2005학년도 중등교사 신규임용 후보자 선정 경쟁시험

수험번호 : ( )　　성　명 : ( )

○　　○　　○　　○

(2)교시 **정보 · 컴퓨터**	(25)문항 (80)점	작성시간 (150)분	감독관 확인	인

○ 문답지 전체 면수가 맞는지 확인하시오.

○ 문답지 모든 면의 상단 해당란에 수험번호와 성명을 쓰시오.

○ 답안을 수정할 때에는 두 줄(〓)을 긋고 고친 내용을 쓰시오.

○ 다음의 경우에는 문답지 전체나 해당 문항의 답안을 채점하지 않습니다.
- 문답지 상단의 점선 아래에 개인 정보가 노출된 문답지 전체
- 연필로 작성된 문항
- 수정 테이프나 수정액을 사용하여 수정된 문항

※ 각 문항에 대한 답은 문답지의 답란에 쓰되, 주어진 조건에 맞게 쓰시오.

1. A 중학교 특기적성 교육활동 시간에 1학년은 워드프로세서 '다단나누기'를 이용하여 가족신문을 완성하였고, 2학년은 비주얼 베이직의 명령문 중 반복문을 적용하여 1 100까지 합을 구하고 그 결과를 출력하는 프로그램을 만들었다. A 중학교의 컴퓨터 교육은 타일러(Taylor)의 3가지 컴퓨터 교육의 역할 중 어느 유형에 속하는지 쓰고, 각각의 유형에 대하여 1줄 이내로 설명하시오. [3점]

- 중학교 1학년 컴퓨터교육 :

  설명 :

- 중학교 2학년 컴퓨터교육 :

  설명 :

[2 4] 다음은 <교사의 일기>를 통해 본 K 고등학교 1학년 '정보 사회와 컴퓨터' 수업의 한 장면이다. 다음 물음에 답하시오.

<교사의 일기>

1교시, 1학년 학생들이 지난 한달간 정보윤리에 대해 수행한 프로젝트 결과를 발표하는 시간이다. 원활한 과제수행을 위해 35명의 학생들을 5명씩 7모둠으로 나누었고, 각 모둠에는 컴퓨터 활용능력이 높은 학생 1명과 낮은 학생 2명이 포함되었다. ① 모둠의 장은 컴퓨터 활용능력이 높은 학생으로, 컴퓨터 활용능력이 낮은 학생을 도와주고 가르치는 도우미 역할도 담당하였다. 오늘은 불법복제에 대해 15분간 강의한 후, 같은 주제로 프로젝트를 수행한 [모둠2]가 발표하였다. [모둠2]는 '불법복제, 이제 그만'이라는 포스터와 사진자료를 칠판에 붙이고, 실물화상기를 이용해 불법 복사된 CD를 보여주었다. 파워포인트를 이용한 불법복제 사례에 대한 발표가 있었고, 학생들은 미리 배포된 인쇄 자료를 참고하였다. 정보통신윤리위원회에서 제공한 짧은 애니메이션도 상영하였다. ② [모둠2]의 5명은 전문가로서 6명씩 나누어진 소그룹의 리더가 되어 '불법복제방지'에 대해 토론한 후 전체 발표를 마무리하였다. [모둠2]의 발표 후 학생들은 '모둠 발표 체크리스트'로 모둠의 협력 정도, 발표 태도 등을 평가하였고, 나는 '과제평가 체크리스트'로 수행 결과를 평가하였다. [모둠2]는 성찰보고서를 평가물로 제출하였다.

2. 위 사례에 적용된 교수 · 학습방법, 교재 및 매체 그리고 평가방법을 아래 빈칸의 수만큼 채우시오. [4점]

- 교수 · 학습방법 : 동료 교수법, 토론법, , ,
- 교재 및 매체 : 사진, 포스터, , , , ,
- 평가방법 : , ,

3. 밑줄 친 ①에 적용된 교수 · 학습방법의 기반이 되는 비고츠키(Vygotsky)의 이론을 쓰고, 이에 대해 3줄 이내로 설명하시오. [3점]

- 이론 :
- 설명 :

수험번호 : (　　)　　성　명 : (　　)

○　　○　　○　　○

---

4. 위 <교사의 일기>에서 밑줄 친 ②에 적용된 소그룹 토론법의 유형을 쓰시오. [2점]

• 유형 :

