



시험에 나오는 것만 공부한다!

**시나공시리즈**

# 기출문제 & 정답 및 해설 2025년 1회 정보처리기사 필기



## 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.

※ 다음 문제를 읽고 알맞은 것을 골라 답안카드의 답란(①, ②, ③, ④)에 표기하시오.

### 제1과목 소프트웨어 설계

#### 1. 소프트웨어 공학에서 워크스루(Walkthrough)에 대한 설명으로 틀린 것은?

- ① 사용사례를 확장하여 명세하거나 설계 다이어그램, 원시코드, 테스트 케이스 등에 적용할 수 있다.
- ② 복잡한 알고리즘 또는 반복, 실시간 동작, 병행 처리와 같은 기능이나 동작을 이해하려고 할 때 유용하다.
- ③ 인스펙션(Inspection)과 동일한 의미를 가진다.
- ④ 단순한 테스트 케이스를 이용하여 프로덕트를 수작업으로 수행해 보는 것이다.

#### 2. 애자일 방법론에 해당하지 않는 것은?

- ① 기능 중심 개발                      ② 개발 및 검증
- ③ 익스트림 프로그래밍                ④ 칸반

#### 3. 익스트림 프로그래밍에 대한 설명으로 틀린 것은?

- ① 대표적인 구조적 방법론 중 하나이다.
- ② 소규모 개발 조직이 불확실하고 변경이 많은 요구를 접하였을 때 적절한 방법이다.
- ③ 익스트림 프로그래밍을 구동시키는 원리는 상식적인 원리와 경험을 최대한 끌어 올리는 것이다.
- ④ 구체적인 실천 방법을 정의하고 있으며, 개발 문서 보다는 소스 코드에 중점을 둔다.

#### 4. 럼바우(Rumbaugh) 분석 기법에서 정보 모델링이라고도 하며, 시스템에서 요구되는 객체를 찾아내어 속성과 연산 식별 및 객체들 간의 관계를 규정하여 다이어그램을 표시하는 모델링은?

- ① Object                                      ② Dynamic
- ③ Function                                    ④ Static

#### 5. 설계 기법 중 하향식 설계 방법과 상향식 설계 방법에 대한 비교 설명으로 가장 옳지 않은 것은?

- ① 하향식 설계에서는 통합 검사 시 인터페이스가 이미 정의되어 있어 통합이 간단하다.
- ② 하향식 설계에서 레벨이 낮은 데이터 구조의 세부 사항은 설계 초기 단계에서 필요하다.
- ③ 상향식 설계는 최하위 수준에서 각각의 모듈들을 설계하고 이러한 모듈이 완성되면 이들을 결합하여 검사한다.
- ④ 상향식 설계에서는 인터페이스가 이미 성립되어 있지 않더라도 기능 추가가 쉽다.

#### 6. 요구사항 명세에 대한 설명으로 틀린 것은?

- ① 분석된 요구사항을 바탕으로 모델을 작성하고 문서화하는 것이다.
- ② 기능 요구사항은 빠짐없이 완전하고 명확하게 기술해야 한다.
- ③ 잘못된 부분이 확인될 경우 그 내용을 요구사항 정의서에서 추적할 수 있어야 한다.
- ④ 구체적인 명세를 위해 자료 사전(DD)이 사용될 수 있다.

#### 7. 바람직한 소프트웨어 설계 지침이 아닌 것은?

- ① 결합도를 최소화하고 응집도를 최대화한다.
- ② 복잡도와 중복성을 줄이고 일관성을 유지시킨다.
- ③ 하나의 입구와 하나의 출구를 갖도록 해야 한다.
- ④ 모듈의 크기를 가능한 작게 구성하여 병행성 수준을 높여야 한다.

#### 8. UML(Unified Modeling Language)에 대한 설명 중 틀린 것은?

- ① 기능적 모델은 사용자 측면에서 본 시스템 기능이며, UML에서는 Use Case Diagram을 사용한다.
- ② 정적 모델은 객체, 속성, 연관관계, 오퍼레이션의 시스템의 구조를 나타내며, UML에서는 Class Diagram을 사용한다.
- ③ 동적 모델은 시스템의 내부 동작을 말하며, UML에서는 Sequence Diagram, State Diagram, Activity Diagram을 사용한다.
- ④ State Diagram은 객체들 사이의 메시지 교환을 나타내며, Sequence Diagram은 하나의 객체가 가진 상태와 그 상태의 변화에 의한 동작 순서를 나타낸다.

