2019-서울시-컴퓨터일반-A형-해설

대방고시 전산직/계리직, 하이클래스 군무원 곽후근(gobarian@gmail.com)

1번부터 10번 해설동영상은 https://www.youtube.com/watch?v=1gEJOdw6BYQ&t=112s,
11번부터 20번 해설동영상은 https://www.youtube.com/watch?v=gJtIChV80lU&t=508s
을 참고하기 바랍니다.

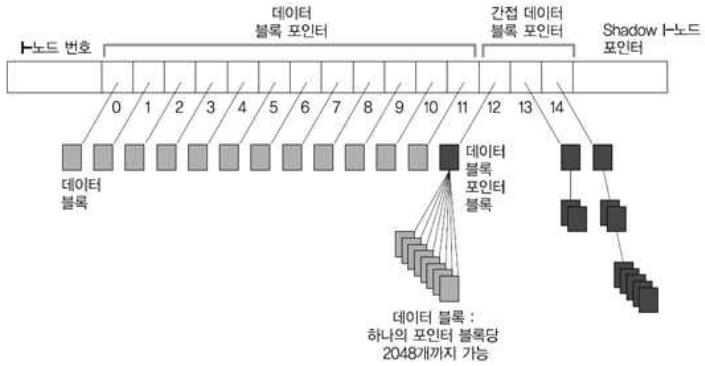
1. C 프로그램을 컴파일하면 <보기>와 같은 것들이 실행된다. 이 중 3번째로 실행되는 것은?

<보기> 링커(linker), 어셈블러(assembler), 전처리기(preprocessor), 컴파일러(compiler)

① 링커(linker)
② 어셈블러(assembler)
③ 전처리기(preprocessor)
④ 컴파일러(compiler)

정답 체크 :
C언어를 컴파일하면 전처리기, 컴파일러, 어셈블러, 링커, 로더의 순으로 동작한다.
(2) 어셈블러 : 어셈블리 언어를 이진 파일로 만든다.
오답 체크 :
소스코드를 실행 파일로 만드는 순서
(1) 링커 : 미리 컴파일한 라이브러리 파일과 이진 파일을 합친다.
(3) 전처리기 : 컴파일 전에 전처리(#include, #define 등)를 수행한다.
(4) 컴파일러 : 소스 코드를 어셈블리 언어 혹은 이진 파일로 만든다.
Tip!: 이외에 로더는 이진 파일을 메모리에 적재한다.

2. 유닉스 파일 시스템에 대한 설명으로 가장 옳지 않은 것은?
① 슈퍼블록은 전체 블록의 수, 블록의 크기, 사용 중인 블록의 수 등 파일 시스템의 정보를 가지고 있다.
② 아이노드는 파일의 종류, 크기, 소유자, 접근 권한 등 각종 속성 정보를 가지고 있다.
③ 파일마다 데이터 블록, 아이노드 외에 직접 블록 포인터와 단일 . 이중 . 삼중 간접 블록 포인터로 구성된 인덱스 정보를 가진 인덱스 블록을 별도로 가지고 있다.
④ 디렉터리는 하위 파일들의 이름과 아이노드 포인터 (또는 아이노드 번호)를 포함하는 디렉터리 엔트리들로 구성된다.
정답 체크 :
(3) 파일 구조 : 아래 그림과 같이 직접 블록 포인터는 존재하지 않는다.



[그림 13-21] I-노드의 구조

오답 체크 :

(1) 슈퍼블록 : 데이터 블록의 개수, 실린더(Cylinder) 그룹의 개수, 데이터 블록의 크기 및 조각(Fragment), 하드웨어 정보, 마운트 포인트 정보, 파일 시스템 상태 정보를 가진다.

(2) 아이노드 : 파일(file)의 종류와 접근 모드, 파일의 소유자와 그룹에 대한 UID, GID, 파일 크기, 파일의 마지막으로 접근/변조 시간, I-노드가 변경된 시간, 데이터 저장에 사용된 총 데이터 블록 수를 가진다.

(4) 디렉토리 : 파일 이름, 디렉터리 엔트리(디렉토리를 표현하는 데에 쓰이는 자료구조, 일반적으로는 파일이름, 파일속성 등 파일에 대한 여러가지 정보가 저장되는데, 유닉스 계열에서는 파일이름과 아이노드 번호만 저장된다.)를 가진다.

3. <보기>는 8비트에 부호 있는 2의 보수 표현법으로 작성한 이진수이다. 이에 해당하는 십진정수는?