5. 다음은 교수설계시 고려해야할 8가지 시사점이다. 시사점을 행동주의, 인지주의, 구성주의 관점에서 분석하여 각 이론에 해당하는 번호를 모두 나열하시오. [4점]

번호	교수설계　시사점
①	학습결과는 관찰될 수 있고 측정될 수 있도록 설계해야 한다.
②	지식의 구조를 명료화하고 다양한 인지전략을 적용하도록 설계해야 한다.
③	성공적인 학습을 위해 피드백과 긍정적 강화를 제공하도록 설계해야 한다.
④	학습내용을 선수학습과 연계하여 관련 있는 사건으로 구성하고, 학습의 위계를 제시하도록 설계해야 한다.
⑤	자기 성찰을 통한 학습의 기회를 제공하도록 설계해야 한다.
⑥	추상적인 원리와 법칙보다는 실제적인 상황 혹은 문제를 제시하도록 설계해야 한다.
⑦	학습결과 확인과 파지를 높이기 위한 반복학습과 연습의 기회를 포함하도록 설계해야 한다.
⑧	능동적인 학습을 위한 자원 혹은 도구를 갖춘 학습 환경을 조성하도록 설계해야 한다.

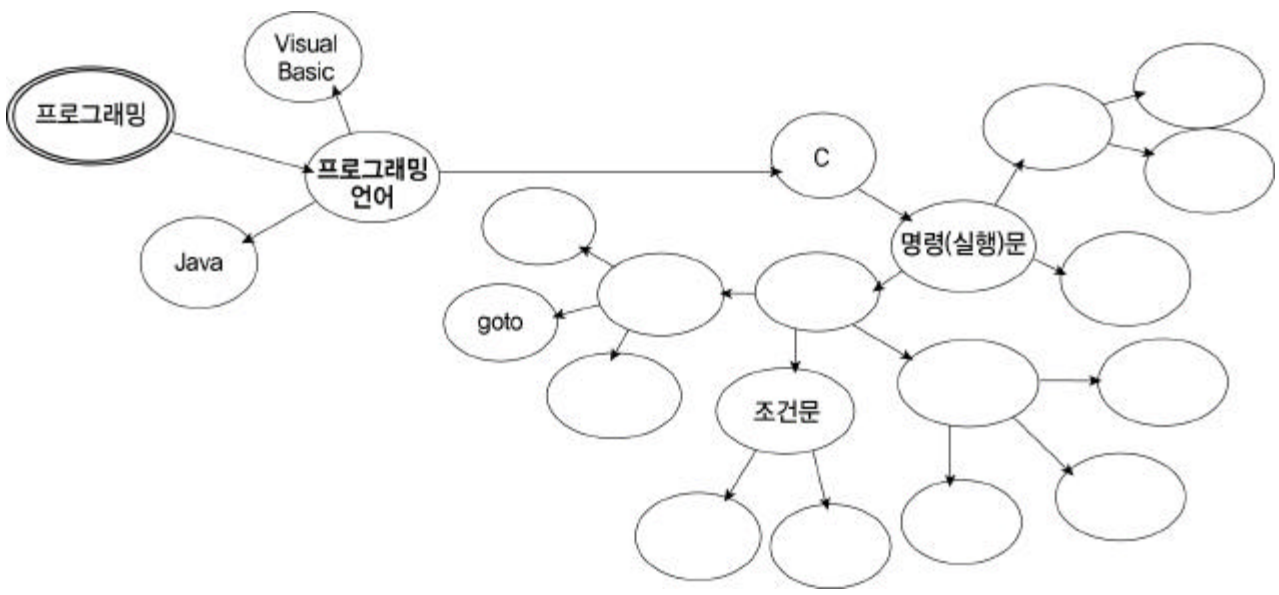
• 행동주의 :　　• 구성주의 :

• 인지주의 :

6. A 교사는 '프로그래밍' 수업의 도입단계에서 명령어 종류에 대한 개념을 지식의 구조로 소개하려 한다. '개념 지도'라고도 불리는 이 방법은 개념과 다른 개념들 간의 상위 또는 하위 수준의 개념들을 도식화해주고 그 관계를 보여준다. <보기>의 항목들을 이용하여 제시된 지식의 구조를 완성하시오. [3점]

<보　기>

if, switch, for, while, printf, break, 반복문, do　while, 배정(치환)문, continue, 입출력문, 제어문, 분기문, scanf