#### 9. 코드 설계에서 일정한 일련번호를 부여하는 방식의 코드는?

- ① 연상 코드                                      ② 블록 코드
- ③ 순차 코드                                      ④ 표의 숫자 코드

#### 10. 시스템의 구성 요소로 볼 수 없는 것은?

- ① Process                                      ② Feedback
- ③ Maintenance                                ④ Control

#### 11. 파이프 필터 형태의 소프트웨어 아키텍처에 대한 설명으로 옳은 것은?

- ① 노드와 간선으로 구성된다.
- ② 서브시스템이 입력 데이터를 받아 처리하고 결과를 다음 서브시스템으로 넘겨주는 과정을 반복한다.
- ③ 계층 모델이라고도 한다.
- ④ 3개의 서브시스템(모델, 뷰, 제어)으로 구성되어 있다.

#### 12. 데이터 흐름도(DFD)의 구성 요소에 포함되지 않는 것은?

- ① Data Flow                                      ② Data Dictionary
- ③ Process                                        ④ Data Store













정답 및 해설

1. ③	2. ②	3. ①	4. ①	5. ④	6. ④	7. ④	8. ④	9. ③	10. ③
11. ②	12. ②	13. ①	14. ②	15. ②	16. ③	17. ④	18. ①	19. ①	20. ②
21. ③	22. ①	23. ③	24. ②	25. ③	26. ①	27. ④	28. ④	29. ②	30. ④
31. ①	32. ①	33. ③	34. ①	35. ①	36. ②	37. ②	38. ④	39. ②	40. ③
41. ②	42. ②	43. ①	44. ①	45. ①	46. ③	47. ③	48. ③	49. ④	50. ④
51. ③	52. ②	53. ①	54. ②	55. ②	56. ①	57. ③	58. ③	59. ④	60. ①
61. ①	62. ①	63. ②	64. ④	65. ③	66. ④	67. ③	68. ④	69. ③	70. ②
71. ①	72. ④	73. ③	74. ③	75. ③	76. ①	77. ②	78. ①	79. ④	80. ①
81. ④	82. ①	83. ②	84. ②	85. ②	86. ④	87. ①	88. ④	89. ④	90. ③
91. ④	92. ①	93. ②	94. ①	95. ④	96. ②	97. ①	98. ②	99. ②	100. ②

- 1 인스펙션(Inspection)은 워크스루를 발전시킨 형태로, 소프트웨어 개발 단계에서 산출된 결과물의 품질을 평가하고 이를 개선하기 위한 방법 등을 제시합니다.
- 2
  - 개발 및 검증은 애자일 방법론에 해당하지 않습니다.
  - 애자일 모형을 기반으로 하는 소프트웨어 개발 모형에는 스크럼(Scrum), XP(eXtreme Programming), 칸반(Kanban), Lean, 크리스탈(Crystal), ASD(Adaptive Software Development), 기능 중심 개발(FDD, Feature Driven Development), DSDM(Dynamic System Development Method), DAD(Disciplined Agile Delivery) 등이 있습니다.
- 3 익스트림 프로그래밍(eXtreme Programming)은 구조적 방법론이 아니라, 애자일 개발 방법론을 기반으로 하는 소프트웨어 개발 모형입니다.
- 4 정보 모델링이라고 불리는 럼바우 분석 기법의 모델링은 객체 모델링(Object Modeling)입니다.
- 5 상향식 설계는 하위 모듈에서 상위 모듈 방향으로 설계하는 것으로 인터페이스가 이미 성립되어 있어야만 기능 추가가 가능합니다.
- 6 요구사항 명세 시 구체적인 명세를 위해 사용되는 것은 소단위 명세서(Mini-Spec)입니다.
- 7 모듈 크기는 시스템의 전반적인 기능과 구조를 이해하기 쉬운 크기로 분해해야 합니다.
- 8 상태 다이어그램(State Diagram)은 하나의 객체가 가진 상태와 그 상태의 변화에 의한 동작 순서를 나타내며, 시퀀스 다이어그램(Sequence Diagram)은 객체들 사이의 메시지 교환을 나타냅니다.
- 9 코드 설계에서 일정한 일련번호를 부여하는 방식의 코드는 순차 코드(Sequence Code)입니다.
- 10 시스템의 구성 요소에는 입력(Input), 처리(Process), 출력(Output), 제어(Control), 피드백(Feedback)이 있습니다.
- 11
  - 파이프 필터 형태의 소프트웨어 아키텍처에 대한 설명으로 옳은 것은 ②번입니다.
  - ①번은 자료 구조 중 그래프, ③번은 레이어 패턴, ④번은 모델-뷰-컨트롤러 패턴에 대한 설명입니다.
- 12 Data Dictionary는 자료 흐름도의 구성 요소가 아닙니다.
- 13 조작 방법은 가급적 단순화시켜 인지적 부담이 감소할 수 있게 해야 합니다.
- 14 누구나 쉽게 이해하고 사용할 수 있어야 한다는 사용자 인터페이스(UI)의 설계 원칙은 직관성입니다.
  - 유효성 : 사용자의 목적을 정확하고 완벽하게 달성해야 함
  - 학습성 : 누구나 쉽게 배우고 익힐 수 있어야 함
  - 유연성 : 사용자의 요구사항을 최대한 수용하고 실수를 최소화해야 함
- 15 디자인 패턴에 따라 소프트웨어를 개발하게 되면 요구사항을 직관적으로 구현하지 못하고 패턴에 맞게 구현해야하기 때문에 개발 초기에는 노력과 시간에 따른 비용 부담이 커집니다.
- 16 확장(Extends)은 관계의 한 형태로, 시퀀스 다이어그램의 구성 요소가 아닙니다.
- 17 문제에서 설명하는 내용은 디자인 패턴(Design Pattern)입니다.
- 18 모델링 작업의 결과물은 다른 모델링 작업에 영향을 줄 수 있습니다.
- 19 구조적 다이어그램에는 클래스(Class), 객체(Object), 컴포넌트(Component), 배치(Deployment), 복합체 구조(Composite Structure), 패키지(Package) 다이어그램이 있습니다.
- 20 하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화를 클래스(Class)라고 합니다.
- 21 [전문가의 조언]  
소스 코드 품질 분석 도구는 정적 분석 도구와 동적 분석 도구로 나뉘며, 정적 분석 도구에는 pmd, cppcheck, SonarQube, checkstyle, ccm, cobertura 등이 있고, 동적 분석 도구에는 Avalanche, Valgrind 등이 있습니다.
- 22 [전문가의 조언]  
경계값 분석(Boundary Value Analysis)은 블랙박스 테스트의 종류에 해당합니다.

