

정보처리기사 실기 기출해설

정규화 Normalization	
반정규화 De-Normalization	
이상현상 Anomaly	
XML eXtensible Markup Language	
JSON JavaScript Object Notation	
CSV Comma Separated Values	
YAML	
살충제 패러독스	
오류-부재의 궤변	
보안의 3요소	
Smurf Attack	
Ping of Death	
Teardrop Attack	
LAND Attack	
SQL Injection	
XSS Cross Site Scripting	
Rainbow Table Attack	

모듈 결합도 유형 6 <u>데스 제외 공내</u>	
데이터 결합도 Data	
스탬프 결합도 Stamp	
제어 결합도 Control	
외부 결합도 External	
공통 결합도 Common	
내용 결합도 <u>Content</u>	
모듈 응집도 유형 6 <u>우논시절 통순기</u>	
기능적 응집도 Functional	
순차적 응집도 <u>Sequential</u>	
통신적 응집도 Communication	
절차적 응집도 Procedural	
시간적 응집도 <u>Temporal</u>	
논리적 응집도 Logical	
우연적 응집도 Coincidental	
애플리케이션 성능지표 4	
RIP	
OSPF	

트랜잭션 Transaction						
A C I D	원자성 Atomicity					
	일관성 Consistency					
	독립적 Isolation					
	영속성 Durability					
데이터 마이닝						
통신 프로토콜 기본요소 3						
프로세스 스케줄링						
FCFS First Come First Service						
SJF						
HRN						
SRT						
암호화 알고리즘						
두부이결다조 정규화						
UI 설계 원칙 4 직학유유						
SELECT문		<table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>				

OSI 7 참조 모델			
네트워크 계층 프로토콜			
전송 계층 프로토콜	전송계층		
응용 계층 프로토콜	응용계층		
물리 계층 장비			
데이터링크 계층 장비			
네트워크 계층 장비			
전송 계층			
데이터링크 계층			
나선형 모형 Spiral Model			
XP 의피존용담			
SCRUM			
<u>SOAP</u>			
형상 관리 절차			
버전 관리 도구			
AJAX			
애플리케이션 테스트 유형 분류	프로그램 실행 여부		
	테스트 기법		
	테스트에 대한 시각		
블랙박스 테스트 동원인비오경			
테스트 기반			

IPSec																												
IPSec의 헤더 프로토콜																												
SSL, SHTTP																												
goF 디자인 패턴 생구행																												
NAT Network Address Translation 네트워크 주소 변환																												
chmod 설정	<table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>2</td><td>1</td><td>4</td><td>2</td><td>1</td><td>4</td><td>2</td><td>1</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> chmod a.txt										4	2	1	4	2	1	4	2	1									
4	2	1	4	2	1	4	2	1																				
<u>RTO</u> Recovery Time Objective 복구 시간 목표																												
<u>RPO</u> Recovery Point Objective 복구 시점 목표																												
시멘틱 웹																												
REST																												
형상 통제																												
<u>EAI</u> Enterprise Application Integration 기업 응용 프로그램 통합 4																												
코드 커버리지 Code Coverage (테스트 커버리지)																												

스키마																			
관계 대수 연산자																			
라우팅 프로토콜 Routing Protocol																			
라우팅 프로토콜 영역에 따른 분류																			
RIP Routing Information Protocol 라우팅 정보 프로토콜																			
OSPF Open Shortest Path First																			
SOAP Simple Object Access Protocol 단순 객체 접근 프로토콜																			
생성자																			
구조 다이어그램 Structure Diagram																			
클래스 관계 Class Relationship	<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>																		
행위 다이어그램 Behavior Diagram																			
유스케이스 다이어그램 Use Case Diagram	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																		

테스트 오라클 Test Oracle	
SQL 예약어 min(), max() group by having 조건	
회복기법	
듀얼스택 Dual Stack	
터널링 Tunneling	
주소변환 Address Translation	
Hadoop 하둡	
Apache Sqoop 스콧	
NoSQL	
빅데이터 Big Data	
NAT Network Address Translation 네트워크 주소 변환	
데이터 모델 구성 요소3 구연제	

