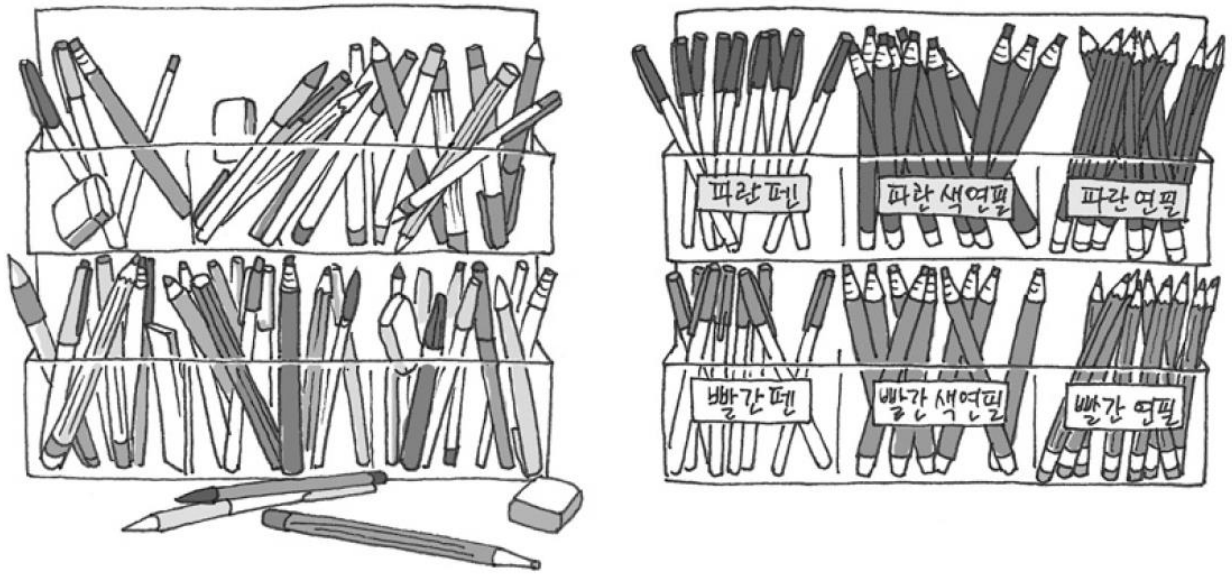
데이터구조와 알고리즘
(Data type, Linear data structure)

남춘성

1. 자료구조 개요

❖자료구조란?

- 자료를 효율적으로 사용하기 위해서 자료의 특성에 따라서 분류하여 구성하고 저장 및 처리하는 모든 작업



[나쁜 자료구조]

[좋은 자료구조]

[그림 1-1] 자료구조의 적용 예

1. 자료구조 개요

- 자료구조의 역할
  - 컴퓨터 프로그래밍에 있어서 가장 기초적인 학문분야
  - 컴퓨터 프로그램의 기본 골격
    - 프로그램이 효율적이고 안전하게 동작하게 하기 위해서 반드시 필요

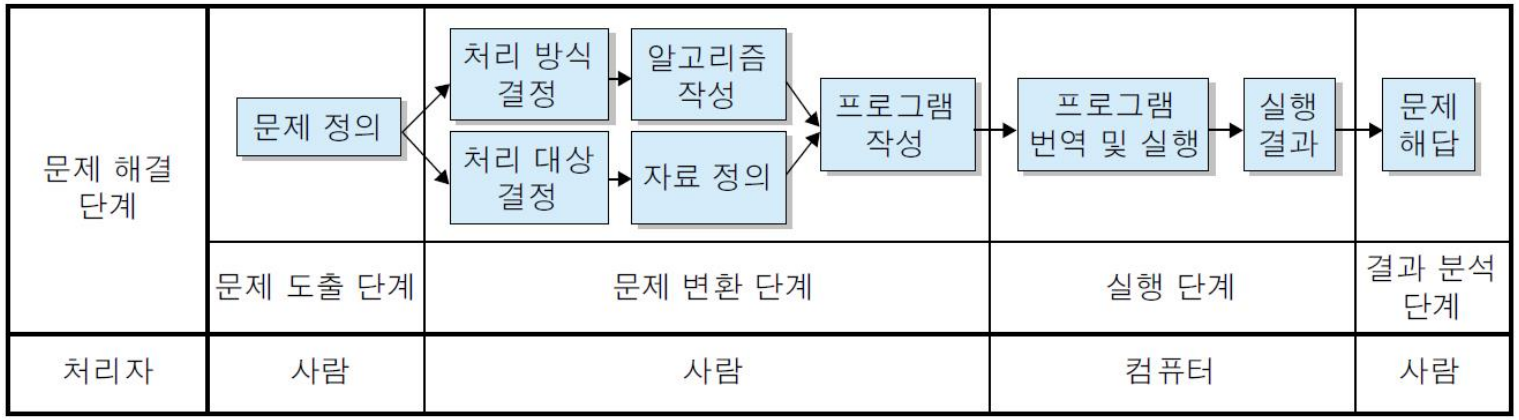
- 간과하기 쉬운 경우
  - 프로그램의 크기가 작은 경우
  - 대형 프로젝트의 초기 단계