23 [전문가의 조언]

소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견된다는 법칙은 파레토 법칙(Pareto Principle)입니다.

24 [전문가의 조언]

트리(Tree)에서 차수(Degree)는 가장 차수가 많은 노드의 차수입니다. 문제에 주어진 트리(Tree)에서 각 노드의 차수는 A=2, B=3, C=1, F=2이므로 B노드의 차수인 3이 트리의 차수가 됩니다.

25 [전문가의 조언]

오류를 감지하고 복구하는 기술은 디지털 저작권 관리(DRM) 기술에 속하지 않습니다.

26 [전문가의 조언]

선택 정렬은 n개의 레코드 중에서 최소값을 찾아 첫 번째 레코드 위치에 놓고, 나머지 n-1개 중에서 다시 최소값을 찾아 두 번째 레코드 위치에 놓는 방식을 반복하여 정렬하는 방식입니다.

· 원본 : 

8	3	4	9	7
---	---	---	---	---

① 1회전 : 

8	3	4	9	7
---	---	---	---	---

 → 

3	8	4	9	7
---	---	---	---	---

  
 첫 번째부터 마지막 값 중 최소값 3을 찾아 첫 번째 값 8과 위치를 교환합니다.

② 2회전 : 

3	8	4	9	7
---	---	---	---	---

 → 

3	4	8	9	7
---	---	---	---	---

  
 두 번째부터 마지막 값 중 최소값 4를 찾아 두 번째 값 8과 위치를 교환합니다.

③ 3회전 : 

3	4	8	9	7
---	---	---	---	---

 → 

3	4	7	9	8
---	---	---	---	---

  
 세 번째부터 마지막 값 중 최소값 7을 찾아 세 번째 값 8과 위치를 교환합니다.

④ 4회전 : 

3	4	7	9	8
---	---	---	---	---

 → 

3	4	7	8	9
---	---	---	---	---

  
 네 번째부터 마지막 값 중 최소값 8을 찾아 네 번째 값 9와 위치를 교환합니다.

27 [전문가의 조언]

인력은 최대한 많이 투입하는 것이 아니라 가능한 효율적으로 투입되어야 합니다.

28 [전문가의 조언]

- ④번은 Gradle에 대한 설명입니다.
- Jenkins는 Java 기반 오픈 소스 형태의 서버 서블릿 컨테이너에서 실행되는 서버 기반 도구입니다.

29 [전문가의 조언]

프로젝트 개발 비용은 형상 관리 항목에 해당하지 않습니다.

30 [전문가의 조언]

EAI의 Hybrid 유형은 단말들 사이에 버스와 허브를 미들웨어로 두는 구축 방식입니다.

31 [전문가의 조언]

큐(Queue)는 한쪽에서는 삽입이, 다른 한쪽에서는 인출이 이루어지는 선형 자료 구조입니다.

32 [전문가의 조언]

문제에서 설명하는 자료 구조는 트리(tree)입니다.

33 [전문가의 조언]

테스트(Test)를 통해 오류를 발견한 후 디버깅(Debugging)을 통해 오류 코드를 추적하고 수정하는 작업을 수행합니다.

