

흥달샘과 함께하는

2024년 1회 정보처리기사 실기 기출해설 특강 학습자료

이 자료는 대한민국 저작권법의 보호를 받습니다.

작성된 모든 내용의 권리는 작성자에게 있으며, 작성자의 동의 없는 사용이 금지됩니다.

본 자료의 일부 혹은 전체 내용을 무단으로 복제/배포하거나 2차적 저작물로 재편집하는 경우,
5년 이하의 징역 또는 5천만 원 이하의 벌금과 민사상 손해배상을 청구합니다.

YouTube 흥달샘 (<https://bit.ly/3KtwdLG>)

E-Mail hungjik@naver.com

01. 아래 응집도와 관련하여 응집도가 높은 순에서 낮은 순으로 답안을 작성하시오.

① 기능	② 교환	③ 우연	④ 시간
------	------	------	------

02. 다음에서 설명하는 네트워크 용어를 영문 약어로 작성하시오.

다음은 인터넷 프로토콜 스위트의 데이터 교환을 위해 네트워크 장치 간의 정보를 배포하는 데 사용되는 링크 상태 라우팅 프로토콜이다. 이 프로토콜은 최단 경로 우선 원칙을 사용하여 라우팅 결정을 내리며, 라우터가 동일한 계층의 다른 라우터와 직접적으로 정보를 교환할 수 있도록 한다. 특히 대규모 네트워크에서 효율적이며, 라우팅 테이블을 최신 상태로 유지하기 위해 자주 업데이트된다. 라우터는 이 프로토콜을 사용하여 네트워크의 변경 사항에 신속하게 반응하고, 네트워크의 각 세그먼트에서 정보를 동기화하는데 필요한 데이터를 교환한다.

03. 다음은 애플리케이션 테스트에 대한 내용이다. 해당하는 답을 보기에서 골라 작성하시오.

다음은 소프트웨어 테스팅에서 사용되는 테스트 커버리지 기준 중 하나이다. 이 기준은 각 결정점에서 모든 가능한 결정 결과를 최소 한 번 이상 실행하도록 요구하며, 모든 조건식이 참과 거짓이 되도록 만들어진다. 이를 통해 개발자와 테스터는 소프트웨어 내의 결정점에서 발생할 수 있는 다양한 조건 변경 사항들이 올바르게 처리되고 있는지 검증할 수 있다. 이러한 테스트 커버리지는 조건의 모든 논리적 경로를 테스트하여 소프트웨어의 품질을 향상시키는 데 중요한 역할을 한다.

04. 아래 내용을 보고 보기에서 알맞은 용어를 골라 작성하시오.

다음은 컴퓨터 시스템 내에서 사용자의 허가 없이 자신의 존재를 숨기고, 관리자 레벨의 접근 권한을 얻기 위해 설계된 악성 소프트웨어의 한 유형이다. 이 소프트웨어는 시스템의 기본적인 기능에 깊숙이 침투하여 활동을 감추고, 다양한 보안 도구로부터 탐지를 피할 수 있다. 종종 시스템 로그 파일을 수정하거나, 키스트로크를 기록하고, 비밀번호를 도용하는 등의 활동을 포함할 수 있다. 이러한 특성 때문에 보안 전문가들에게는 큰 위협이며, 제거가 매우 어려울 수 있다.

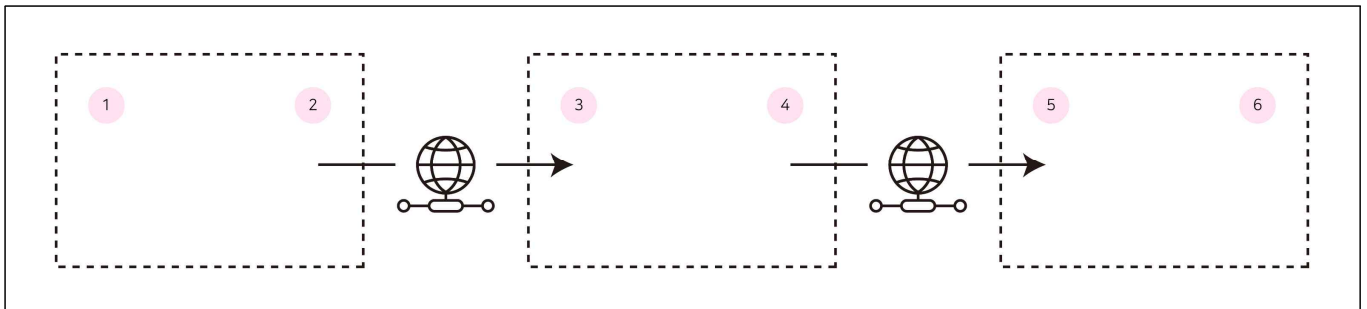
05. 아래 내용을 보고 보기에서 알맞은 용어를 골라 작성하시오.

