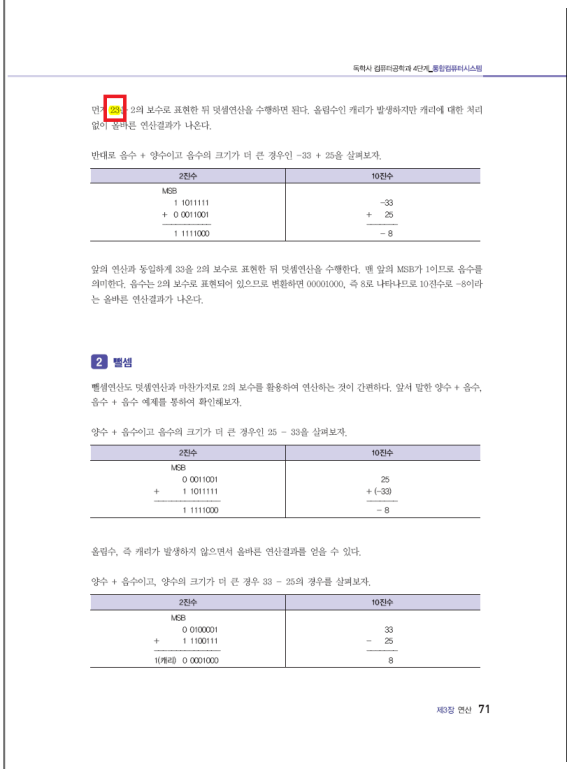
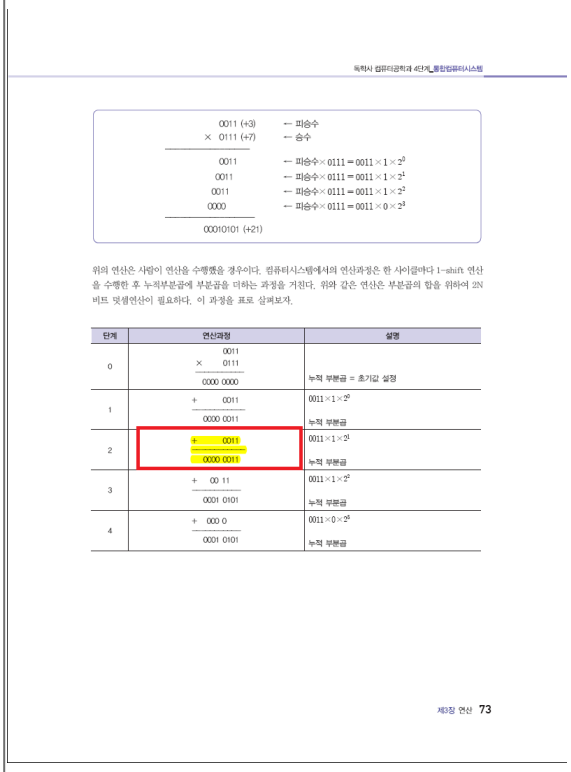
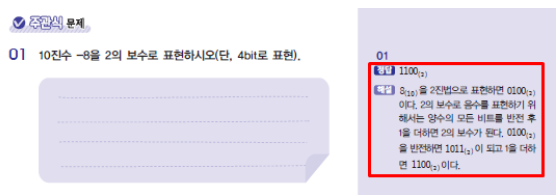
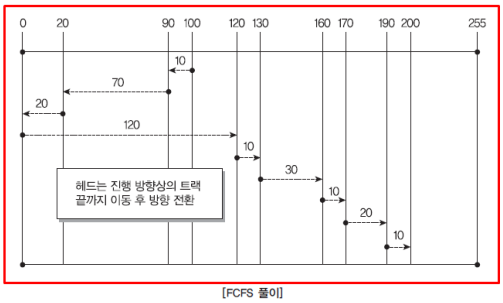
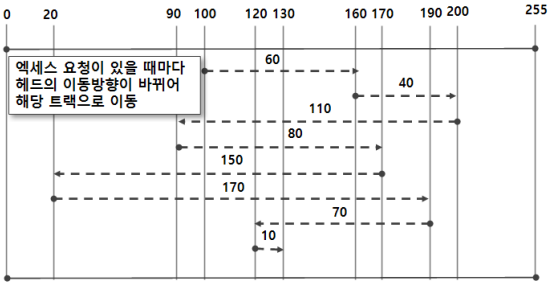
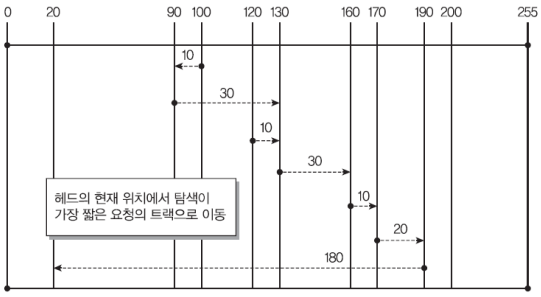
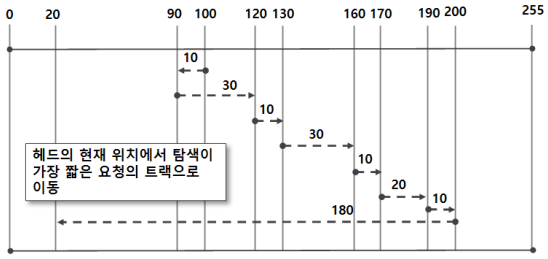
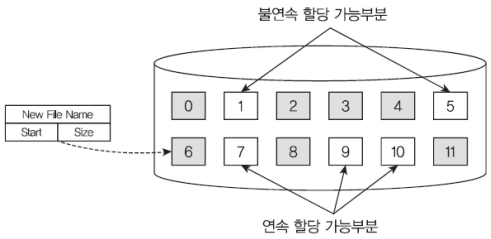
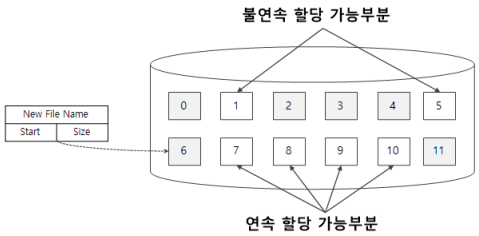
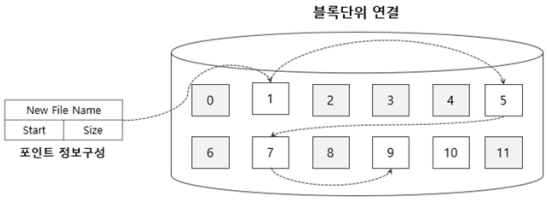
위치	오류유형	수정 전	수정 후
71p	문제-본문	 독학사 컴퓨터공학과 4단계_통합컴퓨터시스템 먼저 33을 2의 보수로 표현한 뒤 덧셈연산을 수행하면 된다. 올림수인 캐리가 발생하지만 캐리에 대한 처리 없이 올바른 연산결과가 나온다. 반대로 음수 + 양수이고 음수의 크기가 더 큰 경우인 −33 + 25을 살펴보자. 2진수: MSB 1 1011111 + 0 0011001 = 1 1111000 10진수: −33 + 25 = −8 앞의 연산과 동일하게 33을 2의 보수로 표현한 뒤 덧셈연산을 수행한다. 단 앞의 MSB가 1이므로 음수를 의미한다. 음수는 2의 보수로 표현되어 있으므로 변환하면 00001000, 즉 8로 나타나므로 10진수로 −8이라는 올바른 연산결과가 나온다. 2 뺄셈 뺄셈연산도 덧셈연산과 마찬가지로 2의 보수를 활용하여 연산하는 것이 간편하다. 앞서 말한 양수 + 음수, 음수 + 음수 예제를 통하여 확인해보자. 