34 [전문가의 조언]

인터페이스 구현 검증 도구에는 xUnit, STAF, FitNesse, NTAF, Selenium, watir 등이 있습니다.

35 [전문가의 조언]

소프트웨어 패키징은 개발자가 아니라 사용자를 중심으로 진행합니다.

36 [전문가의 조언]

하나 이상의 하드웨어 환경에서 운용되기 위해 쉽게 수정될 수 있는 시스템 능력을 이식성(Portability)이라고 합니다.

- **효율성(Efficiency)** : 사용자가 요구하는 기능을 할당된 시간 동안 한정된 자원으로 얼마나 빨리 처리할 수 있는지 정도를 나타냄
- **사용성(Usability)** : 사용자와 컴퓨터 사이에 발생하는 어떠한 행위에 대하여 사용자가 정확하게 이해하고 사용하며, 향후 다시 사용하고 싶은 정도를 나타냄

37 [전문가의 조언]

모듈이나 컴포넌트 단위로 기능을 확인하는 테스트는 단위 테스트(Unit Test)입니다.

- **통합 테스트(Integration Test)** : 단위 테스트가 완료된 모듈들을 결합하여 하나의 시스템으로 완성시키는 과정에서의 테스트
- **시스템 테스트(System Test)** : 개발된 소프트웨어가 해당 컴퓨터 시스템에서 완벽하게 수행되는가를 점검하는 테스트
- **인수 테스트(Acceptance Test)** : 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 테스트하는 방법

38 [전문가의 조언]

검증(Verification)은 개발자의 입장에서 개발한 소프트웨어가 명세서에 맞게 만들어졌는지를 점검하는 것이고, 확인(Validation)은 사용자의 입장에서 개발한 소프트웨어가 고객의 요구사항에 맞게 구현되었는지를 확인하는 것입니다.

39 [전문가의 조언]

강도(Stress) 테스트에 대한 설명으로 옳은 것은 ②번입니다.  
 · ①번은 회복(Recovery) 테스트, ③번은 성능(Performance) 테스트, ④번은 안전(Security) 테스트에 대한 설명입니다.

40 [전문가의 조언]

N-S 차트는 GOTO나 화살표를 사용하지 않습니다.

41 [전문가의 조언]

논리적, 물리적 데이터 구조를 정의하는 것은 데이터 정의 언어(DDL; Data Definition Language)의 기능입니다.

42 [전문가의 조언]

로킹 단위가 작아지면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가합니다.

43 [전문가의 조언]

이행적 함수 종속은 A → B이고 B → C일 때, A → C를 만족하

는 관계를 의미합니다.

44 [전문가의 조언]

- DDL(데이터 정의어)의 3가지 명령어는 CREATE, ALTER, DROP입니다.
- UPDATE는 DML(데이터 조작어)의 명령어입니다.

45 [전문가의 조언]

데이터베이스 로그를 이용해 회복 작업을 수행하는 것은 즉각 갱신 기법입니다.

46 [전문가의 조언]

트랜잭션의 연산은 데이터베이스에 모두 반영되도록 완료(Commit)되든지 아니면 전혀 반영되지 않도록 복구(Rollback)되어야 한다는 특성은 원자성(Atomicity)입니다.

47 [전문가의 조언]

차수(Degree)는 속성의 수를 의미하므로 릴레이션의 차수는 4입니다.

48 [전문가의 조언]

사용자로부터 권한을 취소(회수)하는 명령어는 revoke입니다.

```

1 revoke select
2 on department
3 from X1;
    
```

- ① 검색(select) 권한을 취소하라.
- ② <department> 테이블에 대한 권한을 취소하라.
- ③ 사용자 'X1'에 대한 권한을 취소하라.

49 [전문가의 조언]

E-R 다이어그램에서 속성은 타원으로 표시합니다.

50 [전문가의 조언]

제3정규형(3NF)에서 BCNF로 정규화하기 위해서는 모든 결정자가 후보키가 될 수 있도록 결정자가 후보키가 아닌 것을 제거해야 합니다.

51 [전문가의 조언]

뷰는 기본 테이블이나 또 다른 뷰를 이용해서 만든 가상 테이블로서, 기본 테이블과 비교할 때 삽입, 삭제, 갱신 연산에 제약이 있습니다.

52 [전문가의 조언]

테이블에 속한 튜플의 수를 카디널리티(Cardinality), 속성의 수를 차수(Degree)라고 하므로 카디널리티는 4, 차수는 6입니다.

53 [전문가의 조언]

갱신문의 기본 형식은 'UPDATE~ SET~ WHERE'입니다.

54 [전문가의 조언]

병행제어 기법의 종류에는 로킹, 타임 스탬프 순서, 최적 병행 수행, 다중 버전 기법이 있습니다.

55 [전문가의 조언]

SQL의 논리 연산자에는 AND, OR, NOT이 있습니다.

