라우팅 기능을 지원한다.

답 :

문제 2 인터페이스에 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

- (①) : 사용자가 시스템이나 서비스를 이용하면서 느끼고 생각하게 되는 총체적인 감정 및 경험
- (②) : 사용자와 시스템 간의 상호작용이 원활하게 이뤄지도록 도와주는 장치나 소프트웨어
예) CLI, GUI 등

답

- ①
- ②

문제 3 데이터베이스의 상태 변화를 일으키는 트랜잭션(Transaction)의 특성 중 원자성(Atomicity)에 대해 간략히 서술하시오. (5점)

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 4 데이터베이스에 대한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 답을 쓰시오. (5점)

테이블을 만들 때는 이상(Anomaly)을 방지하기 위해 데이터들의 중복성 및 종속성을 배제하는 정규화를 수행한다. 아래 그림은 부분 함수적 종속을 제거하여 제 () 정규형을 만드는 과정이다.

<Table R>

A(key)	B(key)	C	D
A345	1001	Seoul	Pmre
D347	1001	Busan	Preo
A210	1007	Gwangju	Ciqen
A345	1007	Seoul	Esto
B230	1007	Daegu	Loid
D347	1201	Busan	Drag

<Table R의 함수적 종속 관계>

A, B → C, D
A → C



<Table R1>

A(key)	B(key)	D
A345	1001	Pmre
D347	1001	Preo
A210	1007	Ciqen
A345	1007	Esto
B230	1007	Loid
D347	1201	Drag

<Table R2>

A(key)	C
A345	Seoul
D347	Busan
A210	Gwangju
B230	Daegu

<Table R> 에서 함수적 종속관계를 보았을 때, C는 key에 해당하는 A와 B중 A에만 종속되는 부분 함수적 종속이므로 C를 테이블에서 분리하면 <Table R1>과 <Table R2>와 같이 제 () 정규형에 해당하는 테이블이 완성된다.

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 5 SQL과 관련한 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 답을 쓰시오. (5점)

UPDATE문은 테이블에 있는 튜플의 내용을 갱신할 때 사용하는 명령문으로, DML에 해당한다. 다른 DML로는 INSERT, DELETE가 있으며, 각각 새로운 튜플을 삽입하거나 삭제할 때 사용한다.

<학부생> 테이블

학부	학과번호	입학생수	담당관
정경대학	110	300	김해울
공과대학	310	250	이성관
인문대학	120	400	김해울
정경대학	120	300	김성수
인문대학	420	180	이을해

다음은 <학부생> 테이블에서 입학생수가 300이상인 튜플의 학과번호를 999로 갱신하는 SQL문이다.

(①) 학부생 (②) 학과번호 = 999 WHERE 입학생수 >= 300;

답

- ①
- ②

문제 6 다음은 <사원> 테이블과 <동아리> 테이블을 조인(Join)한 <결과>를 확인하여 <SQL문>의 괄호(①, ②)에 들어갈 알맞은 답을 쓰시오. (5점)

<사원>

코드	이름	부서
1601	김명해	인사
1602	이진성	경영지원
1731	박영광	개발
2001	이수진	

<동아리>

코드	동아리명
1601	테니스
1731	탁구
2001	볼링



<결과>

코드	이름	동아리명
1601	김명해	테니스
1602	이진성	
1731	박영광	탁구
2001	이수진	볼링

<SQL문>

SELECT a.코드, 이름, 동아리명 FROM 사원 a LEFT JOIN 동아리 b (①) a.코드 = b.(②);