양수 + 음수이고 음수의 크기가 더 큰 경우인 25 − 33을 살펴보자. 2진수: MSB 0 0011001 + 1 1011111 = 1 1111000 10진수: 25 + (−33) = −8 올림수, 즉 캐리가 발생하지 않으면서 올바른 연산결과를 얻을 수 있다. 양수 + 음수이고, 양수의 크기가 더 큰 경우 33 − 25의 경우를 살펴보자. 2진수: MSB 0 0100001 + 1 1100111 = 1(캐리) 0 0001000 10진수: 33 − 25 = 8 제3장 연산 71	25
		수정 사유	오타 수정
73p	문제-본문	 독학사 컴퓨터공학과 4단계_통합컴퓨터시스템 0011 (+3) ← 피승수 × 0111 (+7) ← 승수 0011 ← 피승수×0111 = 0011×1×2^0 0011 ← 피승수×0111 = 0011×1×2^1 0011 ← 피승수×0111 = 0011×1×2^2 0000 ← 피승수×0111 = 0011×0×2^3 00010101 (+21) 위의 연산은 사람이 연산을 수행했을 경우이다. 컴퓨터시스템에서의 연산과정은 한 사이클마다 1-shift 연산을 수행한 후 누적부분곱에 부분곱을 더하는 과정을 거친다. 위와 같은 연산은 부분곱의 합을 위하여 2N 비트 덧셈연산이 필요하다. 이 과정을 표로 살펴보자. 단계 / 연산과정 / 설명 0: 0011 × 0111 = 0000 0000 — 누적 부분곱 = 초기값 설정 1: + 0011 = 0000 0011 — 0011×1×2^0 — 누적 부분곱 2: + 0011 = 0000 0011 — 0011×1×2^1 — 누적 부분곱 3: + 00 11 = 0001 0101 — 0011×1×2^2 — 누적 부분곱 4: + 000 0 = 0001 0101 — 0011×0×2^3 — 누적 부분곱 제3장 연산 73	+ 0011 ---------- 0000 1001
		수정 사유	내용 수정