56 [전문가의 조언]

원하는 릴레이션을 정의하는 방법을 제공하는 비절차적 특성을 지닌 것은 관계해석입니다.

57 [전문가의 조언]

HAVING은 GROUP BY와 함께 사용되며, 그룹에 대한 조건을 지정할 때 사용합니다.

58 [전문가의 조언]

허가 받은 사용자만이 갱신할 수 있다는 설명은 데이터베이스의 무결성 규정(Integrity Rule)이 아닌 소프트웨어 개발 시 충족시켜야 할 보안 요소인 무결성(Integrity)에 대한 설명입니다.

59 [전문가의 조언]

분산 데이터베이스 시스템의 주요 구성 요소는 분산 처리기, 분산 데이터베이스, 통신 네트워크가 있습니다.

60 [전문가의 조언]

CREATE TABLE문에서는 속성 타입을 변경할 수 없습니다.

61 IPv6의 패킷 헤더는 40옥텟(octet)의 고정된 길이를 갖습니다.

62 모듈 간의 결합도는 약하게 해야 모듈 독립성이 향상됩니다.

63 코드의 실행 결과는 60이며, 사용된 코드의 의미는 다음과 같습니다.

```

#include <stdio.h>
struct st {           구조체 st를 정의한다.
    int a;           정수형 변수 a를 선언한다.
    int c[10];       10개의 요소를 갖는 정수형 배열 c를 선언한다.
};
int main(int argc, char* argv[]) {
    1 int i = 0;
    2 struct st ob1;
    3 struct st ob2;
    4 ob1.a = 0;
    5 ob2.a = 0;
    6 for (i = 0; i < 10; i++) {
    7     ob1.c[i] = i;
    8     ob2.c[i] = ob1.c[i] + i;
    }
    9 for (i = 0; i < 10; i = i + 2) {
    10    ob1.a = ob1.a + ob1.c[i];
    11    ob2.a = ob2.a + ob2.c[i];
    }
    12 printf("%d", ob1.a + ob2.a);
    13 return 0;
}
    
```

- ① 정수형 변수 i를 선언하고 0으로 초기화한다.
- ② 구조체 st의 변수 ob1을 선언한다.



- ③ 구조체 st의 변수 ob2를 선언한다.



- ④ ob1.a에 0을 저장한다.
- ⑤ ob2.a에 0을 저장한다.
- ⑥ 반복 변수 i가 0부터 1씩 증가하면서 10보다 작은 동안 ⑦, ⑧번을 반복 수행한다.
- ⑦ ob1.c[i]에 i의 값을 저장한다.
- ⑧ ob2.c[i]에 ob1.c[i]와 i를 합한 값을 저장한다. 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	ob1		ob2	
	a	c[i]	a	c[i]
0	0	0	0	0
1		1		2
2		2		4
3		3		6
4		4		8
5		5		10
6		6		12
7		7		14
8		8		16
9		9		18
10				

- 9 반복 변수 i가 0부터 2씩 증가하면서 10보다 작은 동안 10, 11번을 반복 수행한다.
  - 10 ob1.a에 ob1.c[i]의 값을 누적시킨다.
  - 11 ob2.a에 ob2.c[i]의 값을 누적시킨다.
- 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	ob1		ob2	
	a	c[i]	a	c[i]
0	0	0	0	0
2	2	2	4	4
4	6	4	12	8
6	12	6	24	12
8	20	8	40	16
10				

- 12 ob1.a와 ob2.a의 값을 합하여 정수로 출력한다.

**결과 60**

- 13 main() 함수에서의 'return 0'은 프로그램의 종료를 의미한다.

- 64 보기에 제시된 연산자를 우선 순위대로 나열하면, '산술 → 시프트 → 관계 → 비트' 순입니다.
- 65 코드의 실행 결과는 53 44이며, 사용된 코드의 의미는 다음과 같습니다.

```

1 a, b = map(int, input().split("t"));
2 print(a, b)
    
```

- 1 input() 메소드로 입력받은 값을 "t"를 구분자로 하여 분리한 후 정수로 변환하여 a, b에 저장한다. 문제에서 "53t44"를 입력하였으므로, "t"를 구분자로 53과 44가 분리된 후 정수로 변환되어 각각 a와 b에 저장된다.
  - map(): 2개 이상의 값을 원하는 자료형으로 변환할 때 사용하는 함수
  - input().split('분리문자')
    - 입력받은 값을 '분리문자'로 구분하여 반환한다.
    - '분리문자'를 생략하면 공백으로 값을 구분한다.
- 2 a와 b를 출력한다. Python의 print() 메소드에서 2개 이상의 값을 출력할 때, sep 속성값을 정의하지 않으면 기본값이 공백이므로 다음과 같이 출력된다.