<보기> 10111100

① -60

② -68

③ 94

④ 188

정답 체크 :

(2)

주어진 2진수를 십진수로 바꾸는 방법은 2의 보수를 취하고 -를 붙이면 된다.

2의 보수를 취하면 01000100, 즉 십진수 68이 된다.

십진수에 음수화를 하면 -68이 된다.

4. <보기>가 설명하는 것은?

<보기> 다음에 실행할 명령어의 주소를 보관하는 레지스터 이다. 계수기로 되어 있어 실행할 명령어를 메모리에서 읽으면 명령어의 길이만큼 증가하여 다음 명령어를 가리키며, 분기 명령어는 목적 주소로 갱신할 수 있다.

① 명령어 레지스터

② 프로그램 카운터

③ 데이터 레지스터
④ 주소 레지스터
정답 체크 :
(2) 프로그램 카운터 : 다음에 수행할 명령어의 주소를 저장한다(PC, 특수 목적 레지스터).
오답 체크 :
(1) 명령어 레지스터 : 기억장치로부터 읽어온 명령어를 수행하기 위하여 일시적으로 저장한다(IR, 특수 목적 레지스터).
(3) 데이터 레지스터 : 데이터를 저장하기 위해 사용하는 일반 목적용 레지스터이다.
(4) 주소 레지스터 : 주소를 저장하기 위해 사용하는 일반 목적용 레지스터이다.

5. 운영체제에서 가상 메모리의 페이지 교체 기법에 대한 설명으로 가장 옳지 않은 것은?
① FIFO 기법에서는 아무리 참조가 많이 된 페이지라도 교체될 수 있다.
② LRU 기법을 위해서는 적재된 페이지들의 참조된 시간 또는 순서에 대한 정보가 필요하다.
③ Second-chance 기법에서는 참조 비트가 0인 페이지는 교체되지 않는다.
④ LFU 기법은 많이 참조된 페이지는 앞으로도 참조될 확률이 높을 것이란 판단에 근거한 기법이다.
정답 체크 :
(3) Second-chance : 아래 그림과 같이 참조 비트가 1인 페이지는 교체되지 않는다.

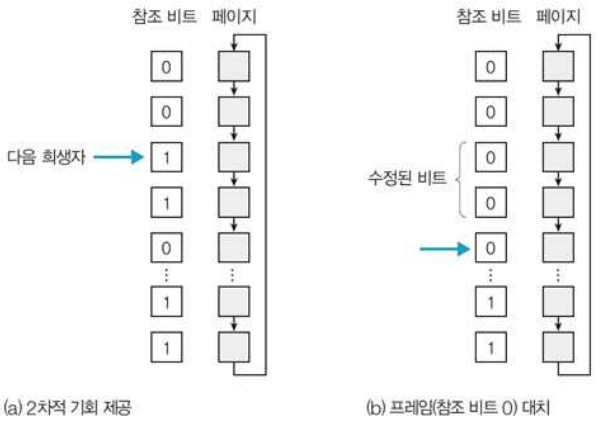


그림 8-27 시계 페이지 대치 알고리즘

오답 체크 :
(1) FIFO : 처음 들어온 페이지를 교체한다.
(2) LRU : 참조 시간이 오래된 페이지를 교체한다. (지역성의 원리이다.)
(4) LFU : 참조 횟수가 적은 페이지를 교체한다. (지역성의 원리이다.)

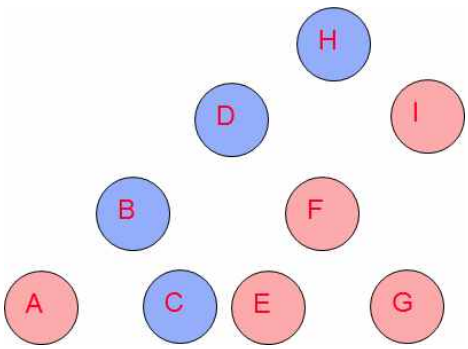
6. 네트워킹 장비에 대한 설명으로 가장 옳지 않은 것은?
① 라우터(router)는 데이터 전송을 위한 최선의 경로를 결정한다.
② 허브(hub)는 전달받은 신호를 그와 케이블로 연결된 모든 노드들에 전달한다.
③ 스위치(switch)는 보안(security) 및 트래픽(traffic) 관리 기능도 제공할 수 있다.
④ 브리지(bridge)는 한 네트워크 세그먼트에서 들어온 데이터를 그의 물리적 주소에 관계없이 무조건 다른 세그먼트로 전달한다.
정답 체크 :