접근 권한 접근 제어 방식			
V-모델 단통시인			
정규화, 키			
정보보안 3A, AAA			
소프트웨어 생산성 향상 3R			
신기술			
RADIUS			
로킹 Locking			
패킷 교환 방식 Packet Switching			
SQL	UPDATE		
	INSERT		
	DELETE		
	JOIN		
		INDEX	
	ALTER		
	TRUNCATE		
	GRANT		
	REVOKE		
		count(*)	

정보처리기사 실기 기출전체풀이

기출복원 01회 / 2020년 1회 – 용어정리, SQL											
1. 비정규화의 개념을 약술하시오.											
2. 특수한 목적을 갖는 마크업 언어를 만드는 데 사용되는 다목적 <u>마크업 언어</u> 이다.											
3. 살충제 패러독스에 개념을 약술하시오.											
4. 재사용 모듈을 설계할 때는 누구나 쉽게 이해하고 사용할 수 있도록 사용법이 공개되어야 하고, 공유도와 ①은/는 높이고, 제어도와 ②은/는 낮추어 설계/구성되어야 하며 유지보수가 용이하고 지나치게 제한적이지 않아야 한다.											
5. 모듈의 Fan-in 수 확인하기											
6. ()은/는 기존의 XML을 대체하는 독립적인 개방형 표준 형식이다. 자바 스크립트를 기반으로 개발되었으며 데이터 객체를 속성과 값의 쌍으로 표현한다. 보편적으로 AJAX 기술에서 많이 사용된다.											
7. 다음 설명은 성능 측정 지표이다. ① : 정해진 시간에 처리할 수 있는 연산, 트랜잭션의 수이다. ② : 명령이 입력된 후 응답 출력이 개시될 때까지의 시간이다. ③ : 사용자가 데이터 및 명령 입력 시점~트랜잭션 처리 후 결과 출력 완료까지 걸리는 시간											
9. ()은/는 공격 대상의 송수신측 주소를 동일하게 변조하여, 송신측에서 전달하는 패킷이 다시 송신측으로 전달되는 루프 현상을 발생시켜 시스템에 심각한 장애를 일으키는 수법이다.											
13. 통신 프로토콜(Communication Protocol)의 기본 구성 요소 3가지를 쓰시오.											
14. 프로세스 스케줄링 기법 HRN(Highest Response-ratio Next)의 우선순위 공식을 쓰시오.											
15. 트랜잭션 Transaction 의 주요 특성에 대한 설명이다. <table border="1"><tr><td>①</td><td>트랜잭션의 연산은 모두 실행되거나, 모두 실행되지 않아야 한다.</td></tr><tr><td rowspan="2">일관성</td><td>트랜잭션을 마친 후에도 시스템이 가지는 고정 요소는</td></tr><tr><td>트랜잭션 후에도 이전과 같아야 한다.</td></tr><tr><td>②</td><td>하나의 트랜잭션 실행은 동시에 실행 중인 다른 트랜잭션의 간섭을 받으면X</td></tr><tr><td>지속성</td><td>트랜잭션의 결과는 항상 유지, 보존되어야 한다.</td></tr></table>	①	트랜잭션의 연산은 모두 실행되거나, 모두 실행되지 않아야 한다.	일관성	트랜잭션을 마친 후에도 시스템이 가지는 고정 요소는	트랜잭션 후에도 이전과 같아야 한다.	②	하나의 트랜잭션 실행은 동시에 실행 중인 다른 트랜잭션의 간섭을 받으면X	지속성	트랜잭션의 결과는 항상 유지, 보존되어야 한다.		
①	트랜잭션의 연산은 모두 실행되거나, 모두 실행되지 않아야 한다.										
일관성	트랜잭션을 마친 후에도 시스템이 가지는 고정 요소는										
	트랜잭션 후에도 이전과 같아야 한다.										
②	하나의 트랜잭션 실행은 동시에 실행 중인 다른 트랜잭션의 간섭을 받으면X										
지속성	트랜잭션의 결과는 항상 유지, 보존되어야 한다.										
16. OSI 7 // 전기, 기계적인 신호를 주고 받는 역할을 하며, 대표적인 장비로는 통신 케이블이나 허브, 리피터 등이 해당되는 계층이다.											
17. 릴리즈 노트(Release Note)의 작성 항목 중, 노트의 이름과 제품 이름, 버전 번호, 릴리즈 날짜 및 참고 날짜 등을 포함하는 항목은 무엇인지 쓰시오.											
18. 데이터 마이닝(Data mining)의 개념을 약술하시오.											
19. 이전에 쓰이던 MD4 알고리즘을 대체하기 위해 고안되었으며, 128bit 암호화 해시 함수를 사용 주로 프로그램이나 파일의 무결성을 검사하는 용도로 사용된다.											
20. LOC 계산문제											