**결과 53 44**

- 66 수신 측으로부터 이전에 송신한 패킷에 대한 긍정 수신 응답(ACK)이 전달된 경우 윈도우 크기는 증가하고, 수신 측으로부터 이전에 송신한 패킷에 대한 부정 수신 응답(NAK)이 전달된 경우 윈도우 크기는 감소합니다.
- 67 코드의 실행 결과는 8이며, 사용된 코드의 의미는 다음과 같습니다.

```

#include <stdio.h>
int main(void) {
1 int n = 4;
2 int* pt = NULL;
3 pt = &n;
4 printf("%d", &n + *pt - *&pt + n);
5 return 0;
}
    
```

- 1 정수형 변수 n을 선언하고 4로 초기화한다.
- 2 정수형 포인터 변수 pt에 Null 값을 저장한다.
- 3 pt에 n의 주소를 저장한다.
- 4 printf("%d", &n + \*pt - \*&pt + n);
  - (a) (b) (c) (d)
  - 3번을 수행했으므로 n의 주소 @와 pt에 저장된 주소를 가리키는 @는 같은 주소를 가지므로 @-@ = 0이다.
  - @에서 \*pt는 n의 값 4를 의미하고, @도 n의 값 4이므로 @+@ = 8이다.
  - @+@-(c)+@의 결과 8을 정수로 출력한다.

**결과 8**

- 5 프로그램은 종료한다.
  - ※ @와 @의 주소값은 16진 정수의 임의값을 갖지만, @-@의 연산결과로 0이 되므로 값을 알 필요는 없습니다.

- 68 리눅스에서 커널의 버전을 확인할 때 사용하는 명령어는 uname입니다.
  - ls : 현재 디렉터리 내의 파일 목록을 확인함
  - chmod : 파일의 보호 모드를 설정하여 파일의 사용 허가를 지정함
  - rm : 파일을 삭제함
- 69 배열 순회(배열 순례)는 공간 구역성에 해당됩니다.
- 70 코드의 실행 결과는 ogani이며, 사용된 코드의 의미는 다음과 같습니다.

```

#include <stdio.h>
main() {
1 char* s = "Sinagong";
2 for (int i = 5; i > 0; i--)
3     printf("%c", *(s + i));
}
    
```

- 1 문자형 포인터 변수 s를 선언하고, 문자열 "Sinagong"이 저장된 곳의 주소를 저장한다.
 

주소	s+0	s+1	s+2	s+3	s+4	s+5	s+6	s+7
값	S	i	n	a	g	o	n	g
- 2 반복 변수 i가 5에서 시작하여 1씩 감소하면서 0보다 큰 동안 3번을 반복 수행한다.
- 3 's+i'의 주소에 있는 문자를 출력한다.

**결과 ogani**

- ※ 반복문 실행에 따른 변수의 변화는 다음과 같다.

반복횟수	i	*(s+i)	출력
1	5	o	o
2	4	g	og
3	3	a	oga
4	2	n	ogan
5	1	i	ogani
반복실행 안됨	0		

71 문제의 설명은 물리 계층(Physical Layer)에 대한 설명입니다.

- **데이터 링크 계층(Data Link Layer)**
  - 두 개의 인접한 개방 시스템들 간에 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 한다.
  - 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어 기능을 한다.
- **네트워크 계층(Network Layer, 망 계층)**
  - 개방 시스템들 간의 네트워크 연결을 관리하는 기능과 데이터의 교환 및 중계 기능을 한다.
  - 경로 설정(Routing), 트래픽 제어, 패킷 정보 전송을 수행한다.
- **전송 계층(Transport Layer)**
  - 종단 시스템(End-to-End) 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능을 한다.
  - 주소 설정, 다중화(데이터의 분할과 재조립), 오류 제어, 흐름 제어를 수행한다.

72 · while문의 조건식과 println() 메소드의 인수 형식이 잘못되어서 ④번과 같은 오류가 발생하는 코드입니다.  
· 올바르게 출력하려면, 다음과 같이 수정되어야 합니다.

```

1 int x = 1, y = 6;
2 while (y-- > 0) {
3     x++;
4 }
5 System.out.println("x=" + x + " y=" + y);
    
```

- ① 정수형 변수 x, y를 선언하고 각각 1과 6으로 초기화한다.
- ② y가 0보다 큰 동안 ③번을 반복 수행한다. y는 후치감소 연산자이므로 조건식이 판별된 후 1 감소한다.
- ③ 'x = x + 1'과 동일하다. x에 1씩 누적시킨다.
- ④ "x="을 출력한 후 x의 값을, "y="을 출력한 후 y의 값을 출력하고 커서를 다음 줄의 처음으로 이동시킨다.

