



시험에 나오는것만 공부한다!

시나공시리즈

기출문제

2022년 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 필기구만 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 데이터베이스에 대한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 쓰시오. (5점)

- ()은 관계 데이터의 연산을 표현하는 방법으로, 관계 데이터 모델의 제안자인 코드(E. F. Codd)가 수학의 술어 해석(Predicate Calculus)에 기반을 두고 관계 데이터베이스를 위해 제안했다.
- 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특성을 지니며, 원하는 정보를 정의할 때 계산 수식을 사용한다.
- 튜플 해석식을 사용하는 튜플 ()과 도메인 해석식을 사용하는 도메인 ()으로 구분된다.

답 :

문제 2 암호화 알고리즘에 대한 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 용어를 쓰시오. (5점)

- 암호화 알고리즘은 패스워드, 주민번호, 은행계좌와 같은 중요 정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법을 의미한다.
- 스위스의 라이(Lai)와 메시(Messey)는 1990년 PES를 발표하고, 이후 이를 개선한 IPES를 발표하였다. IPES는 128비트의 Key를 사용하여 64비트 블록을 암호화하는 알고리즘이며 현재는 (①)라고 불린다.
- (②)는 국가 안전 보장국(NSA)에서 개발한 암호화 알고리즘으로, 클리퍼 칩(Clipper Chip)이라는 IC 칩에 내장되어있다. 80비트의 Key를 사용하여 64비트 블록을 암호화하며, 주로 전화기와 같은 음성 통신 장비에 삽입되어 음성 데이터를 암호화한다.

답

- ①
- ②

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 3 다음은 <제품>(제품명, 단가, 제조사) 테이블을 대상으로 “H” 제조사에서 생산한 제품들의 ‘단가’ 보다 높은 ‘단가’를 가진 제품의 정보를 조회하는 <SQL문>이다. 괄호에 알맞은 답을 적어 <SQL문>을 완성하시오. (5점)

<SQL문>

```
SELECT 제품명, 단가, 제조사
FROM 제품
WHERE 단가 > (    ) (SELECT 단가 FROM 제품 WHERE 제조사 = 'H');
```

답 :

문제 4 다음 <TABLE>을 참조하여 <SQL문>을 실행했을 때 출력되는 결과를 쓰시오. (<TABLE>에 표시된 ‘NULL’은 값이 없음을 의미한다.) (5점)

<TABLE>

INDEX	COL1	COL2
1	2	NULL
2	4	6
3	3	5
4	6	3
5	NULL	3

<SQL문>

```
SELECT COUNT(COL2)
FROM TABLE
WHERE COL1 IN (2, 3)
      OR COL2 IN (3, 5);
```

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 5 네트워크 보안에 대한 다음 설명에서 괄호에 공통으로 들어갈 알맞은 용어를 영문 약어로 쓰시오. (5점)

- ()은 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션이다.
- 암호화된 규격을 통해 인터넷망을 전용선의 사설망을 구축한 것처럼 이용하므로 비용 부담을 줄일 수 있다.
- ()을 사용하면 두 장치 및 네트워크 사이에 암호화된 보안 터널이 생성되며, 터널에 사용되는 프로토콜에 따라 SSL ()과 IPsec ()으로 불린다.

답 :

문제 6 객체지향에 대한 다음 설명에 해당하는 용어를 <보기>에서 찾아 쓰시오. (5점)

- 자신이 사용하지 않는 인터페이스와 의존 관계를 맺거나 영향을 받지 않아야 한다는 객체지향 설계 원칙 중의 하나이다.
- 예를 들어 프린터, 팩스, 복사 기능을 가진 복합기의 경우 3가지 기능을 모두 가진 범용 인터페이스보다는 프린터 인터페이스, 팩스 인터페이스, 복사 인터페이스로 분리함으로써 하나의 기능 변경으로 인해 다른 기능이 영향을 받지 않도록 해야 한다.