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 7 다음 Python으로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
a = 100
result = 0
for i in range(1,3):
    result = a >> i
    result = result + 1
print(result)
```

답 :

문제 8 보안 및 암호화와 관련된 다음 설명에 해당하는 용어를 쓰시오. (5점)

- 2001년 미국 표준 기술 연구소(NIST)에서 발표한 대칭키 암호화 알고리즘이다.
- DES의 한계를 느낀 NIST에서 공모한 후 발표하였다.
- 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류된다.

답 :

문제 9 테스트에 대한 다음 설명에서 각 지문 ①~③에 해당하는 커버리지(Coverage)를 <보기>에서 찾아 기호로 쓰시오. (5점)

- ① 최소 한번은 모든 문장이 수행되도록 구성하는 검증 기준
- ② 결정(Decision) 검증 기준이라고도 하며, 조건식이 참(True)/거짓(False)일 때 수행되도록 구성하는 검증 기준
- ③ ②번과 달리 조건식에 상관없이 개별 조건이 참(True)/거짓(False)일 때 수행되도록 구성하는 검증 기준

<보기>

- | | | |
|----------------|-------------------|----------------|
| ㉠ 다중 조건 검증 기준 | ㉡ 선택 검증 기준 | ㉢ 조건 검증 기준 |
| ㉣ 결정(분기) 검증 기준 | ㉤ 결정(분기)/조건 검증 기준 | ㉥ 구문(문장) 검증 기준 |

답

- ①
- ②
- ③

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 10 다음은 <회원> 테이블에서 '이름'이 "이"로 시작하는 회원들을 '가입일' 순으로 내림차순 정렬하는 <SQL문>이다. 괄호(①, ②)에 들어갈 알맞은 답을 쓰시오. (5점)

<회원> 테이블

회원번호	이름	성별	가입일
1001	이진성	남	2021-06-23
1002	조이령	여	2021-06-24
1003	최민수	남	2021-06-28
1004	김차희	여	2021-07-03
1005	이미경	여	2021-07-10

<SQL문>

SELECT * FROM 회원 WHERE 이름 LIKE '()' ORDER BY 가입일 ();

답

- ①
- ②

문제 11 모듈에 대한 다음 설명에서 각 지문 ①~③에 해당하는 응집도(Cohesion)를 <보기>에서 찾아 기호로 쓰시오. (5점)

- ① 내부의 요소들이 기능적으로 연관성은 없으나, 순차적으로 실행될 때의 응집도
- ② 서로 다른 기능을 수행하지만 동일한 입력과 출력을 사용할 때의 응집도
- ③ 하나의 기능에 밀접하게 관련되어 있거나 연관되어 있을 때의 응집도

<보기>

- | | | | |
|-----------|-----------|-----------|-----------|
| ㉠ 기능적 응집도 | ㉡ 순차적 응집도 | ㉢ 교환적 응집도 | ㉣ 절차적 응집도 |
| ㉤ 시간적 응집도 | ㉥ 논리적 응집도 | ㉦ 우연적 응집도 | |

답

- ①
- ②
- ③

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 12 네트워크에 관련된 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

- (①) : 연결형 통신에서 주로 사용되는 방식으로, 출발지와 목적지의 전송 경로를 미리 연결하여 논리적으로 고정한 후 통신하는 방식
- (②) : 비연결형 통신에서 주로 사용되는 방식으로, 사전에 접속 절차를 수행하지 않고 헤더에 출발지에서 목적지까지의 경로 지정을 위한 충분한 정보를 붙여서 개별적으로 전달하는 방식

답

- ①
- ②

문제 13 디자인 패턴에 관련된 다음 설명에서 괄호에 들어갈 알맞은 답을 쓰시오. (5점)

디자인 패턴은 모듈 간의 관계 및 인터페이스를 설계할 때 참조할 수 있는 전형적인 해결 방식 또는 예제를 의미한다. 그 중 () 패턴은 클래스나 객체들이 서로 상호작용하는 방법이나 책임 분배 방법을 정의하는 패턴으로, Interpreter, Observer, Command 등이 그 예에 해당한다.

답 :

문제 14 데이터베이스의 병행제어(Concurrency Control) 기법 중 하나로, 접근한 데이터에 대한 연산을 모두 마칠때까지 추가적인 접근을 제한함으로써 상호 배타적으로 접근을 수행하는 기법을 쓰시오. (5점)

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 15 럼바우(Rumbaugh) 데이터 모델링에 대한 다음 설명에서 각 지문 ①~③에 해당하는 모델링을 <보기>에서 찾아 기호로 쓰시오. (5점)

- ① 다수의 프로세스들 간의 자료 흐름을 중심으로 처리 과정을 표현한 모델링
예) 자료흐름도(DFD)
- ② 시간의 흐름에 따른 객체들 간의 제어 흐름, 상호 작용, 동작 순서 등의 동적인 행위를 표현하는 모델링
예) 상태 변화도(STD), 사건 추적도
- ③ 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 표시하는 모델링
예) ER 다이어그램(ERD)

<보기>

- | | | | |
|-------------|------------|---------------|----------------|
| ㉠ Operation | ㉡ Sequence | ㉢ Information | ㉣ Transaction |
| ㉤ Function | ㉥ I/O | ㉦ Dynamic | ㉧ Cause-Effect |

답

- ①
- ②
- ③

문제 16 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
main() {
    int res = mp(2, 10);
    printf("%d", res);
}
int mp(int base, int exp) {
    int res = 1;
    for (int i = 0; i < exp; i++)
        res *= base;
    return res;
}
```