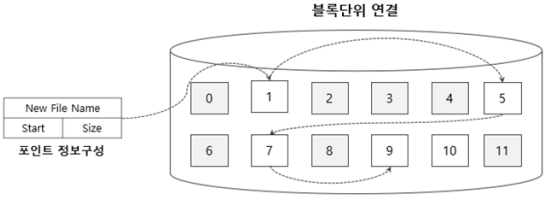
<table>
<tr><th>위치</th><th>오류유형</th><th>수정 전</th><th>수정 후</th></tr>
<tr><td rowspan="2">75p
위에서
2~12번째 줄</td><td rowspan="2">개념,공식-설명</td><td>부호 없는 수의 나눗셈을 예시로 살펴보자.
153÷8을 보면 $10011001_{(2)}$÷$0100_{(2)}$이다. 사람이 연산을 수행했을 경우 다음과 같이 연산을 진행한다.
10진수의 나눗셈과 다르지 않다.
<table>
<tr><td></td><td></td><td>0010011</td><td>몫</td></tr>
<tr><td>제수</td><td>0100</td><td>10011001</td><td>피제수</td></tr>
<tr><td></td><td></td><td>100</td><td></td></tr>
<tr><td>부분나머지</td><td></td><td>0110</td><td></td></tr>
<tr><td></td><td></td><td>0110</td><td></td></tr>
<tr><td>부분나머지</td><td></td><td>0100</td><td></td></tr>
<tr><td></td><td></td><td>0100</td><td></td></tr>
<tr><td></td><td></td><td>0001</td><td>나머지</td></tr>
</table></td><td>부호 없는 수의 나눗셈을 예시로 살펴보자.
153÷8을 보면 $10011001_{(2)}$÷$1000_{(2)}$이다. 사람이 연산을 수행했을 경우 다음과 같이 연산을 진행한다.
10진수의 나눗셈과 다르지 않다.
<table>
<tr><td></td><td></td><td>00010011</td><td>몫</td></tr>
<tr><td>제수</td><td>1000</td><td>10011001</td><td>피제수</td></tr>
<tr><td></td><td></td><td>1000</td><td></td></tr>
<tr><td>부분나머지</td><td></td><td>1100</td><td></td></tr>
<tr><td></td><td></td><td>1000</td><td></td></tr>
<tr><td>부분나머지</td><td></td><td>1001</td><td></td></tr>
<tr><td></td><td></td><td>1000</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td>나머지</td></tr>
</table></td></tr>
<tr><td>수정 사유</td><td>설명 수정</td></tr>
<tr><td rowspan="2">77p
제3편-
실제예상문제-
주관식 문제
번호 : 01</td><td rowspan="2">정답</td><td> 
</td><td>정답) $1000_{(2)}$
해설) 10진수 -8은 4bit로 표현 시 범위가 7에서 -8까지이므로, $1000_{(2)}$으로 표현한다.
<table>
<tr><th>10진수</th><th>2의 보수</th></tr>
<tr><td>-1</td><td>1111</td></tr>
<tr><td>-2</td><td>1110</td></tr>
<tr><td>-3</td><td>1101</td></tr>
<tr><td>-4</td><td>1100</td></tr>
<tr><td>-5</td><td>1011</td></tr>
<tr><td>-6</td><td>1010</td></tr>
<tr><td>-7</td><td>1001</td></tr>
<tr><td>-8</td><td>1000</td></tr>
</table></td></tr>
<tr><td>수정 사유</td><td>정답 및 해설 오류</td></tr>
<tr><td rowspan="2">219p
제8편-
실제예상문제-
해설&정답
번호 : 14</td><td rowspan="2">해설</td><td>해설 : FCFS 프로세스를 요청하는 순서대로 프로세서를 할당한다. 따라서 대기시간이 가장 긴 B가 먼저 요청한 작업이므로 B가 가장 먼저 실행된다.