결과 **x=7 y=-1**

73 UNIX에서 새로운 프로세스를 생성하는 명령어는 fork입니다.  
· **cat** : 파일 내용을 화면에 표시함  
· **chmod** : 파일의 보호 모드를 설정하여 파일의 사용 허가를 지정함  
· **ls** : 현재 디렉터리 내의 파일 목록을 확인함

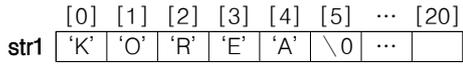
74 128.107.176.0/22 네트워크의 서브넷 마스크는 1의 개수가 22개, 즉 11111111 11111111 11111100 00000000이므로 255.255.252.0입니다.

75 코드의 실행 결과는 R이며, 사용된 코드의 의미는 다음과 같습니다.

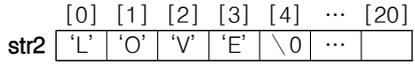
```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[]) {
1 char str1[20] = "KOREA";
2 char str2[20] = "LOVE";
3 char* p1 = NULL;
4 char* p2 = NULL;
5 p1 = str1;
6 p2 = str2;
7 str1[1] = p2[2];
8 str2[3] = p1[4];
9 strcat(str1, str2);
10 printf("%c", *(p1 + 2));
11 return 0;
}
    
```

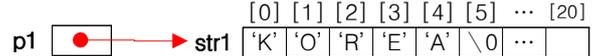
① 20개의 요소를 갖는 문자형 배열 str1을 선언하고 "KOREA"로 초기화한다.



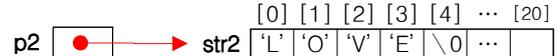
② 20개의 요소를 갖는 문자형 배열 str2를 선언하고 "LOVE"로 초기화한다.



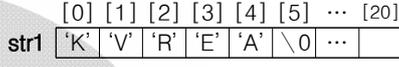
- ③ 문자형 포인터 변수 p1을 선언하고 NULL로 초기화한다.
- ④ 문자형 포인터 변수 p2를 선언하고 NULL로 초기화한다.
- ⑤ p1에 str1 배열의 시작 주소를 저장한다.



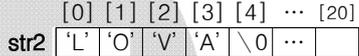
⑥ p2에 str2 배열의 시작 주소를 저장한다.



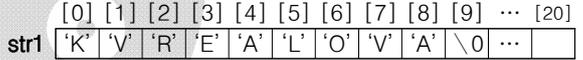
⑦ p2는 str2를 가리키므로 str2[2]의 값인 'V'를 str1[1]에 저장한다.



⑧ p1은 str1을 가리키므로 str1[4]의 값인 'A'를 str2[3]에 저장한다.



⑨ str1의 문자열 뒤에 str2의 문자열을 이어붙인다.  
· **strcat(문자배열A, 문자배열B)** : A 배열에 저장된 문자열의 마지막에 이어서 B 배열에 저장된 문자열을 이어붙인다.



⑩ p1+2가 가리키는 곳의 값을 문자로 출력한다. p1은 str1 배열의 시작주소, 즉 str1[0]의 위치를 가리키므로, p1+2는 str1[2]의 다음 두 번째 요소인 'R'을 가리킨다.

결과 **R**

⑪ main() 함수에서의 'return 0'은 프로그램의 종료를 의미한다.

76 세그먼트이션 기법에서 논리 주소(2, 176)를 물리 주소로 변환하면 세그먼트 번호(2)에 해당하는 열의 시작주소(222) + 변위값(176), 즉 398입니다.

77 Fortran은 과학 기술 계산용 언어로 절차적 프로그래밍 언어에 속합니다.

78 한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 참조하는 경우의 결합도는 내용 결합도(Content Coupling)입니다.

- **제어 결합도(Control Coupling)** : 한 모듈에서 다른 모듈로 논리적인 흐름을 제어하는 데 사용하는 제어 요소(Function code, Switch, Tag, Flag)가 전달될 때의 결합도
- **공통(공유) 결합도(Common Coupling)** : 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도
- **스탬프(검인) 결합도(Stamp Coupling)** : 모듈 간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도

79 파일 디스크립터는 보조기억장치 내에 저장되어 있다가 해당 파일이 오픈될 때 주기억장치로 옮겨집니다.