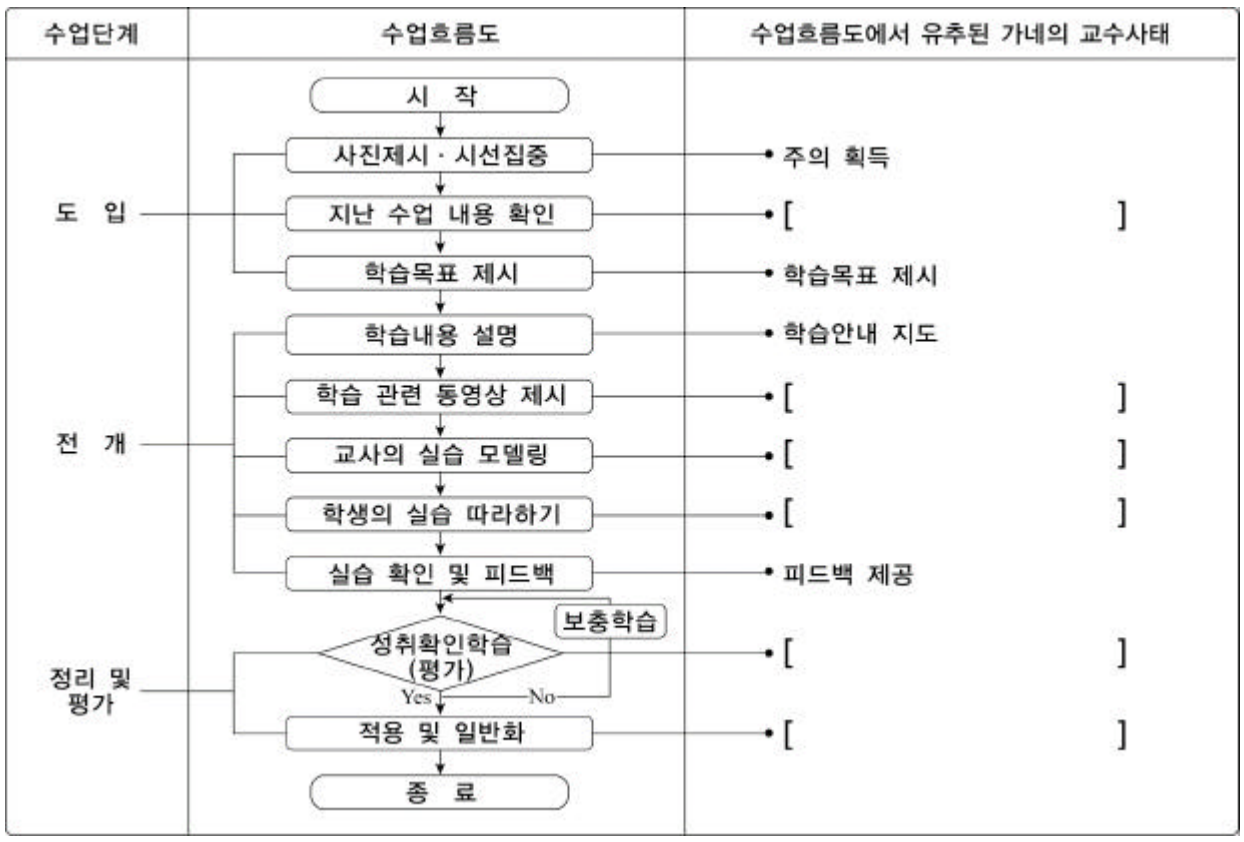


수험번호 : ( ) 성 명 : ( )

○ ○ ○ ○

---

7. K 교사는 수업지도안의 수업단계와 수업흐름도를 가네(Gagné)의 9가지 교수사태(instructional events)와 연계하려 한다. 아래 그림에서 수업흐름도와 관련지어진 교수사태를 보고, 수업흐름도의 나머지 항목에 해당하는 교수사태를 유추하여 쓰시오. 단, 유추된 가네의 교수사태는 순차적이지 않아도 되며, 수업흐름도의 단계에 따라 반복적으로 유추될 수 있다. [6점]

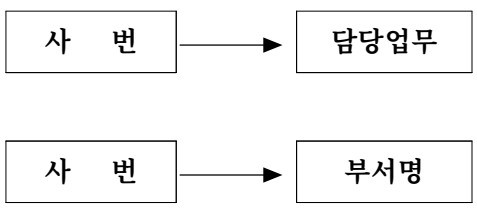


8. 아래 <업무> 릴레이션을 상위 정규형으로 분해하였을 때 생성되는 릴레이션의 다이어그램이 다음과 같다고 가정하자. 이와 같은 방법으로 정규화하였을 때 발생되는 문제점을 1줄로 쓰시오. 그리고 그 문제점이 해결되도록 정규화하였을 때 생성되는 릴레이션의 다이어그램을 그리시오. [3점]