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 17 다음 Java로 구현된 프로그램을 분석하여 괄호에 들어갈 알맞은 예약어를 쓰시오. (5점)

```
public class Test {  
    public static void main(String[] args) {  
        System.out.print(Test.check(1));  
    }  
    (    ) String check(int num){  
        return (num >= 0) ? "positive" : "negative";  
    }  
}
```

답 :

문제 18 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>  
int main() {  
    int ary[3];  
    int s = 0;  
    *(ary + 0) = 1;  
    ary[1] = *(ary + 0) + 2;  
    ary[2] = *ary + 3;  
    for (int i = 0; i < 3; i++)  
        s = s + ary[i];  
    printf("%d", s);  
}
```

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 19 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수 하시오.) (5점)

```
public class ovr1 {
    public static void main(String[] args) {
        ovr1 a1 = new ovr1();
        ovr2 a2 = new ovr2();
        System.out.println(a1.sun(3,2) + a2.sun(3,2));
    }
    int sun(int x, int y) {
        return x + y;
    }
}
class ovr2 extends ovr1 {
    int sun(int x, int y) {
        return x - y + super.sun(x, y);
    }
}
```

답 :

문제 20 통합 테스트에 관련된 다음 설명에서 괄호에 들어갈 알맞은 답을 쓰시오. (5점)

통합 테스트는 단위 테스트가 끝난 모듈을 통합하는 과정에서 발생하는 오류 및 결함을 찾는 테스트 기법으로, 하위 모듈에서 상위 모듈 방향으로 통합하는 상향식 통합 테스트와 상위 모듈에서 하위 모듈 방향으로 통합하는 하향식 통합 테스트가 있다. 상향식 통합 테스트는 미완성이거나 문제가 있는 상위 모듈을 대체할 수 있는 테스트 드라이버가, 하향식 통합 테스트는 미완성이거나 문제가 있는 하위 모듈을 대체할 수 있는 테스트 ()이(가) 있어야 원활한 테스트가 가능하다.

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

기출문제 정답 및 해설

[문제 1]

※ 다음 중 하나를 쓰면 됩니다.

애드혹 네트워크, Ad-hoc Network

[문제 2]

※ 각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① UX, 사용자 경험, User Experience
- ② UI, 사용자 인터페이스, User Interface

[문제 3]

※ 다음 중 밑줄이 표시된 내용은 반드시 포함되어야 합니다.

트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다는 특성

[문제 4]

2

[문제 5]

- ① UPDATE ② SET

[풀이]

• SQL문

UPDATE 학부생	'학부생' 테이블을 갱신하라.
SET 학과번호 = 999	'학과번호'를 999로 갱신하라.
WHERE 입학생수 >= 300;	'입학생수'가 300이상인 튜플만을 대상으로 하라.

• SQL 실행 결과

학부	학과번호	입학생수	담당관
정경대학	999	300	김해율
공과대학	310	250	이성관
인문대학	999	400	김해율
정경대학	999	300	김성수
인문대학	420	180	이율해

[문제 6]

- ① ON ② 코드

[풀이]

SELECT a.코드, 이름, 동아리명	a가 가리키는 <사원> 테이블의 '코드'와 '이름', '동아리명'을 표시한다.
FROM 사원 a LEFT JOIN 동아리 b	• LEFT JOIN이므로, 좌측의 <사원> 테이블이 기준이 되어 <사원> 테이블에 있는 튜플은 모두 표시하고 우측의 <동아리> 테이블에서는 관련이 있는 튜플만 표시한다.
	• <사원>, <동아리> 테이블의 별칭으로 <a>, 를 지정한다. <a>는 <사원> 테이블을, 는 <동아리> 테이블을 가리키게 된다.
ON a.코드 = b.코드;	<사원> 테이블의 '코드'와 <동아리> 테이블의 '코드'를 기준으로 서로 JOIN한다.

[문제 7]

26

[해설]

result의 값이 for문을 반복 수행할 때마다 값이 누적되는 것이 아닌 새로운 값으로 치환된다는 것을 염두에 두고 해설을 참고하세요.