다음은 고도로 조직화된 공격자 그룹이 특정 대상에 대해 장기간에 걸쳐 은밀하게 수행하는 사이버 공격 방법이다. 이러한 공격은 대개 국가 지원을 받는 해커들에 의해 이루어지며, 특정 기업, 정부 기관, 또는 중요 인프라를 목표로 삼는다. 공격자들은 네트워크 내에 오랫동안 숨어있으면서 데이터를 탈취하거나 시스템을 조작하는 등의 활동을 통해 심각한 피해를 입힐 수 있다. 이러한 공격은 그 목적과 수행 방식에 있어 일반적인 사이버 위협과 구별되며, 매우 정교하고 지속적인 모니터링을 요구한다.

06. 아래 내용을 보고 보기에서 알맞은 용어를 작성하시오.

다음은 소프트웨어 공학에서 사용되는 생성 디자인 패턴 중 하나로, 관련 또는 의존하는 객체의 집합을 생성하기 위한 인터페이스를 제공한다. 이 패턴은 구체적인 클래스를 지정하지 않고도 객체의 생성을 캡슐화하는 방법을 제공하여, 클라이언트 코드가 특정 클래스의 인스턴스에 직접 의존하지 않도록 한다. 이를 통해 클라이언트는 여러 제품군 중 필요한 제품군을 선택하여 생성할 수 있으며, 시스템의 확장성과 유연성이 증가된다. 다양한 환경에서 공통적으로 사용될 수 있는 제품군을 생성하는 과정에서 특히 유용하다.

07. 아래 네트워크 구성에서, 2, 4, 6번의 라우터에 할당 가능한 IP를 보기에서 찾아 쓰시오.



[적용된 IP]

- ① 192.168.35.12/24
- ③ 129.200.10.16/22
- ⑤ 192.168.36.32/24

[보기]

- 192.168.35.72
- 129.200.8.249
- 192.168.36.249

08. 다음은 조인에 대한 설명이다. 괄호에 들어갈 답안을 작성하시오.

- (①) : 두 테이블을 특정 조건을 기반으로 결합하는 조인 방법이다. 이 조인은 조건이 '='(동등)뿐만 아니라 '<', '>', '<=', '>=', '!=', '=>' 등의 비교 연산자를 사용할 수 있다. 이 조인을 통해 다양한 조건을 기반으로 두 테이블 간의 관계를 정의하고, 조건에 맞는 행을 결합하여 새로운 결과 집합을 만든다.
- (②) : 두 테이블을 결합할 때, 조인 조건으로 '='(동등) 연산자를 사용하는 조인 방법이다. 이 조인은 두 테이블의 특정 컬럼 값이 동일한 행들을 결합하여 결과 집합을 만든다. 주로 관계형 데이터베이스에서 외래 키와 기본 키를 매칭할 때 사용되며, 결합된 행들만 결과 집합에 포함된다.
- (③) : 두 테이블을 결합할 때, 동일한 이름의 컬럼을 기반으로 동등 비교를 수행하여 조인하는 방법이다. 이 조인은 조인 조건을 명시하지 않아도 동일한 이름의 컬럼들을 자동으로 매칭하여 결합한다. 결과 집합에는 중복되는 컬럼이 하나만 포함되며, 두 테이블 간의 공통된 컬럼 값이 동일한 행들만 결합된다.

09. 다음 페이지 순서를 고려하여, LRU 와 LFU 알고리즘의 부재 횟수를 쓰시오. 단, 페이지 프레임은 3개라고 가정한다.

페이지 참조 순서 : 1 2 3 1 2 4 1 2 5 7

10. 아래 표에서 나타내고 있는 정규형을 쓰시오.

학생아이디	강좌명	담당강사
001	영어	P001
002	한국사	P002
003	영어	P001
003	한국사	P004
004	영어	P003
004	한국사	P004

11. 아래 SQL문장을 수행하였을 때의 결과를 표로 작성하시오.

SELECT B FROM T1 WHERE C IN (SELECT C FROM T2 WHERE D = 'k')	<<T1>>			<<T2>>		
	A	B	C	C	D	E
	1	A	X	X	K	3
	2	B	X	Y	K	3
	1	C	W	Z	S	2
	3	D	W			

12. 아래 SQL문장을 수행하였을 때의 결과를 표로 작성하시오.

SELECT COUNT(*) FROM TABLE WHERE EMPNO > 100 AND SAL >= 3000 OR EMP = 200	<<TABLE>>	
	EMPNO	SAL
	100	1000
	200	3000
	300	1500

13. 다음 C언어 코드의 출력결과를 쓰시오.

```
#include <stdio.h>
int main() {
    int v1 = 0, v2 = 35, v3 = 29;
    if(v1 > v2 ? v2 : v1) {
        v2 = v2 << 2;
    }else {
        v3 = v3 << 2;
    }
    printf("%d", v2+v3);
    return 0;
}
```