<업 무>

사번	담당업무	부서명
10	A 100	기획부
20	E 200	자재부
30	D 300	운송부
40	B 400	기획부
50	A 100	기획부

<다이어그램>



• 문제점 :

• 다이어그램 :

수험번호 : (　　) 성　명 : (　　)

○　○　○　○

---

9. 시스템에 장애(failure)가 발생한 경우 검사시점(checkpoint) 기법을 적용하여 Redo와 Undo를 수행할 트랜잭션을 결정하고 회복(recovery)을 수행할 수 있다. 주어진 <가정>을 적용하여 트랜잭션이 아래 표와 같이 수행될 경우 Redo 연산을 수행할 모든 트랜잭션을 쓰고, 각 트랜잭션에서 Redo 연산을 수행할 범위의 시작 시각과 종료 시각을 쓰시오. [3점]

<가 정>

(가) 검사시점 시각은 11시 30분 30초이다.
(나) 검사시점 간격은 2분으로 설정하였다.
(다) 시스템의 장애가 11시 30분 50초에 발생하였다.

<트랜잭션 수행 시간>

트랜잭션	시작 시각(시:분:초)	종료 시각(시:분:초)
T1	11:30:20	11:30:40
T2	11:30:10	11:30:20
T3	11:30:40	11:30:45
T4	11:30:45	종료되지 않음
T5	11:30:20	종료되지 않음

• Redo 연산을 수행할 트랜잭션

트랜잭션	시작 시각	종료 시각

[10～11] 다음은 다단계 피드백 큐(MFQ) 스케쥴링을 위한 <가정>이다. 물음에 답하시오.

<가 정>

(가) 대기큐의 번호 i는 0부터 시작되고, 최상위 대기큐의 번호는 0이다.
(나) 각 대기큐 i에 대한 시간 할당량은 2^i이다.
(다) 새로 생성되는 프로세스는 최상위 대기큐에 들어온다.
(라) 프로세스가 수행 중에 더 높은 우선순위의 프로세스가 들어올 경우, 현재 남아있는 시간 할당량을 모두 사용하고 CPU를 반환한다.
(마) 시스템에 1개의 프로세스만 존재할 경우 시간 할당량을 모두 사용한 후 하위단계의 대기큐로 이동한다.
(바) 최상위 대기큐를 제외한 각 대기큐는 프로세스들의 대기시간에 대한 임계치로 2^{2i}을 가지며, 프로세스의 대기시간이 이 임계치보다 크면 프로세스는 상위단계의 큐로 이동한다. 대기큐에서 프로세스가 CPU를 할당받을 순서가 되더라도 그 시점에서 대기시간이 임계치보다 크면 상위단계 큐로 이동한다.

10. 다음 표에 주어진 프로세스들이 위의 <가정> 중 (바)를 제외한 나머지 조건에 따라 수행되었을 경우, 각 프로세스가 대기하였던 큐의 수를 구하시오. [2점]

프로세스	도착시간	실행시간
P_1	0	5
P_2	2	3
P_3	4	5
P_4	5	4
P_5	7	3

• 대기하였던 큐의 수

프로세스	P_1	P_2	P_3	P_4	P_5
대기큐의 수					

수험번호 : ( ) 성 명 : ( )

○ ○ ○ ○

---

11. 다단계 피드백 큐 스케쥴링에서 에이징(aging) 기법이 적용되는 이유를 1줄 이내로 쓰고, 문항 10에서 주어진 프로세스들이 위의 <가정>에 따라 수행되었을 때 각 프로세스의 반환 시간과 대기 시간을 각각 구하시오. 단, 각 프로세스에 주어진 실행 시간의 1단위는 10ms이다. [4점]

• 이유 :

• 프로세스의 반환 시간과 대기 시간

프로세스	P_1	P_2	P_3	P_4	P_5
반환 시간(ms)					
대기 시간(ms)					

[12～13] 다음은 가상메모리 관리기법 중, 페이징 시스템에 관한 <가정>이다. 물음에 답하시오.