<보기>

- | | | | |
|-------|---------|-------|-------|
| ㉠ SRP | ㉡ SOLID | ㉢ OCP | ㉣ LSP |
| ㉤ ISP | ㉥ DIP | ㉦ OTP | ㉧ PNP |

답 :

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

연 습 란

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

```
public class Test {
    public static void main(String[] args) {
        int i = 3, k = 1;
        switch(i) {
            case 1: k++;
            case 2: k -= 3;
            case 3: k = 0;
            case 4: k += 3;
            case 5: k -= 10;
            default: k--;
        }
        System.out.print(k);
    }
}
```

답 :

문제 8 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수 하시오.) (5점)

```
#include <stdio.h>
struct A {
    int n;
    int g;
}
main( ) {
    struct A st[2];
    for (int i = 0; i < 2; i++) {
        st[i].n = i;
        st[i].g = i + 1;
    }
    printf("%d", st[0].n + st[1].g);
}
```

답 :

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

연 습 란

문제 9 서브네팅(Subnetting)에 대한 다음 설명에서 괄호에 들어갈 알맞은 답을 쓰시오. (5점)

- 현재 IP 주소가 192.168.1.132이고, 서브넷 마스크가 255.255.255.192일 때,
- 네트워크 주소는 192.168.1.(①)이고, 해당 네트워크에서 네트워크 주소와 브로드캐스트 주소를 제외한 사용 가능 호스트의 수는 (②)개 이다.

답

- ①
- ②

문제 10 애플리케이션 테스트에 대한 다음 설명에서 괄호(①, ②)에 들어갈 알맞은 테스트를 쓰시오. (5점)

- 인수 테스트는 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 방법이다.
- (①) : 선정된 최종 사용자가 여러 명의 사용자 앞에서 행하는 테스트 기법으로, 실제 업무를 가지고 사용자가 직접 테스트한다.
 - (②) : 개발자의 장소에서 사용자가 개발자 앞에서 행하는 테스트 기법으로, 테스트는 통제된 환경에서 행해지며, 오류와 사용상의 문제점을 사용자와 개발자가 함께 확인하면서 기록한다.

답

- ①
- ②

문제 11 다음 설명에 해당하는 테스트를 <보기>에서 찾아 쓰시오. (5점)

- 통합 테스트로 인해 변경된 모듈이나 컴포넌트에 새로운 오류가 있는지 확인하는 테스트이다.
- 이미 테스트된 프로그램의 테스트를 반복하는 것이다.
- 수정한 모듈이나 컴포넌트가 다른 부분에 영향을 미치는지, 오류가 생기지 않았는지 테스트하여 새로운 오류가 발생하지 않음을 보증하기 위해 반복 테스트한다.

<보기>

- | | | | |
|---------------|--------------|-------------|--------------|
| ㉠ Integration | ㉡ Big Bang | ㉢ System | ㉣ Acceptance |
| ㉤ Unit | ㉥ Regression | ㉦ White Box | ㉧ Black Box |

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 12 다음은 <EMPLOYEE> 릴레이션에 대해 <관계 대수식>을 수행했을 때 출력되는 <결과>이다. <결과>의 각 괄호(①~⑤)에 들어갈 알맞은 답을 쓰시오. (5점)

<관계 대수식>

$\pi_{TTL}(EMPLOYEE)$

<EMPLOYEE>

INDEX	AGE	TTL
1	48	부장
2	25	대리
3	41	과장
4	36	차장



<결과>

①
②
③
④
⑤

답

- ①
- ②
- ③
- ④
- ⑤

문제 13 다음 Python으로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
a = "REMEMBER NOVEMBER"
b = a[0:3] + a[12:16]
c = "R AND %s" % "STR"
print(b + c)
```

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하기 바랍니다.

문제 14 경로 제어 프로토콜(Routing Protocol)에 대한 다음 설명에서 괄호(①~④)에 들어갈 알맞은 용어를 <보기>에서 찾아 쓰시오. (5점)

경로 제어 프로토콜은 크게 자율 시스템 내부의 라우팅에 사용되는 (①)와 자율 시스템 간의 라우팅에 사용되는 (②)로 구분할 수 있다.
(①)는 소규모 동종 자율 시스템에서 효율적인 RIP와 대규모 자율 시스템에서 많이 사용되는 (③)로 나뉘어진다. (③)는 링크 상태(Link State)를 실시간으로 반영하여 최단 경로로 라우팅을 지원하는 특징이 있다.
(④)는 (②)의 단점을 보완하여 만들어진 라우팅 프로토콜로, 처음 연결될 때는 전체 라우팅 테이블을 교환하고, 이후에는 변화된 정보만을 교환한다.

<보기>

㉠ BGP	㉡ AS	㉢ HOP	㉣ OSPF
㉤ NAT	㉥ ISP	㉦ EGP	㉧ IGP

답

- ①
- ②
- ③
- ④

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

연 습 란

문제 15 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.) (5점)