    → 구조적인 결함 발생

1. 자료구조 개요

❖컴퓨터 분야에서 자료구조를 왜 배워야 하는가?

- 컴퓨터가 효율적으로 문제를 처리하기 위해서는 문제를 정의하고 분석하여 그에 대한 최적의 프로그램을 작성해야 한다.
  - 자료구조에 대한 개념과 활용 능력 필요!



[그림 1-2] 컴퓨터의 문제 해결 과정

1. 자료구조 개요

- 프로그램의 구조
  - 자료(데이터,data)와 명령으로 구성

Program
Data + Order

예	프로그램의 공통점	
	Order	Data
윈도우 탐색기	파일의 복사, 이동 및 삭제	파일 및 폴더의 계층 구조 정보
전자사전	단어 검색	단어의 철자 및 의미

1. 자료구조 개요

- 자료구조
  - 컴퓨터에 자료를 효율적으로 저장하는 방식
    - 메모리(저장 공간) 절약 (Space Complexity)
    - 프로그램 수행(실행) 시간 단축 (Time Complexity)

  - 프로그램의 수행 시간 혹은 저장 공간을 고려한 자료구조의 설계
    -> 프로그램이 어떻게 사용되는지에 따라 결정
    (프로그램의 목적 및 기능에 부합하는 자료구조 설계)

1. 자료구조 개요

- 자료의 형태에 따른 분류
  - 단순 구조
    - 정수, 실수, 문자, 문자열, 등의 기본 자료형
  - 선형 구조
    - 자료들 간의 앞뒤 관계가 1:1의 선형 관계
    - 리스트, 연결 리스트, 스택, 큐, 덱 등
  - 비선형 구조
    - 자료들 간의 앞뒤 관계가 '1:다', 또는 '다:다'의 관계
    - 트리 그래프 등
  - 파일 구조
    - 레코드의 집합인 파일에 대한 구조
    - 순차파일, 색인 파일, 직접파일 등