SQL	8. 학생STUDENT 테이블에서 전자과 학생 50명, 정보통신과 학생 100명, 건축과 학생 50명의 정보가 저장되어 있을 때, 의 튜플 수를 쓰시오. ① SELECT DEPT FROM STUDENT; ② SELECT DISTINCT DEPT FROM STUDENT; ③ SELECT COUNT(DISTINCT DEPT) FROM STUDENT WHERE DEPT='정보통신'	
-----	---	--

비정규화, 성능 측정 지표, 네트워크 공격 기술, 통신 프로토콜 구성요소, 트랜잭션의 특징, 릴리즈 노트의 작성항목, 데이터 마이닝, MD4 알고리즘

기출복원 01회 / 2020년 1회 – C, JAVA, Python				
C	배열 스왑 기능	<pre> int a[]={ 95, 75, 85, 100, 50 }; int ii, j, temp; int n = sizeof(a) / sizeof(int); // int n=5; for(ii=0; i<n-1; ii++){ // 0부터 3까지 4번 for(j=0; j<4-ii; j++){ // 0부터 4-i까지 if(a[j]>a[j+1]){ temp=a[j]; a[j]=a[j+1] a[j+1]=temp; // j를 다음 index로 스왑 } } } for(ii=0; ii<5; ii++){ printf("%d", a[i]) } return 0; </pre>		
JAVA	새 배열 만들기	<pre> public static int[] makeArray(int n){ int[] t = new int[n]; for(int ii=0; i<n; ii++){ t[i]=i; } return t; // n length만큼 mkArr, [0, 1, 2, 3..] } public static void main(String[] atgs){ int[] a = makeArray(4); // [0, 1, 2, 3] for(int ii=0; ii<a.length; ii++) System.out.print(a[i]+" "); } </pre>		
	break없는 switch문	<pre> public static void main(String[] args){ int ii=3, k=1; switch(i){ // 3 case 1: k++; case 2: k+=3; case 3: k=0; // 0으로 초기화 case 4: k+=3; // 3 case 5: k-=10; // -7 default: k--; // -8 } System.out.print(k); } </pre>		
Python	출제문제 없음			