<가 정>

(가) 시스템의 주소는 워드 단위로 지정되고, 1개의 정수는 1개의 워드에 저장된다.
(나) 페이지 크기는 100워드이고, 페이지 교체기법은 LRU(Least Recently Used) 기법을 사용한다.
(다) 프로그램 코드는 0번 페이지에 모두 저장되고, 정수형 2차원 배열 a[20][20]와 b[20][20]는 각각 가상주소 100번지와 500번지부터 저장된다.
(라) 프로그램에서 2차원 배열을 제외한 나머지 변수는 0번 페이지에 저장된다.
(마) 프로세스에게 3개의 페이지 프레임이 할당된다.

12. 다음 <프로그램 2>는 <프로그램 1>이 수행될 때 페이지 부재율이 감소되도록 프로그램 코드를 변경하기 위한 것이다. <프로그램 2>의 ①, ②, ③을 채우시오. 단, 두 프로그램에서 반복문의 수행 횟수와 수행 결과는 동일해야 한다. [2점]

<프로그램 1>

```
main()
{
    int i, j, a[20][20], b[20][20];
    for(i=0; i<20; i++)
      for(j=0; j<20; j++){
          a[j][i]=0;
          b[i][j]=a[j][i]+1; }
}
```

<프로그램 2>

```
main()
{
   int i, j, a[20][20], b[20][20];
       ①
       ②  {
          a[j][i]=0;
       ③ ; }
 }
```

①: ②: ③:

13. 문항 12의 <프로그램 1>이 수행될 때 발생하는 페이지 부재수의 합(총 페이지부재수)을 쓰고, 그 중 3번째 발생하는 페이지 부재에서 주기억장치로 적재되는 배열의 인덱스를 쓰시오. 단, 프로그램 코드가 처음으로 주기억장치에 적재될 때는 페이지 부재수로 계산하지 않으며, 배열의 인덱스는 배열의 이름을 포함하여 처음과 마지막 인덱스만 쓰시오. [3점]

• 페이지 부재수의 합 : • 배열의 인덱스 :

수험번호 : ( ) 성 명 : ( )

○ ○ ○ ○

[14 15] 인터넷에서 IP 데이터그램(datagram)을 전송할 때 분할(fragmentation)과 재결합이 이루어진다. 다음 IPv4 데이터그램의 포멧을 참조하여 분할에 관한 물음에 답하시오.

14. 다음 조건으로 IP 데이터그램을 분할할 때, 분할된 마지막 데이터그램에 대하여 ①, ②, ③에 들어갈 각각의 내용을 10진수로 쓰시오. 단, ②번 필드에서 현재 사용되지 않는 예약비트(reserved bit)의 값은 0으로 가정한다. [4점]

<조 건>

(가) 데이터와 프로토콜 오버헤드를 포함하여 네트워크 계층의 최대 전송 단위(MTU: Maximum Transmission Unit)는 200바이트이다.
(나) IP 데이터그램의 헤더 길이는 20바이트이며, 옵션 필드는 없다.
(다) 패킷은 MTU 범위내에서 가능한 최대로 크게 분할한다.
(라) 보내고자 하는 전송계층의 데이터 길이(IP 패킷의 데이터 필드 길이)는 700바이트이다.

①: 바이트 ②: ③:

15. IP 데이터그램의 헤더 길이가 20바이트이고 옵션 필드가 없다고 가정할 때, 1개의 IP 데이터그램에 대한 최대 분할의 수를 쓰시오. [2점]

• 최대 분할의 수 :

16. 흐름제어의 슬라이딩 윈도우 방식에서 프레임들은 0부터 임의의 수까지 순차번호가 매겨져 있다. 슬라이딩 윈도우 방식의 하나인 Go-Back-N ARQ에서 정지-대기(stop and wait) 현상을 피하기 위해서는 충분히 큰 윈도우가 필요하다. 통신링크에서 필요한 윈도우의 크기가 50인 경우에 프레임의 순차번호를 나타내기 위해 필요한 최소 비트수를 쓰시오. 그리고 통신링크가 요구하는 윈도우 크기가 k이고 프레임의 순차번호를 나타내기 위한 최소 비트수를 n이라고 할 때, n과 k의 관계를 식으로 나타내시오. [4점]