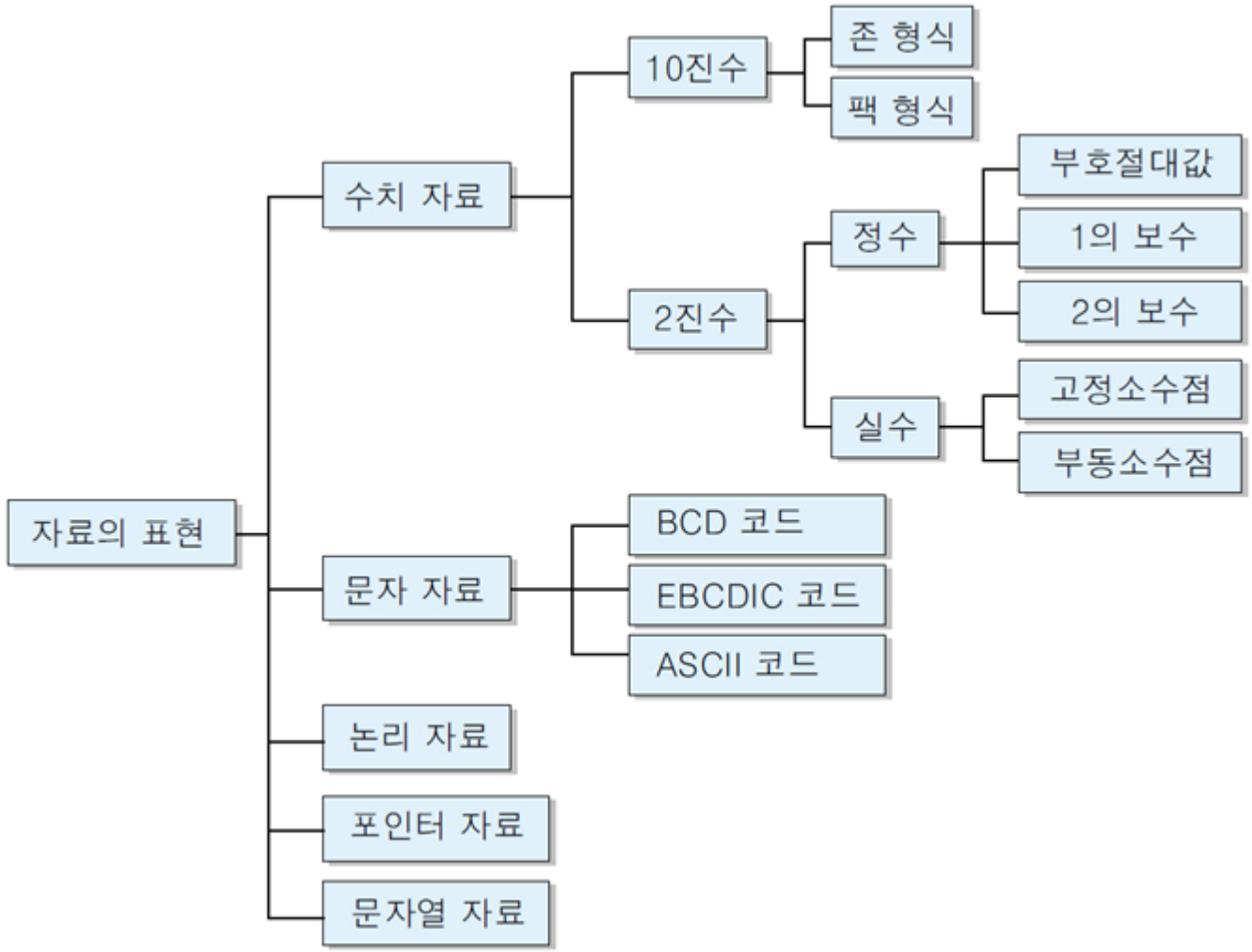
2. 자료구조의 분류

❖자료의 형태에 따른 분류

- 단순 구조
  - 정수, 실수, 문자, 문자열, 등의 기본 자료형
- 선형 구조
  - 자료들 간의 앞뒤 관계가 1:1의 선형 관계
  - 리스트, 연결 리스트, 스택, 큐, 덱 등
- 비선형 구조
  - 자료들 간의 앞뒤 관계가 '1:다', 또는 '다:다'의 관계
  - 트리 그래프 등
- 파일 구조
  - 레코드의 집합인 파일에 대한 구조
  - 순차파일, 색인 파일, 직접파일 등