기출복원 02회 / 2020년 2회 – 용어정리, SQL		
1. ()은/는 고객의 요구사항 변형 민첩하고 유연하게 대응할 수 있도록 개발을 진행하는 방법론이다. 소규모 프로젝트, 숙련된 개발자, 급변하는 요구사항이 존재하는 개발 환경에 적합하며 대표적인 개발 모델에는 XP, Scrum, FDD, DSDM 등이 있다.		
2. 데이터 베이스 설계의 순서		
3. HTTP, HTTPS, SMTP 등 웹 서비스에서 사용되는 XML 기반의 메시지 프로토콜 다양한 프로그램 언어에서 쉽게 작성 가능하고 실행할 수 있는 프로토콜이다.		
4. 소프트웨어 개발의 전 과정에서 발생하는 산출물들의 버전을 관리하는 모든 활동이다. 동일 프로젝트를 여러 개발자가 동시에 개발할 수 있게 되며, 불필요한 수정을 제한한다. 버전 관리를 통해 배포본 관리에 유용하다.		
5. 차세대 웹 2.0의 기술들을 통칭하는 용어로서, 비동기식 자바 스크립트와 XML을 활용하는 기술이다. 표준 데이터 형식으로는 JSON을 사용한다.		
6. UI 설계의 기본 원칙 중, 사용자의 목적을 정확하고 빠르게 달성할 수 있어야 한다는 특성이다.		
7. ()은/는 프로그램을 실행하지 않고 코딩 표준, 코딩 스타일, 코드 복잡도 및 기타 결함 등을 발견하기 위해 사용된다. 테스트를 수행하는 사람이 작성된 소스 코드를 이해하고 있어야만 분석가능		
10. SQL Injection에 대해 약술하시오.		
14. ROLLBACK 명령에 대해 약술하시오.		
15. IP Spoofing, IP Sniffing과 같은 보안 허점을 해결하기 위한 방안으로 네트워크 계층에서 동작하며, IP패킷을 제거하거나 삽입을 불가능하게 하는 보안 프로토콜		
16. 리눅스 커널을 기반으로 동작하는 완전 개방형 플랫폼이다.		
17. 리눅스 서버에 data.txt라는 파일에 대해 다음 조건에 알맞은 권한부여 명령문을 쓰시오.		
18. 재해 복구 시스템(DRS)에서 주로 사용되는 용어로, 복구대상이 복구되기까지 최대 허용하는 대기시간		
19. 디자인 패턴의 세부 패턴과 관련된 설명에 가장 부합하는 용어 1:다 관계의 오브젝트에 대해 감시하고 있다가 특정 객체의 상태가 변하면 다른 모든 객체에 그 사항을 알리고 필요한 경우 자동으로 수정이 이루어지도록 하는 패턴		
20. 웹 상의 모든 데이터와 데이터베이스를 무료로 공개하고 연계하는 것 웹에 게시되는 데이터에 URL을 부여하고 관련 정보를 구조적으로 제공하는 연계 데이터를 저작권 없이 무료로 제공함 / 재사용, 중복최소화		

SQL	8. 학생 테이블에서 3~4학년 학생의 학번과 이름을 출력하는 SQL문을 작성하시오(IN 문법사용)		
	9. 학생 테이블에서 학생명 속성을 오름차순 정렬 인덱스를 생성하는 SQL문 작성(인덱스 idx_name)		

프로토콜, UI 설계 기본 원칙, 정적/동적 분석도구, DCL 명령어 개념, 보안 프로토콜, 플랫폼, chmod명령문, 디자인패턴의 세부패턴, Linked Open Data

기출복원 02회 / 2020년 2회 – C, JAVA, Python				
C	출제문제 없음			-
JAVA	12. 클래스 사용 메소드는 오버로딩	<pre> class Parent { void show(){ System.out.println("Parent"); } } class Child extends Parent { void show(){ System.out.println("Child") } } public class Main{ public static void main(String[] args){ Parent pa = () Child(); pa.show(); } } 처리결과 = Child </pre>		
	13. super()	<pre> class A { int a; public A(int n){ a=n; } public void print(){ System.out.println("a="+a); } } class B extends A { public B(int n) { super(n); super.print(); } } public class Main { public static void main(String[] args){ B obj = new B(10); } } </pre>		
Python	11. update([]) 여러값 push, <u>중복값 무시</u>	<pre> asia={'한국', '중국', '일본'} asia.add('베트남') asia.add('중국') asia.remove('일본') asia.update(['홍콩', '한국', '태국']) </pre>		

SQL: IN(~,~,~) , 괄호에 조건들
 CREATE INDEX ON ~() ON 테이블명(사용할 키 / 정렬)
 Python: 배열.update([]) , 중복값은 무시함