```
#include <stdio.h>
int len(char* p);

int main( ) {
    char* p1 = "2022";
    char* p2 = "202207";
    int a = len(p1);
    int b = len(p2);
    printf("%d", a + b);
}

int len(char* p) {
    int r = 0;
    while (*p != '\0') {
        p++;
        r++;
    }
    return r;
}
```

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

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

```
#include <stdio.h>
int main( ) {
    int a[4] = { 0, 2, 4, 8 };
    int b[3];
    int* p;
    int sum = 0;
    for (int i = 1; i < 4; i++) {
        p = a + i;
        b[i - 1] = *p - a[i - 1];
        sum = sum + b[i - 1] + a[i];
    }
    printf("%d", sum);
}
```

답 :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 17 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수 하시오.) (5점)

```
class Test {
    public static void main(String args[]) {
        cond obj = new cond(3);
        obj.a = 5;
        int b = obj.func( );
        System.out.print(obj.a + b);
    }
}

class cond {
    int a;
    public cond(int a) {
        this.a = a;
    }
    public int func( ) {
        int b = 1;
        for (int i = 1; i < a; i++)
            b += a * i;
        return a + b;
    }
}
```

답 :

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

연 습 란

문제 18 함수적 종속(Functional Dependency)에 대한 다음 설명에서 괄호(①~③)에 들어갈 알맞은 용어를 <보기>에서 찾아 쓰시오. (단, 테이블 <R>의 속성 '학생'과 '학과'의 밑줄은 키(Key)임을 의미한다.) (5점)

<R>

학생	학과	성적	학년
이순신	컴퓨터공학	A+	2
이순신	전기공학	B	2
유관순	경제학	B+	1
강감찬	문예창작	C	3
강감찬	한국사	C+	3
홍길동	영문학	B	4

- 테이블 <R>에서 '성적'은 기본키인 {학생, 학과}에 대해 (①) Functional Dependency이다.
- 테이블 <R>에서 '학년'은 기본키인 {학생, 학과} 중 '학생'만으로 식별이 가능하므로 기본키에 대해 (②) Functional Dependency이다.
- 임의의 테이블에 속성 A, B, C가 있을 때, $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때 $A \rightarrow C$ 인 관계는 (③) Functional Dependency이다.

<보기>

- | | | | |
|-----------|----------------|--------------|-----------|
| ㉠ Hybrid | ㉡ Multi Valued | ㉢ Transitive | ㉤ Full |
| ㉣ Defined | ㉥ Natural | ㉦ Relational | ㉧ Partial |

답

- ①
- ②
- ③

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

문제 19 인터넷에 대한 다음 설명에서 괄호(①~③)에 들어갈 알맞은 용어를 <보기>에서 찾아 쓰시오. (5점)

인터넷이란 TCP/IP 프로토콜을 기반으로 하여 전 세계 수많은 컴퓨터와 네트워크들이 연결된 광범위한 컴퓨터 통신망이다.

- (①) : 월드 와이드 웹(WWW)에서 HTML 문서를 송수신 하기 위한 표준 프로토콜로, GET과 POST 메소드를 통해 메시지를 주고 받는다.
- (②) : 다른 문서나 그림으로 이동할 수 있는 연결을 가지고 있는 텍스트를 의미한다.
- (③) : 인터넷의 표준 문서인 하이퍼텍스트 문서를 만들기 위해 사용하는 언어이다.

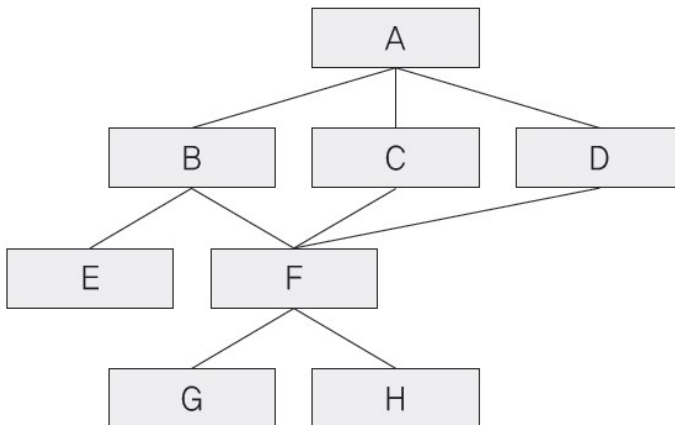
<보기>

- | | | | |
|---------|--------|--------|-------------|
| ㉠ UDDI | ㉡ XML | ㉢ WSDL | ㉣ Hypertext |
| ㉤ DHTML | ㉥ HTML | ㉦ SOAP | ㉧ HTTP |

답

- ①
- ②
- ③

문제 20 다음의 모듈 관계를 표현한 시스템 구조도를 참고하여 모듈 F의 팬인(Fan-In)과 팬아웃(Fan-Out)을 구하시오. (5점)



답

- ① 팬인(Fan-In) :
- ② 팬아웃(Fan-Out) :

연 습 란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

기출문제 정답 및 해설

[문제 1]

※ 다음 중 하나를 쓰면 됩니다.

관계해석, Relational Calculus

[문제 2]

※ 다음 중 하나를 쓰면 됩니다.

① IDEA, International Data Encryption Algorithm

② Skipjack

[문제 3]

ALL

[답안 작성 시 주의 사항]

SQL에 사용되는 예약어, 필드명, 변수명 등은 대소문자를 구분하지 않기 때문에 대문자로만 또는 소문자로만 작성해도 정답으로 인정됩니다.

[해설]