(4) 브리지 : L2 switch
지문의 설명은 더미 허브를 의미한다. 브리지는 허브나 리피터보다 더 복잡한 경향을 가진다. 브리지는 들어오는 데이터 패킷을 분석하여 브리지가 주어진 패킷을 다른 세그먼트의 네트워크로 전송할 수 있는지를 결정할 수 있다. (스위칭 허브와 비슷한 개념)
오답 체크 :
(1) 라우터 : L3 switch
(2) 허브 : L2 switch(더미 허브는 MAC 주소에 상관없이 모든 포트로 브로드캐스팅, 스위칭 허브는 학습 기능을 이용해서 MAC 주소와 관련된 포트로 포워딩)
(3) 스위치 : L2 switch에도 보안 기능과 트래픽 관리 기능이 들어간다. 예전에는 L4, L7 스위치에만 적용했는데 L4, L7의 경우 로컬 네트워크에서 발생하는 보안 문제를 처리할 수 없기 때문에 L2에도 보안 기능을 제공하기 시작했다.

7. 다음의 정렬된 데이터에서 2진탐색을 수행하여 C를 찾으려고 한다. 몇 번의 비교를 거쳐야 C를 찾을 수 있는가? (단, 비교는 '크다', '작다', '같다' 중의 하나로 수행되고, '같다' 가 도출될 때까지 반복된다.)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

① 1번
② 2번
③ 3번
④ 4번
정답 체크 :
(4)
이진 탐색 트리의 원리에 따라 이진 탐색 트리를 중위 순회하면 정렬된 데이터를 아래와 같이 얻을 수 있다. 그림에서 파란색이 비교 횟수이므로 C를 찾기 위해 4번의 비교를 수행함을 알 수 있다.



8. 인터넷 서비스 관련 용어들에 대한 설명으로 가장 옳지 않은 것은?
① ASP는 동적 맞춤 형 웹 페이지의 구현을 위해 사용된다.
② URL은 인터넷상에서 문서나 파일의 위치를 나타낸다.
③ HTM L은 웹 문서의 전달을 위한 통신 규약이다.
④ SSL은 안전한 웹 통신을 위한 암호화를 위해 사용된다.
정답 체크 :

(3) HTML : 지문의 설명은 HTTP를 나타낸다. HTML은 웹 페이지를 만들기 위한 비순차적 마크업(태그) 언어이다.
오답 체크 :
(1) ASP : MS에서 동적으로 웹 페이지들을 생성하기 위해 개발한 서버 측 스크립트 엔진이다. (vs. 정적 페이지)
(2) URL : 네트워크 상에서 자원이 어디 있는지를 알려주기 위한 규약이다.
(4) SSL : 브라우저와 웹 서버 간에 데이터를 안전하게 주고받기 위한 업계 표준 프로토콜이다.

9. <보기>의 배열 A에 n 개 의 원 소가 있다고 가정하자. 다음 의사 코 드에 대한 설명으로 가장 옳지 않은 것은?

```
<보기>
Function(A[ ], n) {
  for last ← n downto 2 // last를 n에서 2까지 1씩 감소
    for i ← 1 to last-1
      if (A[i] > A[i+1]) then A[i] ↔A[i+1]; // A[i]와 A[i+1]를 교환
}
```

① 제일 큰 원 소를 끝자리로 옮기는 작업을 반복한다.
② 선택 정렬을 설명하는 의사 코드이다.
③ $O(n^2)$의 수행 시간을 가진다.
④ 두 번째 for 루프의 역할은 가장 큰 원소를 맨 오른쪽으로 보내는 것이다.
정답 체크 :
(2) 해당 정렬은 선택 정렬이 아니라 버블 정렬이다.
오답 체크 :
(1) 두 개의 for문을 통해 제일 큰 원소를 끝자리로 옮기는 작업을 반복한다.
(3) 첫번째 for문 n개를 두번째 for문이 평균 n/2번 수행하므로 연산 횟수는 $n^2/2$이고 이를 빅오로 표기하면(최고차항으로 표기하면) $O(n^2)$이다.
(4) 두번째 for문은 가장 큰 원소를 맨 오른쪽으로 보낸다.