기출복원 03회 / 2020년 3회 – 용어정리, SQL		
1. 형상 통제에 대해 간략히 설명하시오.		
2. ① 미들웨어 없이 앱간 직접 연결, 변경이나 재사용 어려움 ② 단입 점점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식		
3. UI의 설계 원칙 중 직관성에 대해 약술하시오.		
4. 분기 커버리지 수행 순서 문제		
5. 요구사항 명세서를 기반으로 구현된 기능을 테스트 케이스로 설계 기능 및 인터페이스, 데이터 접근 및 성능 등의 오류를 발견하기 위해, 테스트 후반부에 진행		
13. C++의 생성자 Constructor에 대해 약술하시오.		
14. 데이터베이스 용어 중 스키마 Schema에 대해 약술하시오.		
15. 릴레이션 A에서 릴레이션 B의 모든 조건을 만족하는 튜플을 제외한 후 프로젝션 연산자		
16. 라우팅 프로토콜 중 RIP의 단점을 개선한 것으로, 최단 경로 알고리즘을 사용하여 VLSM을 지원하고 홉카운트의 제한이 없어 대규모 네트워크에 적합한 라우팅 프로토콜(영문약어)		
17. 호스트나 라우터에서 IP패킷을 처리할 때 발생하는 오류 또는 예상치 못한 상황에 대한 정보를 제공할 수 있게 하는 인터넷 프로토콜로 네트워크 컴퓨터상에서 돌아가는 운영체제에서 오류 메시지를 전송받는데 사용되는 프로토콜을 영문 약어로 쓰시오.		
18. 헝가리언 표기법이란?		
19. 리팩토링의 목적		
20. ()은/는 컴퓨터와 컴퓨터, 원거리 통신 장비 사이에서 메시지를 주고 받는 통신 규약 통신을 원하는 두 개체 간에 무엇을, 어떻게, 언제 통신할 것인가에 대해 미리 정의된 상호 간의 약속이다. 외교상의 의례나 국가 간의 약속을 정한 의정서에서 유래됨		

SQL	6. 테이블 속성 추가		
	7. 성적 테이블에서 과목별 평균 점수가 70점 이상 과목이름, 최소점수, 최대점수 출력		
	8. 학생 테이블에서 길동 튜플 삭제		

C++ 개념, 릴레이션 연산자 기호, OSPF, 리팩토링
SQL문법(ALTER ADD, AS+GROUP BY+HAVING),

기출복원 03회 / 2020년 3회 – C, JAVA, Python				
C	9.	<pre>int i=0, c=0; while(i<10){ // 0~9까지 i++; // i++ c*=i; } printf("%d", c);</pre>		
	10.	<pre>int r1 (){ return 4; } int r10 () { return (30+r1()); } int r100 (){ return (200+r10()); } void main(){ printf("%d", r100()); }</pre>		

JAVA	11.	<pre>public static void main(String[] args){ int i = 0; int sum = 0; while(ii<10){ ii++; if(ii%2==1) continue; sum+=i; } System.out.print(sum); }</pre>		
	12. abstract	<pre>abstract class Vehicle { String name; abstract public String getName(String val); public Vehicle(String val){ this.name=val; // this의 name } public String getName(){ <u>return</u> "Vehicle name: "+name; } } class Car extends Vehicle { public Car(String val){ super(val); // name=super.name=val; } public String getName(String val){ return "Car name : "+val; } public String getName(byte val[]){ 문자열배열? return "Car name : "+val; } } public class Main{ public static void main(String[] args){ Vehicle obj=new Car("Spark"); System.out.println(obj.getName()); } }</pre>		