```

② SELECT 제품명, 단가, 제조사  '제품명', '단가', '제조사' 속성을 표시한다.
    FROM 제품                  <제품> 테이블의 자료를 검색한다.
    WHERE 단가 > ALL (         '단가'가 하위 질의로 검색된 모든(ALL) 단가보다 큰 자료만을 대상으로 한다.
        ① SELECT 단가          '단가' 속성을 표시한다.
            FROM 제품          <제품> 테이블의 자료를 검색한다.
            WHERE 제조사 = 'H'); 제조사가 "H"인 튜플만을 대상으로 한다.
    
```

문제의 질의문은 하위 질의가 있는 질의문입니다. 먼저 WHERE 조건에 지정된 하위 질의의 SELECT문을 해석한 다음 그 결과를 본 질의의 조건에 있는 '단가' 속성과 비교합니다. <제품> 테이블에 다음과 같은 자료가 들어있다고 가정하여 설명합니다.

<제품>

제품명	단가	제조사
냉장고	200	H
TV	150	H
세탁기	300	H
건조기	250	A
핸드폰	400	B
컴퓨터	500	C

① <제품> 테이블에서 '제조사' 속성의 값이 "H"인 튜플의 '단가' 속성의 값을 검색합니다.

단가
200
150
300

② <제품> 테이블에서 '단가' 속성의 값이 ①번에서 검색된 모든 단가보다 큰 자료를 대상으로 '제품명', '단가', '제조사' 속성을 표시합니다.

제품명	단가	제조사
핸드폰	400	B
컴퓨터	500	C

[문제 4]

3

[해설]

SELECT COUNT(COL2)	'COL2'의 개수를 표시한다.
FROM TABLE	<TABLE>을 대상으로 검색한다.
WHERE COL1 IN (2, 3)	'COL1'이 2 또는 3이거나,
OR COL2 IN (3, 5);	'COL2'가 3 또는 5인 튜플만을 대상으로 한다.

- 질의문의 조건을 만족하는 튜플은 다음과 같습니다.

INDEX	COL1	COL2
1	2	NULL
2	4	6
3	3	5
4	6	3
5	NULL	3

- 조건에 맞는 'COL2' 속성만 추출하면 다음과 같습니다.

COL2
NULL
5
3
3

- COUNT(COL2)는 'COL2' 필드의 개수를 계산하지만 'NULL' 값은 제외하므로 COUNT(COL2)의 결과는 3입니다.

[문제 5]

VPN

[문제 6]

ISP

[문제 7]

-8

[해설]