정답 : ②</td><td>해설 : FCFS는 프로세스를 요청하는 순서대로 할당한다. 따라서 대기시간이 가장 짧은 A 작업이 가장 먼저 실행된다.

정답 : ①</td></tr>
<tr><td>수정 사유</td><td>정답 및 해설 오류</td></tr>
</table>

위치	오류유형	수정 전	수정 후
258p 제10편-제2장- 제2절-1. 탐색 최적화-(1) FCFS	개념,공식-설명	• Queue : 160, 200, 90, 170, 20, 190, 120, 130 • Current head starts at : 100 • 총 이동거리 : 60 + 49 + 110 + 80 + 150 + 170 + 70 + 10 = 690  [FCFS 풀이]	· 총 이동거리: 60 + 40 + 110 + 80 + 150 + 170 + 70 + 10 = 690 <하단 그림으로 변경> 
		수정 사유	오타 및 그림 교체
259p 제10편-제2장- 제2절-1. 탐색 최적화-(2) SSTF	개념,공식-설명	 	<하단 그림으로 변경> 
		수정 사유	그림 교체
277p 제10편-제4장- 제4절-1. 디스크 공간 할당 개념	개념,공식-설명	 	<하단 그림으로 변경> 
		수정 사유	그림 교체

<table>
<tr><th>위치</th><th>오류유형</th><th>수정 전</th><th>수정 후</th></tr>
<tr><td rowspan="2">278p
제10편-제4장-
제4절-3. 디스크
연속 할당 기법</td><td rowspan="2">오타</td><td>3. 디스크 연속 할당 기법
같은 파일에 속한 블록들을 체인으로 연결하기 위해 각 블록에 포인터를 두어 다음 블록 위치를 기록하여 연결하는 기법이다. 블록 단위로 구성되며, 하나의 블록은 여러 개의 섹터로 구성되고, 디렉터리는 파일의 첫 번째 블록을 가리키는 포인터를 포함한다. 여기서 블록은 데이터와 포인트를 지칭하는 용어이다.
 

[디스크 연속 할당 기법]
[디스크 연속 할당 기법 장단점]
<table><tr><th>장점</th><th>단점</th></tr><tr><td>·외부 단편화 발생을 방지한다.
·파일 생성 시 공간을 할당하지 않는다.
·파일 크기 변화에 유연하다.</td><td>·블록마다 포인터 공간이 필요하다.
·포인터 유실 시 파일도 유실된다,</td></tr></table></td><td>3. 디스크 연결 할당 기법
같은 파일에 속한 블록들을 체인으로 연결하기 위해 각 블록에 포인터를 두어 다음 블록 위치를 기록하여 연결하는 기법이다. 블록 단위로 구성되며, 하나의 블록은 여러 개의 섹터로 구성되고, 디렉터리는 파일의 첫 번째 블록을 가리키는 포인터를 포함한다. 여기서 블록은 데이터와 포인트를 지칭하는 용어이다.
 

[디스크 연결 할당 기법]
[디스크 연결 할당 기법 장단점]
<table><tr><th>장점</th><th>단점</th></tr><tr><td>·외부 단편화 발생을 방지한다.
·파일 생성 시 공간을 할당하지 않는다.
·파일 크기 변화에 유연하다.</td><td>·블록마다 포인터 공간이 필요하다.
·포인터 유실 시 파일도 유실된다,</td></tr></table></td></tr>
<tr><td>수정 사유</td><td>본문 오타</td></tr>
<tr><td rowspan="2">317p
제12편-제1장-
제2절-2. 4단계
명령어
파이프라인</td><td rowspan="2">오타</td><td>③ 파이프라인에 의한 속도 공식
파이프라인 단계 수 = k이고, 실행할 명령어들의 수 = N일 때 속도 S는 아래와 같다.
$$S_p = \frac{T_1}{T_k} = \frac{k \times N}{k \times (N-1)}$$</td><td>③ 파이프라인에 의한 속도 공식
파이프라인 단계 수 = k이고, 실행할 명령어들의 수 = N일 때 속도 S는 아래와 같다.
$$S_p = \frac{T_1}{T_k} = \frac{k \times N}{k \boxed{+} (N-1)}$$</td></tr>
<tr><td>수정 사유</td><td>본문 오타</td></tr>
<tr><td rowspan="2">350p
제13편-
실제예상문제-
해설&정답
번호 : 01</td><td rowspan="2">문제-본문</td><td>01. 임베디드 시스템은 특정 목적이 아닌 범용 목적을 갖는 시스템이다.</td><td>01. 임베디드 시스템의 특징으로는 제한적인 기능, 크기에 제약, 저전력, 프로세스와 운영체제의 다양함, 하드디스크 없음이 있다. 높은 성능은 임베디드 시스템의 특징이 아닌 슈퍼컴퓨터 등 높은 연산이 필요한 컴퓨터시스템의 특징이다.</td></tr>
<tr><td>수정 사유</td><td>해설 오류</td></tr>
</table>

도서의 오류로 학습에 불편드린 점 진심으로 사과드립니다.
더 나은 도서를 만들기 위해 노력하는 시대교육그룹이 되겠습니다.