기출복원 04회 / 2020년 4/5회 – 용어정리, SQL		
1. goF의 디자인 패턴에는 생성, 구조, ()가 있다. 생성: 객체의 생성에 관련된 패턴 구조: 클래스나 객체를 조합해 더 큰 구조를 만드는 패턴 (): 기능의 구체적인 알고리즘과 객체들의 상호작용에 관련된 패턴		
2. 클래스와 같은 여러 모델 요소들을 그룹화하여 표현하는 수단 서로 다른 패키지들 사이의 의존 관계를 표현하기 위한 UML 다이어그램		
3. 테스트 오라클 유형 / 특정 몇몇 테스트 케이스의 입력 값들에 대해서만 결과를 제공함		
4. 사용자 요구사항에 대한 명세를 테스트 케이스로 만들어 구현 여부를 확인하는 테스트 유형 중, 입력 조건에 유효한 값과 무효한 값을 균등하게 하여 테스트 케이스를 설계/수행		
6. 스니핑Sniffing이란?		
7. 정보 보안 3원칙은 C기밀성 I무결성 A가용성을 만족시키는 것이다. 가용성에 대해서 약술하시오.		
15. 프로세스 상태 전이도 빈칸 채우기		
16. 전 세계의 인터넷이 가능한 기기에 부여되는 유일한 식별 주소 전송 속도가 빠르고 서비스별로 패킷을 구분할 수 있어 품질 보증이 용이하다. 인증성, 기밀성, 데이터 무결성의 지원으로 보안 문제를 해결할 수 있다. 점보그램 옵션 설정을 통해 패킷 크기 제한을 없앨 수 있어서 대역폭이 넓은 네트워크 효율적 사용		
17. C언어, 셸, 계층		
18. 오픈소스를 기반으로 한		
19. P2P 네트워크, 비트코인		
20. 사설 IP주소를 공인 IP주소로 바꿔주는 주소 변환기, 사설망 보호		

SQL	5. 학생 테이블에서 학과별 튜플수를 출력 (AS, GROUP BY 사용)	
-----	--	--

goF 디자인패턴(생성, 구조, 행위) , 프로세스 상태 전이도(생성, 준비, 실행, 대기, 완료) ready, run! , NAT		
SQL: SELECT COUNT(*) AS 카운트수 FROM 테이블 GROUP BY 종류 // 종류별로 카운트! Python: foreach // for <u>sub</u> in for <u>item</u> in <u>sub</u>		

기출복원 04회 / 2020년 4/5회 - C, JAVA, Python					
C	8. 포인터 문제	char *p="KOREA";			
JAVA	9. 배열 이해	<div>처리결과: 1 4 7 10 13 2 5 8 11 14 3 6 9 12 15</div> <pre> public static void main(String[] args){ int[][] array = new int[①][②]; for(int ii=0; ii<3; ii++){ // 0~2 3번 for(int j=0; j<5; j++){ // 0~4 5번 array[i][j]=j*3+(i+1); // System.out.print(array[i][j]+" "); } System.out.println(); } } </pre>			
	10. 음수 % int는 0임 while문 주의!	<div>처리결과: 0001010</div> <pre> public static void main(String[] args){ int a[] = new int[8]; int ii=0, n=10; while(①){ a[i++] = ② ; n /= 2; } for(ii=7; i>=0; ii--) // 7~0 8번 System.out.printf("%d", a[i]); } </pre>			
	11. 오버라이딩 개념 +재귀함수☆	<pre> class Parent { int compute(int num){ if(num <= 1) return num; return compute(num-1)+compute(num-2); } } class Child extends Parent { @Override int compute(int num){ if(num <= 1) return num; return compute(num-1)+compute(num-3); } } public class Main{ public static void main (String[] args){ Parent obj = new Child(); System.out.print(obj.compute(4)) } } </pre>			
Python	12. for sub in for item in sub print() : 개행 print(end=" "): 공백	<pre> lol = [[1, 2, 3], [4, 5], [6, 7, 8, 9]] print(lol[0]) print(lol[2][1]) for sub in lol: for item in sub: print(item, end=" ") // 줄바꿈을 공백으로 대체 print() </pre>			