```

public class Test {
    public static void main(String[] args) {
        ❶ int i = 3, k = 1;
        ❷ switch(i) {
            case 1: k++;
            case 2: k -= 3;
            ❸ case 3: k = 0;
            ❹ case 4: k += 3;
            ❺ case 5: k -= 10;
            ❻ default: k--;
        }
        ❼ System.out.print(k);
    }
}

```

- ❶ 정수형 변수 i, k를 선언하고, 각각 3과 1로 초기화한다.
- ❷ i의 값 3에 해당하는 숫자를 찾아간다. 'case 3' 문장으로 이동한다.
- ❸ k에 0을 저장한다. → k = 0

※ switch 문을 종료하는 break가 없으므로 ④, ⑤, ⑥번을 모두 수행하고 ⑦번으로 이동한다.

- ④ 'k = k + 3;'과 동일하다. k의 값에 3을 더한다. → k = 3
- ⑤ 'k = k - 10;'과 동일하다. k의 값에서 10을 뺀다. → k = -7
- ⑥ 'k = k - 1;'과 동일하다. k의 값에서 1을 뺀다. → k = -8
- ⑦ k의 값을 출력한다.

결과 **-8**

[문제 8]

2

[해설]

```
#include <stdio.h>
struct A {                구조체 A를 정의한다.
    int n;                A의 멤버로 정수형 변수 n을 선언한다.
    int g;                A의 멤버로 정수형 변수 g를 선언한다.
}
main( ) {
    ❶ struct A st[2];
    ❷ for (int i = 0; i < 2; i++) {
    ❸     st[i].n = i;
    ❹     st[i].g = i + 1;
    }
    ❺ printf("%d", st[0].n + st[1].g);
}
```

모든 C 프로그램은 반드시 main() 함수에서 시작한다.

- ❶ A 구조체 형태로 배열 st를 선언한다.

	int n	int g
st[0]		
st[1]		

- ❷ 반복 변수 i가 0부터 1씩 증가하면서 2보다 작은 동안 ❸, ❹번을 반복 수행한다.
- ❸ st[i].n에 i의 값을 저장한다.
- ❹ st[i].g에 i+1의 값을 저장한다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

- 1회전 (i = 0)

	int n	int g
st[0]	0	1
st[1]		

- 2회전 (i = 1)

	int n	int g
st[0]	0	1
st[1]	1	2

- i가 2가 되면서 for문을 빠져나가 ❺번으로 이동한다.

- ❺ 0+2의 결과인 2를 정수로 출력한다.

결과 **2**

[문제 9]

- ① 128 ② 62

[해설]

- IP 주소는 네트워크 부분의 길이에 따라 다음과 같이 A 클래스에서 E 클래스까지 총 5단계로 구성되어 있으며, 각 클래스는 IP 주소의 앞자리로 구분할 수 있습니다.

A Class	0~127로 시작
B Class	128~191로 시작
C Class	192~223으로 시작
D Class	224~239로 시작
E Class	공용되지 않음

- 192.168.1.132는 C 클래스에 속한 주소로, C 클래스의 기본 서브넷 마스크는 255.255.255.0입니다. 이를 2진수로 표현하면 11111111 11111111 11111111 00000000으로, 1의 개수가 24개입니다.
- 문제에 주어진 서브넷 마스크 255.255.255.192를 2진수로 표현하면 11111111 11111111 11111111 11000000으로, C 클래스의 기본 서브넷 마스크보다 1의 개수가 2개, 즉 2비트가 많습니다. 이 2비트를 이용해 네트워크의 개수와 네트워크 안에 포함된 호스트의 개수를 계산합니다.
- 네트워크 개수 = $4(2^{\text{추가된 비트 수}})$
- 호스트 개수 = $256/\text{네트워크 개수} = 256/4 = 64$
- 다음 표와 같이 64개의 호스트를 갖는 4개의 네트워크로 나눌 수 있습니다.
- 네트워크별로 첫 번째 주소는 네트워크 주소이고, 마지막 주소는 브로드캐스트 주소입니다.

네트워크	네트워크 주소	브로드캐스트 주소
1	192.168.1.0	192.168.1.63
2	192.168.1.64	192.168.1.127
3	192.168.1.128	192.168.1.191
4	192.168.1.192	192.168.1.255

- 192.168.1.132는 세 번째 네트워크에 포함되어 있으며, 세 번째 네트워크의 네트워크 주소는 192.168.1.128입니다.
- 호스트의 수는 네트워크마다 64개의 호스트를 가지므로 64개이지만, 문제에서 네트워크 주소와 브로드캐스트 주소를 제외한다고 하였으므로 사용 가능 호스트의 수는 62개입니다.

[문제 10]

- ① 베타 테스트(Beta Test) ② 알파 테스트(Alpha Test)

[문제 11]

Regression

[문제 12]

- ① TTL ② 부장 ③ 대리 ④ 과장 ⑤ 차장

[해설]

문제의 <관계 대수식>에서 사용된 π 는 주어진 릴레이션에서 속성 리스트(Attribute List)에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 PROJECT 연산이므로, <EMPLOYEE> 릴레이션에서 'TTL' 속성이 추출되어 속성명인 'TTL'부터 모든 속성값이 <결과>로 나타납니다.

[문제 13]

REMEMBER AND STR

[답안 작성 시 주의 사항]

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어, 출력값 사이에 공백 없이 REMEMBERANDSTR로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 처리됩니다.

[해설]