```

❶ a = 100
❷ result = 0
❸ for i in range(1,3):
❹     result = a >> i
❺     result = result + 1
❻ print(result)
    
```

- ❶ 변수 a에 100을 저장한다.
- ❷ 변수 result에 0을 저장한다.
- ❸ 반복 변수 i가 1에서 시작하여 1씩 증가하면서 3보다 작은 동안 ❹, ❺번을 반복 수행한다.
- ❹ >>는 오른쪽 시프트 연산자이므로, a에 저장된 값을 오른쪽으로 i비트 이동시킨 다음 그 값을 result에 저장한다. 정수는 4Byte이므로 100을 4Byte 2진수로 변환하여 계산하면 된다.
- ❺ result의 값에 1을 누적시킨다.

반복문 1회 수행(i = 1)

• result = a >> i

- a의 값 100을 4Byte 2진수로 표현하면 다음과 같다.

	32	31	30	29	...	9	8	7	6	5	4	3	2	1
100	0	0	0	0	...	0	0	1	1	0	0	1	0	0
	부호 비트						2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
							128	64	32	16	8	4	2	1

- 부호를 제외한 전체 비트를 i의 값 1만큼 오른쪽으로 1비트 이동시킨다. 양수이므로 패딩 비트(빈자리)에는 0이 채워진다.

	32	31	30	29	...	9	8	7	6	5	4	3	2	1
50	0	0	0	0	...	0	0	0	1	1	0	0	1	0
	부호 비트						2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	패딩 비트						128	64	32	16	8	4	2	1

- 이동된 값을 10진수로 변환하면 50이다. result에는 50이 저장된다.

• result = result + 1

- result의 값 50에 1을 누적시키면 result는 51이 된다.

반복문 2회 수행(i = 2)

• result = a >> i

- a의 값 100에서 부호를 제외한 전체 비트를 i의 값 2만큼 오른쪽으로 2비트 이동시킨다. 양수이므로 패딩 비트(빈자리)에는 0이 채워진다.

	32	31	30	29	...	9	8	7	6	5	4	3	2	1
50	0	0	0	0	...	0	0	0	0	1	1	0	0	1
	부호 비트						2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	패딩 비트						128	64	32	16	8	4	2	1

- 이동된 값을 10진수로 변환하면 25이다. result에는 25가 저장된다.

• result = result + 1

- result의 값 25에 1을 누적시키면 result는 26이 된다.

❻ result의 값 26을 출력한다.

결과 26

[문제 8]

※ 다음 중 하나를 쓰면 됩니다.

AES, Advanced Encryption Standard

[문제 9]

- ① ㉠ ② ㉡ ③ ㉢

[문제 10]

- ① 이% ② DESC

[풀이]

• SQL문

SELECT *	모든 속성을 표시한다.
FROM 회원	<회원> 테이블에서 검색한다.
WHERE 이름 LIKE '이%'	'이름'이 '이'로 시작하는 튜플만을 대상으로 한다.
ORDER BY 가입일 desc;	'가입일'을 기준으로 내림차순 정렬한다.

• SQL 실행 결과

회원번호	이름	성별	가입일
1005	이미경	여	2021-07-10
1001	이진성	남	2021-06-23

[문제 11]

- ① ㉠ ② ㉡ ③ ㉢

[문제 12]

※ 각 문항별로 다음 중 하나를 쓰면 됩니다.

- ① 가상 회선, 가상 회선 방식, VC, Virtual Circuit
② 데이터그램, 데이터그램 방식, Datagram

[문제 13]

※ 다음 중 하나를 쓰면 됩니다.

행위, Behavioral

[문제 14]

※ 다음 중 하나를 쓰면 됩니다.

로킹, Locking

[문제 15]

- ① ㉠ ② ㉡ ③ ㉢

[문제 16]

1024

[해설]