기출복원 05회 / 2021년 1회 – 용어정리, SQL		
1. MAC주소에서 IP주소를 알아내는 프로토콜		
2. ① 구체적으로 데이터베이스를 제작하는 단계 ② 현실 세계를 내포하고 있는 의미, E-R모델이 사용된다. ③ 개체와 개체 간의 관계를 테이블 형태로 정의한다.		
3. 웹 서비스 기술 언어 또는 기술된 정의 파일의 총칭, XML로 기술됨		
4. 요구사항 분석		
8. 완벽한 수준의 정규화를 진행하면 일관성과 안정성은 증가하지만 성능이 느려질 수 있기 때문에 성능향상을 위해 릴레이션을 통합하는 과정이다.		
9. ①: 경계 부분에 결함이 발견될 확률이 높다는 특성을 이용한 테스트 기법 ②: 프로그램의 입력값과 출력값이 특정 그룹으로 분류되며, 분류된 그룹 내의 값들이 시스템에 의해 동일하게 취급되는 특성을 이용한 테스트 기법. 유효값과 무효값 등으로 그룹화, 대표값사용		
10. V모델 테스트		
11. IP주소에 대한 설명 - IPv6는 ①비트씩 8부분으로 구성되어 128비트로 이루어진다. 16진수로 나타내며 콜론(:)으로 구분한다. - IPv4는 ②비트씩 4부분으로 구성되어 32비트로 이루어진다. 10진수로 나타내며 점(.)으로 구분한다.		
12. 프로세스들 사이에 서로 데이터를 주고 받는 행위 또는 그에 대한 방법이나 경로. 소켓, 세마포어, 공유 메모리		
13. EAI		
14. 카디널리티, 디그리		
16. ① 데이터베이스에 표현될 대상, 개체 타입과 개체 타입들 간의 관계 ② 저장된 실제 데이터를 처리하는 방법 및 작업에 대한 명세 제약 조건: 저장될 수 있는 데이터의 논리적인 제약 조건		
17. 시스템 객체의 접근을 개인 또는 그룹의 식별자에 기반한 방법 어떤 종류의 접근을 가진 사용자가 다른 사용자에게 대해 권한을 허용하는 접근제어 방식		
19. ①: 모듈 간 데이터, 지역변수 등을 직접 참조하여 사용하는 경우 ②: 모듈 간의 인터페이스로 배열이나 객체 등의 자료 구조가 전달되는 경우 ③: 파라미터가 아닌 모듈 밖에 선언되어 있는 전역 변수		
20. 공격 대상이 시스템에 접속되어, 세션이 연결되어 있는 상태를 가로채는 공격이다.		

SQL	6. 실행결과 SELECT COUNT(*) FROM EMP_TBL WHERE EMPNO > 100 AND SAL >=3000 or EMPNO=200;	
-----	---	--

데이터 모델의 구성 요소, 블랙박스 테스트, SOAP(WSDL, UDDI), V모델 과정, 결합도 설명 외우기, 요구사항 IPC(Inter-process Communication), DAC(Mandatory Access Control)		
---	--	--

기출복원 05회 / 2021년 1회 – C, JAVA, Python				
C	15.	<pre> struct insa{ char name[10]; int age; } a[] = {"Kim", 28, "Lee", 38, "Han", 32}; int main(){ struct insa *p; p=a; p++; printf("%s\n", p->name); printf("%d\n", p->age); return } </pre>		
JAVA	7.	<pre> public static void main(String[] args){ int a[][]={{45, 50, 75}, {89}}; System.out.println(a[0].length); // 3 System.out.println(a[1].length); // 1 System.out.println(a[0][0]); // 45 System.out.println(a[0][1]); // 50 System.out.println(a[1][0]); // 89 } </pre>		
	18.	<pre> public static void main (String[] args){ int j, i; for (j=0, ii=0; i<=5; ii++){ j+=ii; System.out.print(i); if(ii==5){ System.out.print("="); System.out.print(j); } else { System.out.print("+"); } } } </pre>		
Python	5. for ~ in ~	<pre> class arr: a = ["Seoul", "Kyeonggi", "Inchon", "Deajoen", "Daegu", "Pusan"] str01="" for ii in arr.a: str01+=i[0] print(str01) </pre>		