• 비트수 : 비트

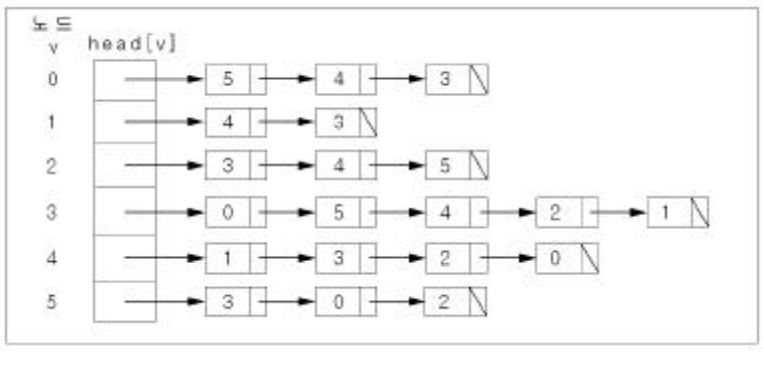
• 관계식 :

수험번호 : ( ) 성 명 : ( )

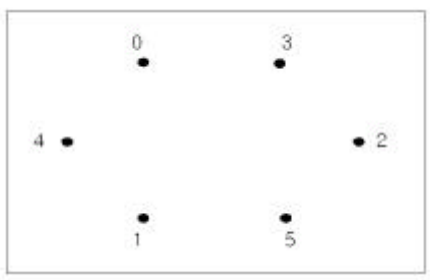
○ ○ ○ ○

[17 18] 그래프의 노드 구조체가 다음과 같다. 주어진 리스트는 무방향 그래프의 인접관계를 표현했고 head[v]는 노드 v의 포인터 배열이다. 다음 물음에 답하시오.

```
struct node {
    int NODE ;
    struct node * LINK ;
}
```



17. 위의 인접리스트를 이용하여 그래프를 완성하시오. 단, 점0 점5는 노드 번호이다. [2점]



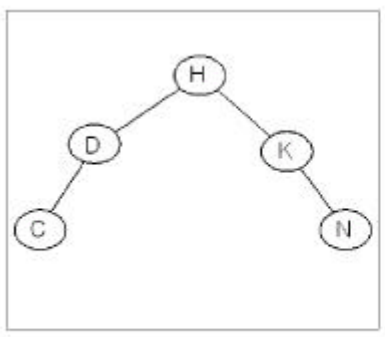
18. 다음 알고리즘과 주어진 인접리스트 정보를 이용하여 그래프의 노드를 순회(traverse)하고자 한다. DFS(int v)의 출발노드 v를 0부터 시작할 때 출력 결과를 쓰시오. 단, DFS() 알고리즘을 수행하기 전에 visited[ ]의 초기 값은 모두 0이고, 배열 첨자는 0부터 시작한다. [4점]

```
DFS(int v) {
    struct node * w ;
    visited[v] = True ;
    printf("%d", v) ;
    for(w=head[v]; w; w=w->LINK)
        if(!visited[w->NODE])
            DFS(w->NODE) ;
}
```

• 출력결과

출력순서	출력노드
1	
2	
3	
4	
5	
6	

19. [그림 1]은 5개의 자료를 갖는 균형이진(AVL) 트리이다. [그림 1]에 자료 'A'를 삽입 한 후의 트리를 [그림 2]에 나타내고, 균형 인수(balanced factor) 값을 쓰시오. 또한 [그림 2]의 트리를 균형이진 트리로 변경하여 [그림 3]에 나타내시오. [3점]



[그림 1] [그림 2] [그림 3]

수험번호 : (　　) 성　명 : (　　)