2. 자료구조의 분류

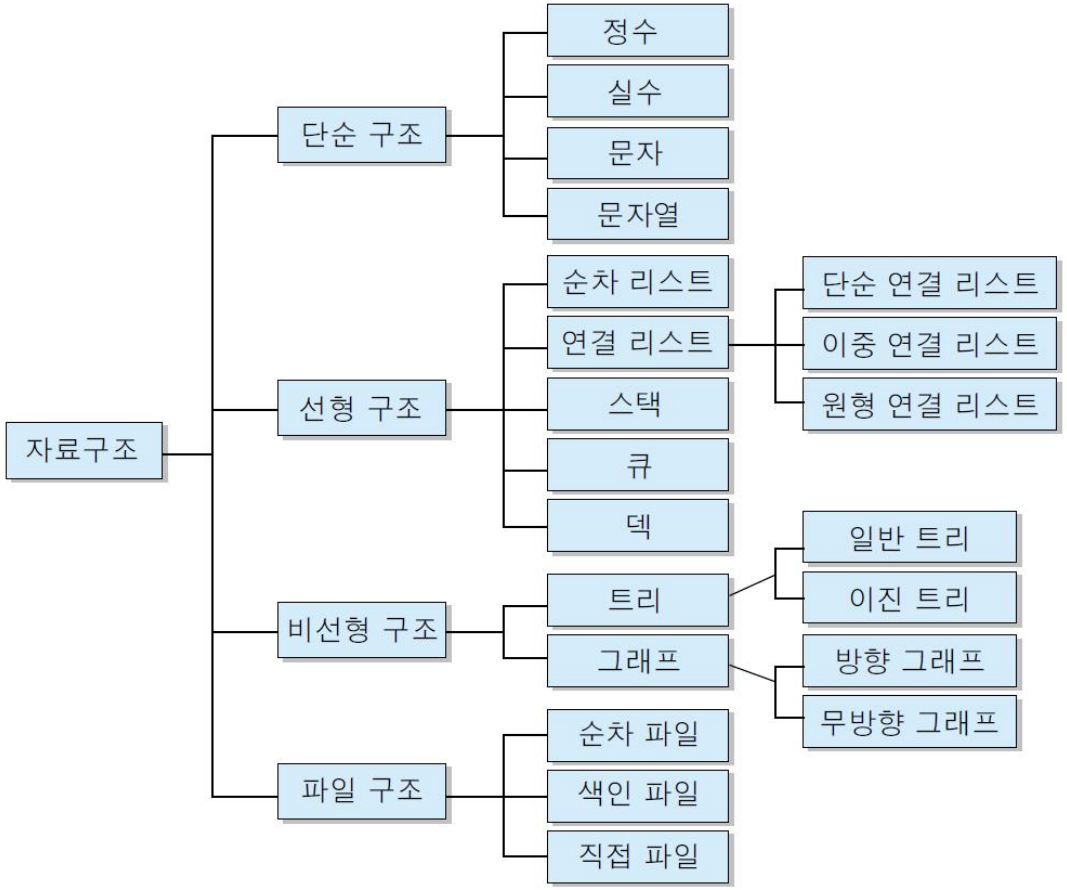
- 컴퓨터 내부에서 표현할 수 있는 자료의 종류



2. 자료구조의 분류

❖자료의 형태에 따른 분류

- 자료구조의 형태에 따른 분류 : [그림 1-4]

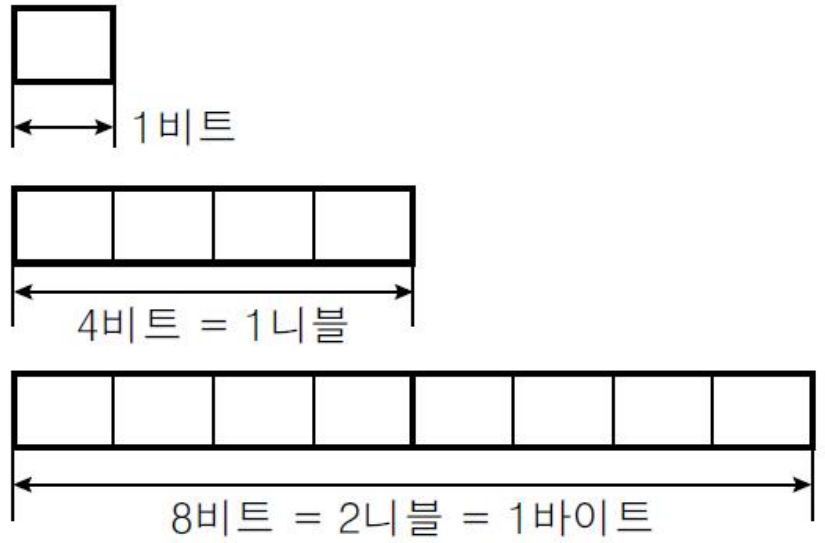


[그림 1-4] 자료구조의 형태에 따른 분류

2. 자료의 표현

❖디지털 시스템에서의 자료 표현

- 숫자, 문자, 그림, 소리, 기호 등 모든 형식의 자료를 2진수 코드로 표현하여 저장 및 처리
- 2진수 코드란?
  - 1과 0, On과 Off, 참(True)과 거짓(False)의 조합
- 2진수 코드의 단위