```

❶ a = "REMEMBER NOVEMBER"
❷ b = a[0:3] + a[12:16]
❸ c = "R AND %s" % "STR"
❹ print(b + c)

```

❶ 변수 a를 선언하고 “REMEMBER NOVEMBER”로 초기화한다.

❷ a에 저장된 문자열의 0부터 2번째 위치까지의 문자열과 12부터 15번째 위치까지의 문자열을 합쳐 b에 저장한다.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]
a	R	E	M	E	M	B	E	R		N	O	V	E	M	B	E	R

b = REMEMBE

❸ c에 “R AND STR”을 저장한다. %s는 서식 문자열로, % 뒤쪽의 “STR”이 대응된다.

• “R AND %s” % “STR”

↑ ↑
 (아래의 문자열이 %s와 STR에 각각 대입됨)

❹ b와 c에 저장된 문자열을 합쳐 출력한다.

결과 **REMEMBER AND STR**

[문제 14]

- ❶ IGP ❷ EGP ❸ OSPF ❹ BGP

[문제 15]

10

[해설]

```

#include <stdio.h>
int len(char* p);            len( ) 함수의 프로토타입 선언이다.

main( ) {
  ❶ char* p1 = "2022";
  ❷ char* p2 = "202207";
  ❸❶ int a = len(p1);
  ❹❷ int b = len(p2);
  ❺ printf("%d", a + b);
}

4❷ int len(char* p) {
  5❸ int r = 0;
  6❹ while (*p != '\0') {
  7❺ p++;
  8❻ r++;
  }
  9❻ return r;
}

```

모든 C 프로그램은 반드시 main() 함수에서 시작한다.

❶ 문자형 포인터 변수 p1을 선언하고 “2022”를 가리키는 주소로 초기화한다.

❷ 문자형 포인터 변수 p2를 선언하고 “202207”을 가리키는 주소로 초기화한다. 다음의 그림에서 p1과 p2가 할당된 공간의 주소는 임의로 정한 것이며, 이해를 돕기 위해 10진수로 표현했다.

주소		메모리						
p1	1000	'2'	'0'	'2'	'2'	\0		
p2	2000	'2'	'0'	'2'	'2'	'0'	'7'	\0

③ 정수형 변수 a를 선언하고 p1의 값을 인수로 len() 함수를 호출한 후 돌려받은 값으로 초기화한다.

④ 정수를 반환하는 len() 함수의 시작점이다. ③번에서 전달받은 p1의 값을 문자형 포인터 변수 p가 받는다.

주소		메모리						
p	1000	1Byte	1Byte	1Byte	1Byte	1Byte		
		'2'	'0'	'2'	'2'	\0		
		p+0	p+1	p+2	p+3	p+4		
		1000	1001	1002	1003	1004		

⑤ 정수형 변수 r을 선언하고 0으로 초기화한다.

⑥ p가 가리키는 곳의 값이 '\0'이 아닌 동안 ⑦, ⑧번을 반복 수행한다.

⑦ 'p = p + 1;'과 동일하다. p의 값을 1씩 누적시킨다.

⑧ 'r = r + 1;'과 동일하다. r의 값을 1씩 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

p	r	*p
1000	0	'2'
1001	1	'0'
1002	2	'2'
1003	3	'2'
1004	4	\0

⑨ 함수를 호출했던 ⑩번으로 r의 값 4를 반환한다.

⑩ a에는 4가 저장된다.

⑪ 정수형 변수 b를 선언하고 p2의 값을 인수로 len() 함수를 호출한 후 돌려받은 값으로 초기화한다.

⑫ 정수를 반환하는 len() 함수의 시작점이다. ⑪번에서 전달받은 p2의 값을 p가 받는다.

주소		메모리						
p	2000	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte
		'2'	'0'	'2'	'2'	'0'	'7'	\0
		p+0	p+1	p+2	p+3	p+4	p+5	p+6
		2000	2001	2002	2003	2004	2005	2006

⑬ 정수형 변수 r을 선언하고 0으로 초기화한다.

⑭ p가 가리키는 곳의 값이 '\0'이 아닌 동안 ⑮, ⑯번을 반복 수행한다.

⑮ 'p = p + 1;'과 동일하다. p의 값을 1씩 누적시킨다.

⑯ 'r = r + 1;'과 동일하다. r의 값을 1씩 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

p	r	*p
2000	0	'2'
2001	1	'0'
2002	2	'2'
2003	3	'2'
2004	4	'0'
2005	5	'7'
2006	6	\0

⑰ 함수를 호출했던 ⑱번으로 r의 값 6을 반환한다.

⑱ b에는 6이 저장된다.

19 4+6의 결과인 10을 정수로 출력한다.

결과 10

[문제 16]

22

[해설]