○　　○　　○　　○

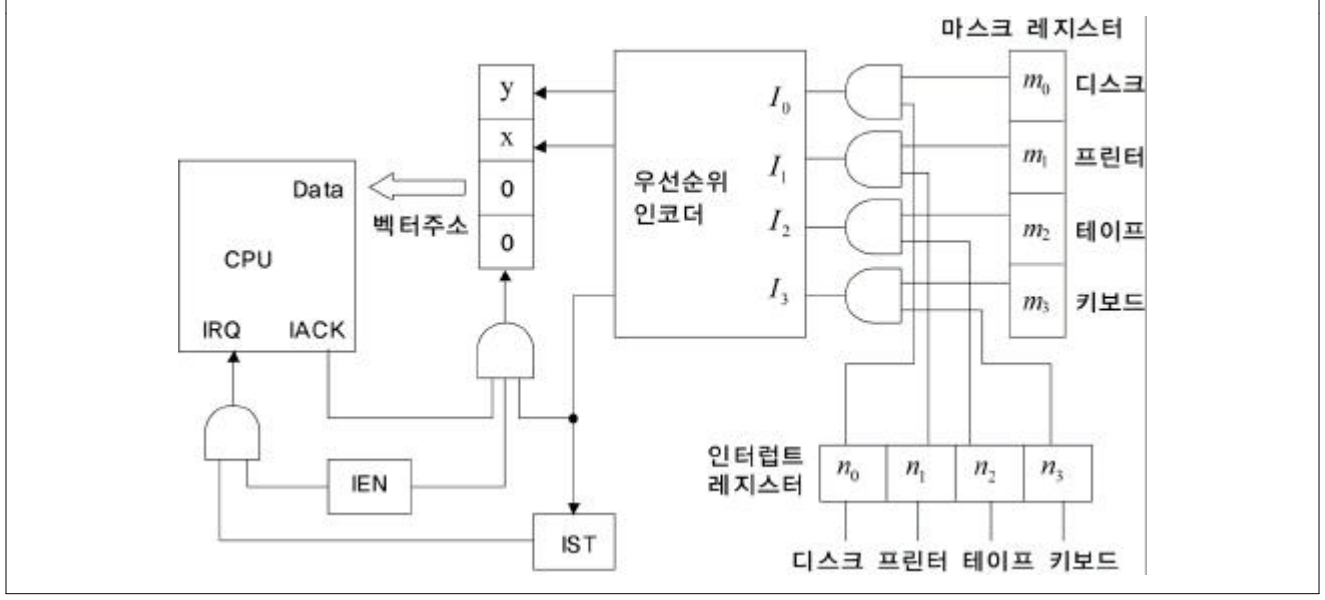
---

20. 다음 <조건>을 만족하는 병렬 우선순위 인터럽트 시스템이 그림과 같은 하드웨어로 구성되어 있다고 가정하고 우선순위 인코더(encoder)의 회로도를 설계하고자 한다. 우선순위 인코더의 출력 x와 y에 대한 카르노맵(Karnaugh Map)을 작성하고, 인터럽트 상태 플립플롭 IST의 부울 함수식(boolean function)을 쓰시오. [5점]

<조 건>

(가) 인터럽트 요청장치는 우선순위가 높은 순서로 디스크, 프린터, 테이프, 키보드가 있으며, 각각의 장치번호는 0, 1, 2, 3이다.
(나) 각 장치의 인터럽트 요청 상태는 비트 단위로 인터럽트 레지스터에 나타난다.
(다) 시스템은 각 장치에 대한 인터럽트 요청의 상태를 조절할 수 있는 마스크 레지스터를 가지고 있다. 장치가 인터럽트를 요청하면 그 장치보다 우선순위가 낮은 장치의 마스크 레지스터 비트 값이 0으로 된다.
(라) 우선순위 인코더는 인터럽트 벡터주소의 하위 2비트(x, y)로 사용될 장치번호를 출력한다.
(마) 마스크 되지 않은 장치가 인터럽트를 발생시키면 인터럽트 상태 플립플롭 IST의 값이 1로 된다.
(바) 인터럽트 요청 가능 상태를 나타내는 플립플롭 IEN이 있으며, 값이 0이면 인터럽트 금지 상태이다.

<병렬 우선순위 인터럽트 하드웨어>



• x에 대한 카르노맵(Karnaugh Map) :

I_0I_1 \ I_2I_3				

• y에 대한 카르노맵(Karnaugh Map) :

I_0I_1 \ I_2I_3				

• IST의 부울 함수식 :

수험번호 : (　　) 성　명 : (　　)

○　○　○　○

[21　22] 3단계 명령어 파이프라인에서 단계별 작업 내용과 수행할 명령어 코드가 다음과 같다. 물음에 답하시오. 단, 명령어(문장) 앞의 숫자는 문장번호를 의미하며, 명령어를 처리하기 위한 파이프라인의 각 단계는 1클럭 사이클이 소요된다고 가정한다.