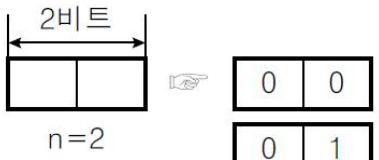


[그림 1-5] 비트, 니블과 바이트

2. 자료의 표현

❖디지털 시스템에서의 자료 표현

- n개의 비트로 2^n개의 상태수 표현
  - 예) n = 2인 겨우

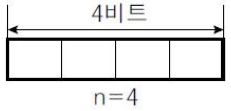


0	0
0	1
1	0
1	1

4개($2^n = 2^2 = 4$)의 상태값 표현

[그림 1-6] n개의 비트로 2^n개의 상태 표현(n=2인 경우)

n = 4인 경우



0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

16개($2^n = 2^4 = 16$)의 상태값 표현

[그림 1-7] n개의 비트로 2^n개의 상태 표현(n=4인 경우)

2. 자료의 표현 : 수치자료의 표현

❖10진수의 표현

- 존(Zone) 형식의 표현
  - 10진수 한 자리를 표현하기 위해서 1바이트(8비트)를 사용하는 형식
  - 존 영역
    - 상위 4비트
    - 1111로 표현
  - 수치 영역
    - 하위 4비트
    - 표현하고자 하는 10진수 한 자리 값에 대한 2진수 값을 표시
  - 존 형식의 구조

존 영역				수치 영역			
				8	4	2	1
x	x	x	x	x	x	x	x

[그림 1-9] 존 형식의 구조

2. 자료의 표현 : 수치자료의 표현

❖10진수의 표현

- 수치 영역의 값 표현 : [표 1-1]

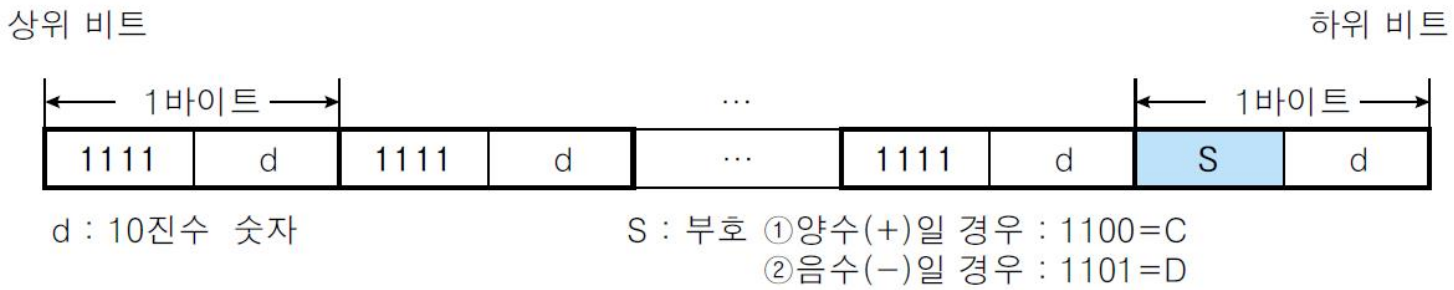
[표 1-1] 4비트의 2진수에 대한 10진수 표현

4비트의 2진수				10진수 변환	10진수
0	0	0	0	0 × 8 + 0 × 4 + 0 × 2 + 0 × 1	0
0	0	0	1	0 × 8 + 0 × 4 + 0 × 2 + 1 × 1	1
0	0	1	0	0 × 8 + 0 × 4 + 1 × 2 + 0 × 1	2
0	0	1	1	0 × 8 + 0 × 4 + 1 × 2 + 1 × 1	3
0	1	0	0	0 × 8 + 1 × 4 + 0 × 2 + 0 × 1	4
0	1	0	1	0 × 8 + 1 × 4 + 0 × 2 + 1 × 1	5
0	1	1	0	0 × 8 + 1 × 4 + 1 × 2 + 0 × 1	6
0	1	1	1	0 × 8 + 1 × 4 + 1 × 2 + 1 × 1	7
1	0	0	0	1 × 8 + 0 × 4 + 0 × 2 + 0 × 1	8
1	0	0	1	1 × 8 + 0 × 4 + 0 × 2 + 1 × 1	9
1	0	1	0	1 × 8 + 0 × 4 + 1 × 2 + 0 × 1	10 = A