```
#include <stdio.h>
int main() {
  ❶ int a[4] = { 0, 2, 4, 8 };
  ❷ int b[3];
  ❸ int* p;
  ❹ int sum = 0;
  ❺ for (int i = 1; i < 4; i++) {
    ❻ p = a + i;
    ❼ b[i - 1] = *p - a[i - 1];
    ❽ sum = sum + b[i - 1] + a[i];
  }
  ❾ printf("%d", sum);
}
```

❶ 4개의 요소를 갖는 정수형 배열 a를 선언하고 초기화한다.

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

❷ 3개의 요소를 갖는 정수형 배열 b를 선언한다.

[0] [1] [2]
b [] [] []

❸ 정수형 포인터 변수 p를 선언한다.

❹ 정수형 변수 sum을 선언하고 0으로 초기화한다.

❺ 반복 변수 i가 1부터 1씩 증가하면서 4보다 작은 동안 ❻~❽번을 반복 수행한다.

❻ p에 a+i의 주소를 저장한다.

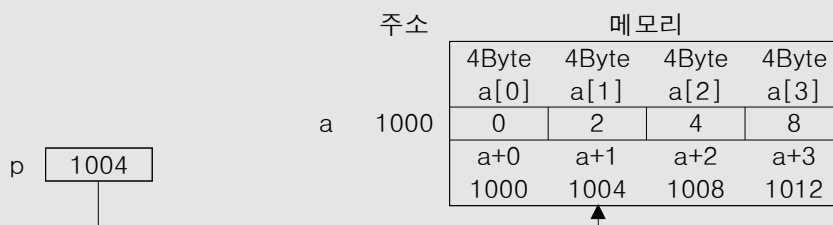
❼ b[i-1]에 p가 가리키는 곳의 값에서 a[i-1]을 뺀 값을 저장한다.

❽ sum에 b[i-1] + a[i]의 값을 누적시킨다.

반복문 실행에 따른 변수들의 변화는 다음과 같다.

• 1회전 (i = 1)

- p에 a+1의 주소인 1004를 저장한다.



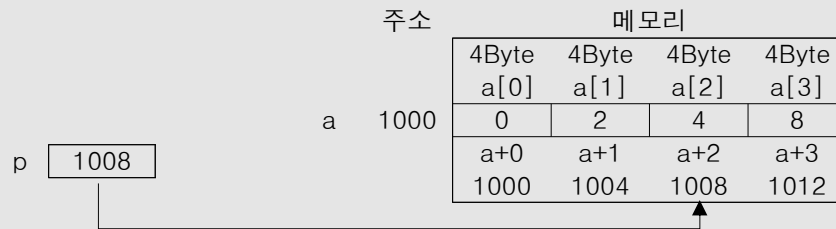
- b[0]에 p가 가리키는 곳의 값 2에서 a[0]의 값 0을 뺀 값인 2를 저장한다.

- sum에 b[0]의 값 2와 a[1]의 값 2를 더한 값 4를 누적한다.

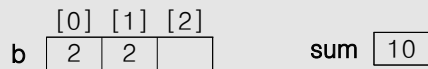
[0] [1] [2]
b [2] [] [] sum [4]

• 2회전 (i = 2)

- p에 a+2의 주소인 1008을 저장한다.

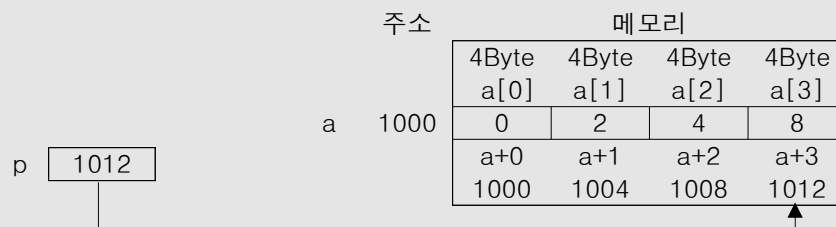


- b[1]에 p가 가리키는 곳의 값 4에서 a[1]의 값 2를 뺀 값인 2를 저장한다.
- sum에 b[1]의 값 2와 a[2]의 값 4를 더한 값 6을 누적한다.

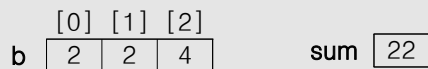


• 3회전 (i = 3)

- p에 a+3의 주소인 1012를 저장한다.



- b[2]에 p가 가리키는 곳의 값 8에서 a[2]의 값 4를 뺀 값인 4를 저장한다.
- sum에 b[2]의 값 4와 a[3]의 값 8을 더한 값 12를 누적한다.



- i가 4가 되면서 for문을 빠져나가 ⑨번으로 이동한다.

⑨ sum의 값 22를 정수로 출력한다.

결과 22

[문제 17]

61

[해설]