10. <보기>의 Java 프로그램의 실행 결과는?

```
<보기>
class A {
  public void f() { System.out.print( " 1 " ); }
  public static void g() { System.out.print( " 2 " ); }
}
class B extends A {
  public void f() { System.out.print( " 3 " ); }
}
class C extends B {
  public static void g() { System.out.print( " 4 " ); }
```

```
}
public class D {
   public static void main(String args[]) {
      A obj = new C();
      obj.f();
      obj.g();
   }
}
```

① 3 2
② 3 4
③ 1 2
④ 1 4
정답 체크 :
(1)
A obj = new C(); // 자식 객체를 생성해서 부모 객체가 받음(default로 부모 객체의 메소드 호출) (자식은 부모를 포함하기 때문에 해당 관계로 성립한다)
obj.f(); // 원래 A.f()가 호출되어야 하나 동적 바인딩(실행 시간에 호출 결정)으로 인해 B.f()가 호출(다형성의 overriding 개념이다)
obj.g(); // 메소드가 static인 경우에는 동적 바인딩이 적용되지 않는다. static은 클래스를 통해 1개만 존재하기 때문이다. (상속이 무의미하다)

11. 어떤 시스템은 7비트의 데이터에 홀수 패리티 비트를 최상위 비트에 추가하여 8비트로 표현하여 저장한다. 다음과 같은 데이터를 저장 장치에서 읽어왔을 때 오류가 발생한 경우는?
① 011010111
② 101101111
③ 011001110
④ 101001101
정답 체크 :
홀수 패리티 비트 : 1의 개수가 홀수 개이면 정상이고 짝수 개이면 오류이다.
(1) : 1의 개수가 짝수 개다.

12. 고객, 제품, 주문, 배송 업체 테이블을 가진 판매 데이터베이스를 SQL을 이용해 구축하고자 한다. 각 테이블이 <보기>와 같은 속성을 가진다고 가정할 때, 다음 중 가장 옳지 않은 SQL 문은? (단, 밑줄은 기본키를 의미한다.)

<보기>
- 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)
- 제품(제품번호, 제품명, 재고량, 단가, 제조업체)
- 주문(주문번호, 주문제품, 주문고객, 수량, 배송지, 주문일자)
- 배송업체(업체번호, 업체명, 주소, 전화번호)

① 고객 테이블에 가입 날짜를 추가한다. → “ALTER TABLE 고객 ADD 가입 날짜 DATE;”
② 주문 테이블에서 배송지를 삭제한다. → “ALTER TABLE 주문 DROP COLUMN 배송지;”

③ 고객 테이블에 18세 이상의 고객만 가입 가능하다는 무결성 제약 조건을 추가한다. → “ALTER TABLE 고객 ADD CONSTRAINT CHK_AGE CHECK(나이> = 18);”
④ 배송 업체 테이블을 삭제한다. → “ALTER TABLE 배송업체 DROP;”
정답 체크 :
(4) 테이블 제거 : DROP TABLE 테이블_이름;
오답 체크 :
(1) 새로운 속성 추가 : ALTER TABLE 데이블_이름 ADD 속성_이름 데이터_타입;
(2) 기존 속성 삭제 : ALTER TABLE 테이블_이름 DROP 속성_이름; (COLUMN?)
(3) 새로운 제약 조건 추가 : ALTER TABLE 테이블_이름 ADD CONSTRAINT 제약조건_이름 제약조건_내용;

13. <보기>의 UML 다이어그램 중 시스템의 구조(structure) 보다는 주로 동작(behavior)을 묘사하는 다이어그램들만 고른 것은?

<보기>
ㄱ. 클래스 다이어그램(class diagram)
ㄴ. 상태 다이어그램(state diagram)
ㄷ. 시퀀스 다이어그램(sequence diagram)
ㄹ. 패키지 다이어그램(package diagram)
ㅁ. 배치 다이어그램(deployment diagram)

① ㄱ , ㄹ
② ㄴ , ㄷ
③ ㄴ , ㅁ
④ ㄷ , ㄹ
정답 체크 :
(2)
(ㄴ) 상태 : 특정 개체의 동적인 행위를 상태와 그것들간의 transition을 통해 묘사하는 다이어그램
(ㄷ) 시퀀스 : Instance 들이 어떻게 상호작용을 하는지를 묘사하는 다이어그램
오답 체크 :
(1), (3), (4)
(ㄱ) 클래스 : Class 관련 요소들의 여러 가지 정적인 관계를 시각적으로 표현한 다이어그램
(ㄹ) 패키지 : 관련된 클래스를 패키지로 grouping하여 의존도를 낮추기 위하여 사용(정적)
(ㅁ) 배치 : 물리적인 컴퓨터 및 장비 등의 하드웨어 요소들과 그것에 배치되는 소프트웨어 컴포넌트, 프로세스 및 객체들의 형상을 묘사하는 다이어그램(정적)

14. <보기 1>의 테이블 R에 대해 <보기 2>의 SQL을 수행한 결과로 옳은 것은?