기출복원 06회 / 2021년 2회 - 용어정리, SQL		
1. 네트워크 장치 필요x, 네트워크 토폴로지가 동적으로 변화 긴급 구조, 긴급 회의, 전시 군사 네트워크		
2. ①: 사람의 감정이나 경험을 나타내는 개념 ②: 사용자 인터페이스, ex) CLI		
3. 트랜잭션의 특징 / 원자성 Atomicity		
4. 부분 함수적 종속성을 제거하여 완전 함수적 종속을 만족하는 정규형은?		
8. DES를 대체하는 대칭키 암호화 방식		
9. 화이트 박스 테스트 검증 기준 ① 최소 한 번은 모든 문장을 수행한다, ② 결정 검증기준이라고도 하며, 조건 별로 T/F ③ 전체 조건식에 상관없이 개별 조건식의 T/F		
11. 응집도 개념 ① 입출력 간 연관성은 없으나, 순서에 따라 수행한다. ② 동일한 입출력을 사용한다. ③ 하나의 기능에 모두 기여하고, 밀접하게 관련되어 있다.		
12. 패킷 교환 방식 ① 목적지 호스트와 미리 연결한 후, 통신하는 연결형 교환 방식 ② 헤더에 붙어서 개별적으로 전달하는 비연결형 교환 방식		
13. 디자인 패턴 / () 패턴은 반복적으로 사용되는 객체들의 상호작용을 패턴화 한 것이다. Interpreter, Observer, Command		
14. 병행제어기법 중, 접근한 데이터에 대한 연산을 모두 마칠 때까지 상호배제하는 기법		
15. 린바우 데이터 모델링 기법 ① 입력값이 출력값일 때, 자료흐름도-DFD ② 시간에 따라 변하는 것, 상태변화도-DFD ③ 구조, ERD		
20. 하향식 테스트 시, 하위 모듈 역할		

SQL	5. 튜플 수정 (UPDATE, SET)		
	6. Inner Join문 (JOIN, ON)		
	10. '이'씨 성을 가진 사람의 이름을 내림차순 (Like %)		

패킷 교환 방식, 병행제어기법(로킹), 데이터 모델링 기법 개념	
SQL: JOIN ON	
C: n=n*int랑 n*=int의 결과가 다름 // 남는 시간에 코딩문제 복기 꼭 하기~~ 포인터 실수 진짜 주의	

기출복원 06회 / 2021년 2회 - C, JAVA, Python				
C	16. <u>res = res * 2 와</u> <u>res *= 2 구분하기</u>	<pre> int mp(int base, int exp); int main(){ int res; res=mp(2, 10); printf("%d", res); return 0; } int mp(int base, int exp){ int res=1; for(int ii=0; ii<exp; ii++){ res=res*base; } return res; } </pre>		
	18. 배열과 포인터 값	<pre> int ary[3]; int s=0; *(ary+0)=1; // [1, 0, 0] ary[1]=*(ary+0)+2; // [1, 2, 0] ary[2]=*ary+3; // [1, 2, 4] for(int ii==0; i<3; ii++){ s=s+ary[i]; } printf("%d", s) </pre>		
JAVA	17.	<pre> public static void main(String[] args){ System.out.print(test.check(1)); } () String check (int num){ return(num>=0)?"P":"N" } </pre>		
	19. 문제 오류	<pre> public class over { public static void main(String[] args){ ovr a1=new ovr1(); ovr 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 { } </pre>		
Python	7. 시프트함수 (함정있음)	<pre> a=100 result=0 for ii in range(1,3); result = a>>i // 시프트함수 result = result+1 print(result) </pre>		