```
#include <stdio.h>
main() {
  ①⑦ int res = mp(2, 10);
  ⑧ printf("%d", res);
}
② int mp(int base, int exp) {
  ③ int res = 1;
  ④ for (int i = 0; i < exp; i++)
  ⑤     res *= base;
  ⑥ return res;
}
```

모든 C언어 프로그램은 반드시 main() 함수에서 시작한다

- ① 정수형 변수 res를 선언하고, 2와 10을 인수로 mp() 함수를 호출한 후 돌려받은 값을 res에 저장한다.
- ② 정수를 반환하는 mp() 함수의 시작점이다. ①번에서 전달받은 2와 10을 base와 exp가 받는다.
- ③ 정수형 변수 res를 선언하고 1로 초기화한다.
- ④ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 exp보다 작은 동안 ⑤번을 반복 수행한다.
- ⑤ 'res = res * base;'와 동일하다. res에 res와 base를 곱한 값을 저장한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

base	exp	i	res
2	10	0	1
		1	2
		2	4
		3	8
		4	16
		5	32
		6	64
		7	128
		8	256
		9	512
		10	1024

- ⑥ res의 값 1024를 mp()를 호출했던 ⑦번으로 반환한다.
- ⑦ ⑥번으로부터 반환받은 값 1024를 res에 저장한다.
- ⑧ res의 값을 출력한다.

결과 1024

[문제 17]

static

[답안 작성 시 주의 사항]

C, Java, Python 등의 프로그래밍 언어에서 사용하는 변수명과 예약어는 대소문자를 구분하기 때문에 작성할 때 대소문자를 구분해서 정확히 작성해야 합니다.

[해설]

static은 클래스 이름으로 메소드에 접근하기 위해 사용하는 예약어로, 메소드를 사용하기 위해서는 메소드가 포함된 클래스의 객체 변수를 선언한 후 **[객체 변수].[메소드]**의 방식으로 접근해야 하지만 **static**을 이용하면 객체 변수 없이 **[클래스 이름].[메소드]**의 방식으로 접근하는 것이 가능해집니다.

```
public class Test {
    public static void main(String[] args) {
        ①④ System.out.print(Test.check(1));
    }
    ② static String check(int num){
        ③ return (num >= 0) ? "positive" : "negative";
    }
}
```

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다

① 1을 인수로 Test 클래스의 check() 메소드를 호출한 후 돌려받은 값을 출력한다.

② 문자열을 반환하는 check() 메소드의 시작점이다. ①번에서 전달받은 1을 num이 받는다.

③ num이 0보다 크거나 같으면 “positive”를 check()를 호출했던 ④번으로 반환하고, 아니면 “negative”를 ④번으로 반환한다. num의 값 1은 0보다 크므로 “positive”를 ④번으로 반환한다.

④ ③번에서 돌려받은 “positive”를 출력한다.

결과 **positive**

[문제 18]

8

[해설]