14. 다음 C언어 코드의 출력결과를 쓰시오.

```
int main(int argc, char*argv[]) {
    char str[100] = "ABCDEFGH";
    reverse(str);
    int len = strlen(str);
    for(int i=1; i<len; i+=2) {
        printf("%c",str[i]);
    }
    printf(" \n");

    return 0;
}

void reverse(char*str) {
    int len = strlen(str);
    char*p1 = str;
    char*p2 = str + len - 1;
    while(p1<p2) {
        char t = *p1;
        *p1 = *p2;
        *p2 = t;
        p1++;
        p2--;
    }
}
```

15. 다음 C언어 코드의 출력결과를 쓰시오.

```
struct ACC {
    int accNum;
    double bal;
};

void init(struct ACC *acc, int x, double y) {
    acc->accNum = x;
    acc->bal = y;
}

double sim(double base, int year) {
    double r = 1.0;
    for(int i = 0; i < year; i++) {
        r *= base;
    }
    return r;
}

void xxx(struct ACC *acc, double en) {
    if( en > 0 && en < acc->bal) {
        acc->bal -= en;
    }
    else {
        acc->bal += en;
    }
}

void yyy(struct ACC *acc) {
    acc->bal *= sim(1.1, 3);
}

int main() {
    struct ACC acc;
    init(&acc, 9981, 2200.0);
    xxx(&acc, 100.0);
    yyy(&acc);
    printf("%d and %0.2f ₩n",
        acc.accNum, acc.bal);
    return 0;
}
```

16. 다음 C언어 코드의 출력결과를 쓰시오.

```

char p[] = "It is 8";
char result[20];
int i;
for(i=0; p[i]!='\0'; i++) {
    if(isupper(p[i]))
        result[i] = (p[i]-'A'+5) % 26 + 'A';
    else if(islower(p[i]))
        result[i] = (p[i]-'a'+10) % 26 + 'a';
    else if(isdigit(p[i]))
        result[i] = (p[i]-'0'+3) % 10 + '0';
    else
        result[i] = p[i];
}
result[i] = '\0';
printf("%s\n",result);

```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90

a	b	c	d	e	f	g	h	i	j	k	l	m
97	98	99	100	101	102	103	104	105	106	107	108	109

n	o	p	q	r	s	t	u	v	w	x	y	z
110	111	112	113	114	115	116	117	118	119	120	121	122

17. 다음은 JAVA 코드이다. 실행되는 순서를 쓰시오.

```
class P {  
    int x, y;  
    P(int x, int y) { ①  
        this.x=x;  
        this y=y;  
    }  
    int getData() { ②  
        return x*y;  
    }  
}  
class C extend P {  
    int x;  
    C(int x) { ③  
        super(x+1, x);  
        this.x=x;  
    }  
    int getData(int n) { ④  
        return super.getData()+n;  
    }  
    public static void main(String[] args) { ⑤  
        P p = new C(10); ⑥  
        System.out.println(p.getData()); ⑦  
    }  
}
```

18. 다음 JAVA 코드의 출력결과를 쓰시오.

```
class Connection {  
    private static Connection _inst = null;  
    private int count = 0;  
    static public Connection get() {  
        if(_inst == null) {  
            _inst = new Connection();  
            return _inst;  
        }  
        return _inst;  
    }  
    public void count() { count++; }  
    public int getCount() { return count; }  
}  
Connection conn1 = Connection.get();  
conn1.count();  
Connection conn2 = Connection.get();  
conn2.count();  
Connection conn3 = Connection.get();  
conn3.count();  
conn1.count();  
System.out.print(conn1.getCount());
```

19. 다음 JAVA 코드의 출력결과를 쓰시오.

```
class One {
    int a, b;
    public One(int a, int b) {
        this.a = a;
        this.b = b;
    }
    public void print() {
        System.out.println(a + b);
    }
}
class Two extends One {
    int po = 3;
    public Two(int i) {
        super(i, i+1);
    }
    public void print() {
        System.out.println(po*po);
    }
    public static void main(String[] args) {
        One one = new Two(10);
        one.print();
    }
}
```

20. 다음 Python 코드의 출력결과를 쓰시오.

```
a = ["Seoul", "Kyeonggi", "Incheon", "Daejun", "Daegu", "Pusan"]
str = "S"
for i in a:
    str = str + i[1]

print(str)
```

정답				
01	① ② ④ ③			
02	OSPF(Open Shortest Path First)			
03	변경 조건/결정 커버리지			
04	Rootkit			
05	APT(Advanced Persistent Threat)			
06	Abstract Factory			
07	② 192.168.35.72 ④ 129.200.8.249 ⑥ 192.168.36.249			
08	① 세타 조인 ② 동등 조인 ③ 자연 조인			
09	① LRU : 6 ② LFU : 6			
10	제 3 정규형			
11	<table><tr><td>B</td></tr><tr><td>A</td></tr><tr><td>B</td></tr></table>	B	A	B
B				
A				
B				
12	<table><tr><td>COUNT(*)</td></tr><tr><td>1</td></tr></table>	COUNT(*)	1	
COUNT(*)				
1				
13	151			
14	GECA			
15	9981 and 2795.10			
16	Nd sc 1			
17	5 → 6 → 3 → 1 → 7 → 2			
18	4			
19	9			
20	Seynaau			