2. 자료의 표현 : 수치자료의 표현

❖10진수의 표현

- 여러 자리의 10진수를 표현하는 방법
  - 10진수의 자릿수만큼 존 형식을 연결하여 사용
  - 마지막 자리의 존 영역에 부호를 표시
    - 양수(+) : 1100
    - 음수(-) : 1101

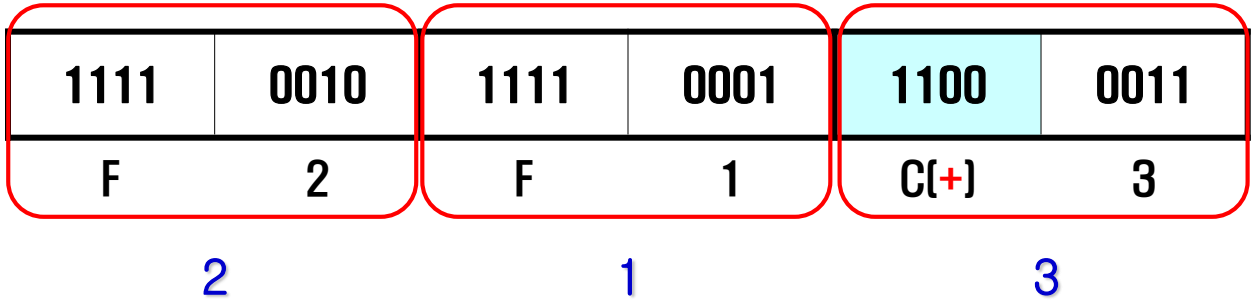


[그림 1-10] 존 형식의 10진수 표현 형식

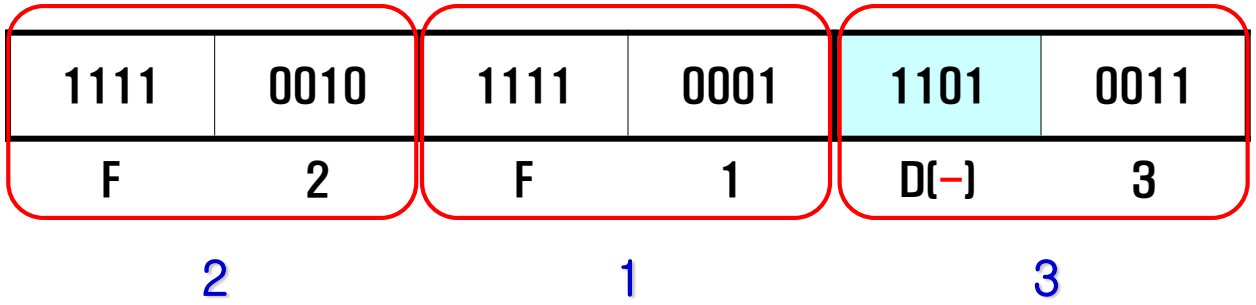
2. 자료의 표현 : 수치자료의 표현

❖10진수의 표현

- 존 형식으로 10진수를 표현하는 예
  - +213



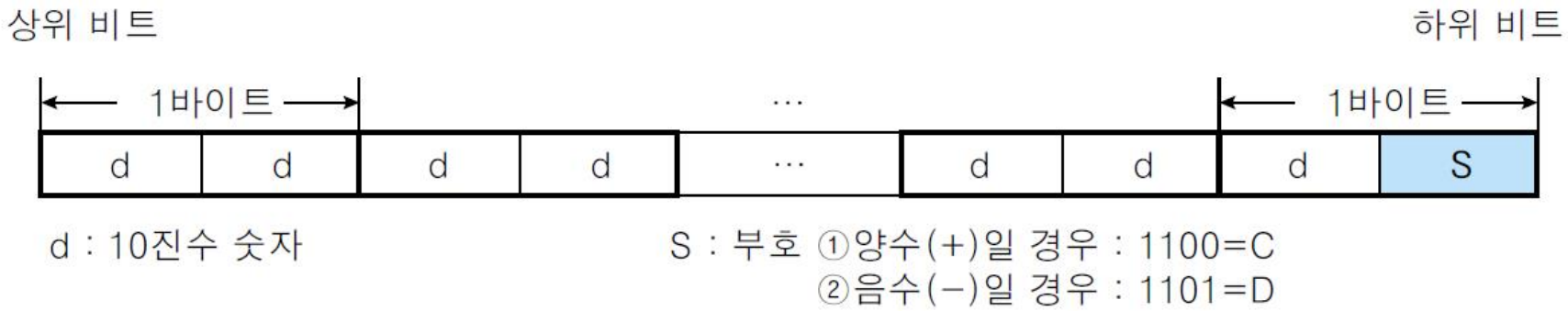
  - -213



2. 자료의 표현 : 수치자료의 표현

❖10진수의 표현

- 팩(Pack) 형식의 표현
  - 10진수 한 자리를 표현하기 위해서 존 영역 없이 4비트를 사용하는 형식
  - 최하위 4비트에 부호를 표시
    - 양수(+) : 1100



[그림 1-11] 팩 형식의 10진수 표현 형식

2. 자료의 표현 : 수치자료의 표현

❖10진수의 표현

- 팩 형식으로 10진수를 표현한 예

① +213

0010	0001	0011	1100
2	1	3	C(+)

② −213

0010	0001	0011	1101
2	1	3	D(−)

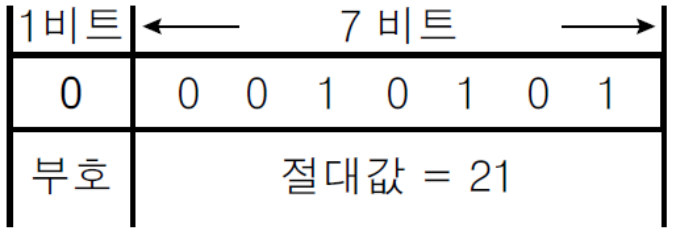
2. 자료의 표현 : 수치자료의 표현

❖2진수의 정수 표현

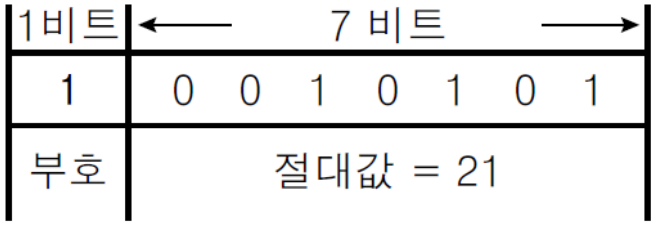
- n비트의 부호 절대값 형식
  - 최상위 1비트 : 부호 표시