```
#include <stdio.h>
int main() {
    ① int ary[3];
    ② int s = 0;
    ③ *(ary + 0) = 1;
    ④ ary[1] = *(ary + 0) + 2;
    ⑤ ary[2] = *ary + 3;
    ⑥ for (int i = 0; i < 3; i++)
    ⑦     s = s + ary[i];
    ⑧ printf("%d", s);
}
```

① 3개의 요소를 갖는 정수형 배열 ary를 선언한다.

메모리

0000 0000		
:		
ary		
0015 FC40 → 0015 FC40	ary[0]	ary[1] ary[2]
:		
	ary	ary+1 ary+2
FFFF FFFF	0015 FC40	0015 FC41 0015 FC42

② 정수형 변수 s를 선언하고 0으로 초기화한다.

③ ary+0이 가리키는 곳에 1을 저장한다.

메모리

0000 0000		
:		
ary		
0015 FC40 → 0015 FC40	1	
:		
	ary	ary+1 ary+2
FFFF FFFF	0015 FC40	0015 FC41 0015 FC42

④ ary[1]에 ary+0이 가리키는 곳의 값 1과 2를 더한 값을 저장한다.

0000 0000		
:		
ary		
0015 FC40 → 0015 FC40	1	3
:		
	ary	ary+1 ary+2
FFFF FFFF	0015 FC40	0015 FC41 0015 FC42

⑤ ary[2]에 ary가 가리키는 곳의 값 1과 3을 더한 값을 저장한다.

0000 0000		
:		
ary		
0015 FC40 → 0015 FC40	1	3 4
:		
	ary	ary+1 ary+2
FFFF FFFF	0015 FC40	0015 FC41 0015 FC42

⑥ 반복 변수 i가 0에서 시작하여 1씩 증가하면서 3보다 작은 동안 ⑦번을 반복 수행한다.

⑦ s에 ary[i]의 값을 누적한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	ary[i]	s
0	1	1
1	3	4
2	4	8
3		

⑧ s의 값을 출력한다.

결과 8

[문제 19]

11

[해설]

```
public class ovr1 {
    public static void main(String[] args) {
        ❶          ovr1 a1 = new ovr1();
        ❷          ovr2 a2 = new ovr2();
        ❸❹❺      System.out.println(a1.sun(3,2) + a2.sun(3,2));
    }
    ❹❻      int sun(int x, int y) {
        ❺❼      return x + y;
    }
}
class ovr2 extends ovr1 {
    ❷          int sun(int x, int y) {
        ❸❼      return x - y + super.sun(x, y);
    }
}
```

클래스 ovr2를 정의하고 부모 클래스로 ovr1을 지정하면서 ovr1에 속한 변수와 메소드를 상속받는다.

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다

- ❶ 클래스 ovr1의 객체변수 a1을 선언한다.
- ❷ 클래스 ovr2의 객체변수 a2를 선언한다.
- ❸ 3과 2를 인수로 a1의 sun() 메소드를 호출한 결과와, 3과 2를 인수로 a2의 sun() 메소드를 호출한 결과를 합하여 출력한 후 커서를 다음 줄로 옮긴다. 먼저 a1의 sun() 메소드를 호출한다.
- ❹ 정수를 반환하는 a1의 sun() 메소드의 시작점이다. ❸번에서 전달받은 3과 2를 x와 y가 받는다.
- ❺ x와 y를 더한 값 5(3+2)를 함수를 호출했던 ❸번으로 반환한다.
- ❻ ❺번으로부터 a1의 sun() 메소드를 호출한 결과로 5를 전달받았으므로, 이번에는 3과 2를 인수로 a2의 sun() 메소드를 호출한다.
- ❼ 정수를 반환하는 a2의 sun() 메소드의 시작점이다. ❹번에서 전달받은 3과 2를 x와 y가 받는다.
- ❸ x에서 y를 뺀 값에 3과 2를 인수로 부모 클래스인 ovr1의 sun() 메소드를 호출한 결과를 더하여 함수를 호출했던 ❺번으로 반환한다.
- ❹~❺ 3과 2를 인수로 ovr1의 sun() 메소드를 수행한 결과를 ❹~❺번에서 구했으므로 그 결과를 그대로 사용하면 된다. ❺번에서 5를 돌려받아 계산한 값 6(3-2+5)을 함수를 호출했던 ❺번으로 반환한다.
- ❺ ❺번으로부터 돌려받은 5와 ❺번으로부터 돌려받은 6을 더한 값 11(5+6)을 출력하고 커서를 다음 줄로 옮긴다.

결과 11

[문제 20]

※ 다음 중 하나를 쓰면 됩니다.

스텝, Stub