<보기 1>

A	B
3	1
2	4
3	2
2	5
3	3
1	5

<보기 2>

```
SELECT SUM(B) FROM R GROUP BY A
HAVING COUNT(B) = 2;
```

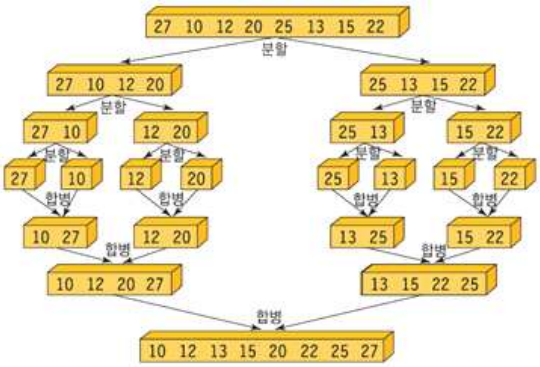
① 2
② 5
③ 6
④ 9
정답 체크 :
GROUP BY A HAVING COUNT(B) = 2 // A를 그룹으로 묶었을 때 동일한 속성값의 개수(중복의 개수)가 2인 것을 찾는다. (3은 3개, 2는 2개, 1은 1개), GROUP BY A HAVING COUNT(*) = 2와 같이 써도 무방하다.
SUM(B) // 찾아진 2에서 B의 값을 합하면 9(=4 + 5)가 된다.

15. <보기>는 데이터가 정렬되는 단계를 일부 보여 준 것이다. 어떤 정렬 알고리즘을 사용하면 이와 같은 데이터의 자리 교환이 일어나겠는가? (단, 제일 위의 행이 주어진 데이터이고, 아래로 내려갈수록 정렬이 진행되는 것이다.)

<보기>

초기 데이터	8 9 4 3 7 1 5 2
	8 9 3 4 1 7 2 5
	3 4 8 9 1 2 5 7
정렬 데이터	1 2 3 4 5 7 8 9

① 삽입 정렬
② 선택 정렬
③ 합병 정렬
④ 퀵 정렬
정답 체크 :
(3) 합병 정렬

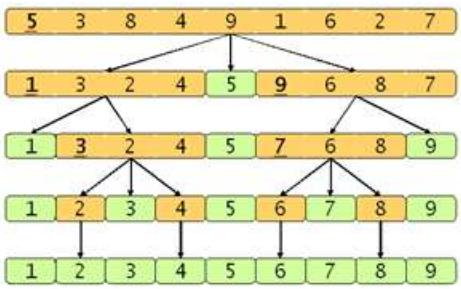


오답 체크 :

(1) 삽입 정렬

(2) 선택 정렬

(4) 퀵 정렬



16. <보기>의 각 설명과 일치하는 데이터 구조로 바르게 짝 지어진 것은?

<보기>
㈎ 먼저 추가된 항목이 먼저 제거된다.
㈏ 먼저 추가된 항목이 나중에 제거된다.
㈐ 항목이 추가된 순서에 상관없이 제거된다.

	㈎	㈏	㈐
①	큐	연결 리스트	스 택
②	스 택	연결 리스트	큐
③	스 택	큐	연결 리스트
④	큐	스 택	연결 리스트

정답 체크 :
(4)
(ㄱ) 큐 : FIFO의 특성을 가진다.
(ㄴ) 스택 : LIFO의 특성을 가진다.
(ㄷ) 연결리스트 : 임의의 위치에 삽입 및 제거할 수 있다.

17. 전화번호의 마지막 네 자리를 3으로 나눈 나머지를 해싱(hashing)하여 데이터베이스에 저장하고자 한다. 나머지 셋과 다른 저장 장소에 저장되는 것은?
① 010-4021-6718
② 010-9615-4815
③ 010-7290-6027
④ 010-2851-5232
정답 체크 :
(1) 6718 : 나머지 1
오답 체크 :
(2) 4815 : 나머지 0
(3) 6027 : 나머지 0
(4) 5232 : 나머지 0

18. 다음 메모리 영역 중 전역 변수가 저장되는 영역은?
① 데이터(Data)
② 스택(Stack)
③ 텍스트(Text)
④ 힙(Heap)
정답 체크 :
(1) 데이터 : 전역 변수가 저장된다.
오답 체크 :
(2) 스택 : 지역 변수, 복귀 주소 등이 저장된다.
(3) 텍스트 : 실행 파일(코드)가 저장된다.
(4) 힙 : 동적 메모리가 저장된다.
Tip! : 변수와 메모리 영역 사이의 관계를 그림으로 나타내면 다음과 같다.