```

class Test {
    public static void main(String args[]) {
        ❶ cond obj = new cond(3);
        ❷ obj.a = 5;
        ❸❹ int b = obj.func( );
        ❺ System.out.print(obj.a + b);
    }
}

class cond {
    int a;
    ❶ public cond(int a) {
        ❷ this.a = a;
    }
    ❸ public int func( ) {
        ❹ int b = 1;
        ❺ for (int i = 1; i < a; i++)
        ❻ b += a * i;
        ❼ return a + b;
    }
}

```

모든 Java 프로그램은 반드시 main() 메소드에서 시작한다.

- ❶ 3을 인수로 생성자를 호출하여 cond 클래스의 객체 변수 obj를 선언한다.
- ❷ cond 클래스 생성자의 시작점이다. ❶번에서 전달받은 3을 a가 받는다.
- ❸ cond 클래스의 a에 cond() a의 값 3을 저장한다. 생성자가 종료되면 호출했던 ❶번의 다음 줄인 ❷번으로 이동한다. → obj.a = 3
 - this : 현재의 실행중인 메소드가 속한 클래스를 가리키는 예약어이다. 여기에서는 cond 클래스의 객체 변수 obj의 생성자로 호출되었으므로 'obj.a'와 같은 의미이다.
- ❹ obj.a에 5를 저장한다. → obj.a = 5
- ❺ 정수형 변수 b를 선언하고 obj.func() 메소드를 호출한 후 돌려받은 값으로 초기화한다.
- ❻ 정수를 반환하는 func() 메소드의 시작점이다.
- ❼ 정수형 변수 b를 선언하고 1로 초기화한다.
- ❽ 반복 변수 i가 1부터 1씩 증가하면서 a보다 작은 동안 ❹번을 반복 수행한다. func() 메소드에는 별도로 생성한 'a'라는 변수가 없으므로 cond 클래스의 a를 가져와 사용한다. 즉 ❹번은 5보다 작은 동안 반복 수행된다.
- ❹ 'b = b + (a * i);'와 동일하다. a에 i를 곱한 값을 b에 누적시킨다.
반복문 실행에 따른 변수들의 변화는 다음과 같다.

a	i	b
5		1
	1	6
	2	16
	3	31
	4	51
	5	

- ❺ a의 값 5와 b의 값 51을 더한 값 56을 메소드를 호출했던 ❸번으로 반환한다.
- ❻ b에는 56이 저장된다.
- ❼ 5+56의 결과인 61을 출력한다.

결과 61

[문제 18]

- ① Full ② Partial ③ Transitive

[문제 19]

- ① HTTP ② Hypertext ③ HTML

[문제 20]

- ① 3 ② 2