단계	작업 내용
1단계	명령어 패치(fetch)
2단계	명령어 디코딩(decoding) 레지스터의 데이터 패치
3단계	연산 및 연산결과 저장

명령어 코드	각 명령어의 의미
1 : LOAD　R5, 100	R5 ← 100
2 : ADD　R1, R1, R3	R1 ← R1+R3
3 : ADD　R2, R2, R3	R2 ← R2+R3
4 : DEC　R5	R5 ← R5-1
5 : IF (R5 ≠ 0) THEN GOTO 2	

21. 위의 명령어 코드를 파이프라인으로 수행하면 분기 문제가 발생하며, 명령어를 재배열하여 분기 문제를 해결할 수 있다. 아래 그림은 재배열된 명령어 코드와 파이프라인에서 명령어가 처리되는 과정을 문장번호로 나타낸 것이다. 재배열된 명령어 코드를 실행하면 분기 문제는 해결되지만, 파이프라인에서 새로운 문제가 발생된다. 명령어 코드를 재배열한 결과 파이프라인에서 발생되는 문제가 무엇인지 쓰고, <명령어의 처리과정> 그림에 제시된 4　9까지의 클럭 사이클에서 이 문제가 발생된 클럭 사이클의 번호를 모두 나열하시오. [3점]

<재배열된 명령어 코드>

```
1: LOAD  R5, 100
4: DEC   R5
5: IF (R5 ≠ 0) THEN GOTO 4
2: ADD   R1, R1, R3
3: ADD   R2, R2, R3
```

클럭 사이클

	1	2	3	4	5	6	7	8	9	...
1단계	1	4	5	2	3	4	5	2	3	...
2단계		1	4	5	2	3	4	5	2	...
3단계			1	4	5	2	3	4	5	...

<명령어의 처리과정>

• 문제점 :

• 클럭 사이클

22. 다음과 같이 명령어 코드의 일부를 수정하고 4개의 명령어를 재배열하면, 문항 21에서 발생되는 문제를 해결할 수 있다. 이를 위해 ①, ②, ③, ④에 들어갈 명령어 코드의 문장번호를 순서대로 나열하시오. 단, 명령어의 위치가 바뀌어도 동일한 결과가 나올 때는 문장번호가 작은 것이 앞에 오도록 한다. [2점]

<수정한 명령어 코드>

```
1: LOAD R5, 99
      ①
      ②
      ③
      ④
6: no-op
7: INC R5
```

<재배열할 명령어 코드>

```
2 : ADD   R1, R1, R3
3 : ADD   R2, R2, R3
4 : DEC   R5
5 : IF (R5 ≠ 0) THEN GOTO 2
```

①의 문장번호 :　　②의 문장번호 :　　③의 문장번호 :　　④의 문장번호 :

수험번호 : ( ) 성 명 : ( )

○ ○ ○ ○

[23 24] 다음은 입력받은 자료를 변환하여 출력하는 프로그램이다. 단, main()에서 표준 입력 값은 4자리 수 이하의 양의 정수로 가정한다. 물음에 답하시오.

```
#include <stdio.h>
#define SIZE 9
void reverse(int number);
int ary[SIZE];

void main() {
        int input, i;
        scanf("%d", &input);
        reverse(input);
        for(i=0; i<SIZE && ary[i]; i++) {
                printf("%d", ary[i]);
        }
}
```

```
void reverse(int number) {
        int i=0, r_digit;
        while(number) {
                r_digit=number%10;
                ary[i++]=r_digit;
                number=_____①_____
        }
}
```

23. 주어진 reverse()는 입력된 정수를 자리수 단위로 분할하여 역순으로 배열에 저장한다. 이와 같은 reverse() 기능이 수행될 수 있도록 ①을 채워 프로그램을 완성하시오. [3점]

①:

24. 위 프로그램에서 1024가 입력되었을 때 출력 결과를 쓰시오. [2점]

• 출력 결과 :

25. <문법 1>은 모호한 문법이다. <문법 1>이 문장 'A=C-3*D'에 대해 모호한 문법임을 보여주는 파스트리를 나타내고, 모호한 이유를 파스트리에 근거하여 설명하시오. 그리고 <문법 2>는 <문법 1>의 모호성을 제거한 것이다. <문법 2>의 ①을 채우시오. [4점]

<문법 1>

```
<assign> ::= <id> = <exp>
<exp>    ::= <exp> - <exp> | <exp> * <exp> | (<exp>) | <id> | <digit>
<id>     ::= A | B | C | D | E
<digit>  ::= 1 | 2 | 3 | 4 | 5
```

<문법 2>

```
<assign> ::= <id> = <exp>
<exp>    ::= <exp> - <term> | <term>
                  ①
<fac>    ::= (<exp>) | <id> | <digit>
<id>     ::= A | B | C | D | E
<digit>  ::= 1 | 2 | 3 | 4 | 5
```

• 파스트리

모호한 이유 :

①:

- 수고하셨습니다 -