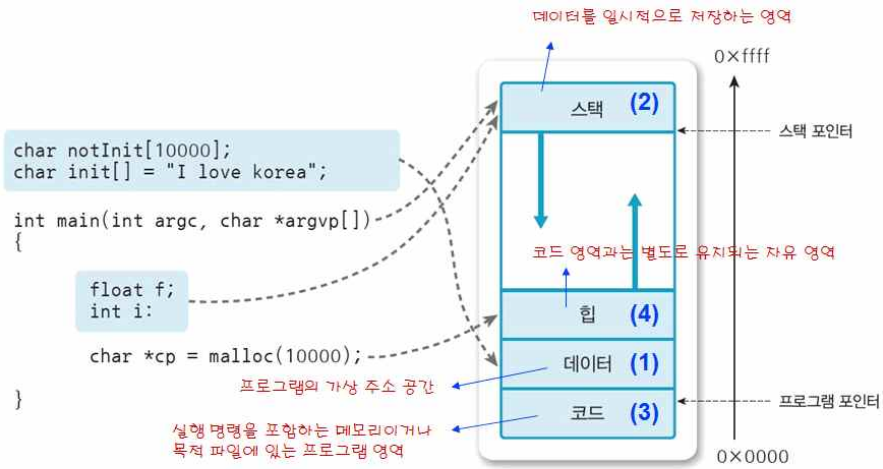


그림 3-2 프로세스의 일반적인 메모리 구조(사용자 관점의 프로세스)

19. UML(Unified Modeling Language)에 대한 설명으로 가장 옳지 않은 것은?

① UML은 방법론으로, 단계별로 어떻게 작업해야 하는지 자세하게 나타낸다.

② UML은 소프트웨어의 구성요소와 그것들의 관계 및 상호작용을 시각화한 것이다.

③ UML은 객체지향 소프트웨어를 모델링하는 표준 그래픽 언어로, 심벌과 그림을 사용해 객체지향 개념을 나타낼 수 있다.

④ UML은 소프트웨어 개발의 중요한 작업인 분석, 설계, 구현의 정확하고 완벽한 모델을 제공한다.

정답 체크 :

(1) UML은 방법론을 적용할 때의 결과물을 나타내기 위한 도구이고 객체지향 소프트웨어를 모델링하는 표준 그래픽 언어이지 방법론이 아니다. 소프트웨어 공학에서 프로세스는 단계적인 작업의 틀을 정의한 것(what)이고, 방법론은 v프로세스의 구체적인 구현(how)을 의미한다.

오답 체크 :

(2) 소프트웨어 집약 시스템의 시각적 모델을 만들기 위한 도안 표기법을 포함한다.

(3) 객체 지향 소프트웨어 집약 시스템을 개발할 때 산출물을 명세화, 시각화, 문서화할 때 사용한다.

(4) 프로세스(분석, 설계, 구현 등)의 방법론을 적용할 때의 결과물을 나타내기 위한 도구이다.

20. <보기>의 C 프로그램을 실행했을 때, 화면에 출력되는 값은? (단, 프로그램의 첫 번째 열의 숫자는 행 번호이고 프로그램의 일부는 아님.)

<보기>

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define N 3
4  int main(void) {
5     int (*in)[N], *out, sum=0;
6
7     in=(int (*)[N])malloc(N * N * sizeof(int));
```

```
8     out=(int *)in;
9
10    for (int i=0; i < N * N; i++)
11       out[i]=i;
12
13    for (int i=0; i < N; i++)
14       sum += in[i][i];
15
16    printf( " %d " , sum);
17       return 0;
18 }
```

① 0
② 3
③ 6
④ 12

정답 체크 :

(4)

int (*in)[N]; // 포인터 배열 = 이중 배열

in=(int (*)[N])malloc(N*N*sizeof(int)); // 이중 배열 메모리 할당

out=(int *)in; // out(단일 배열)과 in(이중 배열)은 운명을 같이 한다

out[i]=i; // out[0] = 0, out[1] = 1, …, out[8] = 8

sum += in[i][i]; // sum = in[0][0] + in[1][1] + in[2][2] = 0 + 4 + 8 = 12