- 80 stdlib.h에 대한 설명으로 옳은 것은 ①번입니다.  
· ②번은 string.h, ③번은 stdio.h, ④번은 math.h에 대한 설명입니다.
- 81 문제에 제시된 내용은 부인 방지의 개념입니다.  
· **기밀성(Confidentiality)** : 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용됨  
· **무결성(Integrity)** : 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있음  
· **가용성(Availability)** : 인가받은 사용자는 언제라도 사용할 수 있음
- 82 네트워크를 통해 연속적으로 자신을 복제하는 악성코드는 웜(Worm)입니다.  
· **로그웨어(Rogue Ware)** : 사용자를 속여 악성코드를 설치하도록 유도하는 소프트웨어로, 주로 바이러스에 감염되었다며 백신 소프트웨어처럼 보이는 악성코드를 설치하도록 유도함  
· **애드웨어(Adware)** : 소프트웨어 자체에 광고를 포함하여 이를 보는 대가로 무료로 사용하는 소프트웨어  
· **반사 공격(Reflection Attack)** : 송신자가 생성한 메시지를 가로채 접근 권한을 얻는 형태의 공격 기법
- 83 소프트웨어 비용 산정 기법에는 하향식 비용 산정 기법과 상향식 비용 산정 기법이 있습니다. 혼합식 비용 산정 기법은 존재하지 않습니다.
- 84 문제의 내용은 위험 관리(Risk Analysis)에 대한 설명입니다.
- 85 문제에 제시된 내용은 PaaS-TA에 대한 설명입니다.  
· **N2OS(Neutralized Network Operating System)** : 한국전 자통신연구원(ETRI)이 개발한 네트워크 운영체제로, 다양한 네트워킹 기술을 적용할 수 있는 개방형 구조의 소프트웨어 프레임워크임  
· **메타버스(Metaverse)** : 가상, 추상을 의미하는 'meta'와 우주, 현실세계를 의미하는 'Universe'의 합성어로 3차원 가상세계를 의미함. 기존의 가상 현실(Virtual Reality) 보다 진보된 개념으로 웹과 인터넷 등의 가상 세계가 현실 세계에 흡수된 형태임
- 86 AES는 개인키 암호화 알고리즘이고, MD4, MD5, SHA-1은 해시 알고리즘입니다.
- 87 최대 홉수를 15로 제한한 라우팅 프로토콜은 RIP(Routing Information Protocol)입니다.
- 88 Sequential이라는 COCOMO 유형은 존재하지 않습니다.
- 89 RSA(Rivest Shamir Adleman)는 공개키 암호화 기법에 속합니다.
- 90 PIMS는 정보통신망에서 개인정보보호를 위한 관리체계로써 Secure SDLC와는 관련이 없습니다.  
· **CLASP(Comprehensive Lightweight Application Security Process)** : Secure Software 사에서 개발한 개발 방법론으로 생명주기 초기 단계의 보안 강화를 목적으로 함  
· **SDL** : 마이크로소프트 사에서 안전한 소프트웨어 개발을 위해 기존의 SDLC를 개선한 방법론  
· **Seven Touchpoints** : SW 보안의 모범 사례를 SDLC에 통합한 소프트웨어 개발 보안 생명주기 방법론
- 91 간트 차트(Gantt Chart)에서 수평 막대의 길이는 각 작업(Task)의 기간을 나타냅니다.
- 92 문제의 지문은 SDS(Software Defined Storage)에 대한 설명입니다.
- 93 대칭 암호 알고리즘은 비밀키(Private Key)를 공유해야 하기 때문에 키 관리의 어려움이 있지만 알고리즘이 단순하여 암호화 및 복호화 속도가 빠릅니다.
- 94 나선형 모형의 주요 태스크에는 계획 수립, 위험 분석, 개발 및 검증, 고객 평가가 있습니다.
- 95 기존 소프트웨어를 다른 운영체제나 하드웨어 환경에서 사용할 수 있도록 변환하는 활동을 이식(Migration)이라고 합니다.
- 96 문제에 제시된 내용은 만리장성 모델(Chinese Wall Model)의 개념입니다.  
· **벨 라파둘라 모델(Bell-LaPadula Model)** : 군대의 보안 레벨처럼 정보의 기밀성에 따라 상하 관계가 구분된 정보를 보호하기 위해 사용하는 모델  
· **비바 무결성 모델(Biba Integrity Model)** : 벨 라파둘라 모델을 보완한 수학적 모델로, 무결성을 보장하는 최초의 모델  
· **클락-윌슨 무결성 모델(Clark-Wilson Integrity Model)** : 무결성 중심의 상업용 모델로, 사용자가 직접 객체에 접근할 수 없고 프로그램에 의해 접근이 가능한 보안 모델
- 97 테일러링은 프로젝트 상황 및 특성에 맞추어 기존의 방법론, 프로세스 등을 수정하는 것이니 만큼, 예상되는 변화를 충분히 고려해야 합니다.
- 98 Public 메소드로부터 Private 배열이 반환될 경우 배열의 주소가 외부로 공개되어 외부에서 Private 배열에 접근할 수 있게 됩니다.
- 99 공개키 암호화 알고리즘은 해시 기법이 아닌 소인수 분해나 이산대수 등 고유의 방법으로 키를 생성합니다.
- 100 Smurf 공격은 네트워크 라우터의 브로드캐스트(Broadcast) 주소를 활용한 DoS 공격입니다.