    - 양수(+) : 0
    - 음수(-) : 1
  - 나머지 n-1 비트 : 이진수 표시

  - 1바이트를 사용하는 부호 절대값 형식의 예

① +21



② −21



2. 자료의 표현 : 수치자료의 표현

❖2진수의 정수 표현

▪1의 보수(1' Complement) 형식

- 음수의 표현에서 부호 비트를 사용하는 대신 1의 보수를 사용하는 방법
- n비트의 2진수를 1의 보수로 만드는 방법
  - n비트를 모두 1로 만든 이진수에서 변환하고자 하는 이진수를 뺀다.
  - 예) 10진수 21을 1의 보수로 만들기(1바이트 사용)

```
  1 1 1 1 1 1 1 1
- 0 0 0 1 0 1 0 1   ☜ 21의 2진수 값
-----------------
  1 1 1 0 1 0 1 0   ☜ 21의 1의 보수
```

- 1바이트 를 사용하는 1의 보수 형식의 예

☞ 부호절대값형식의 양수 표현과 같음!

2. 자료의 표현 : 수치자료의 표현

❖2진수의 정수 표현

- 2의 보수(2' Complement) 형식
  - 음수의 표현에서 부호 비트를 사용하는 대신 2의 보수를 사용하는 방법
  - n비트의 2진수를 2의 보수로 만드는 방법
    - 1의 보수에 1을 더해준다.
    - 예) 10진수 21을 2의 보수로 만들기(1바이트 사용)

```
   1 1 1 1 1 1 1 1
 - 0 0 0 1 0 1 0 1    ☜ 21의 2진수 값
-----------------
   1 1 1 0 1 0 1 0    ☜ 21의 1의 보수
+                 1
-----------------
   1 1 1 0 1 0 1 1    ☜ 21의 2의 보수
```

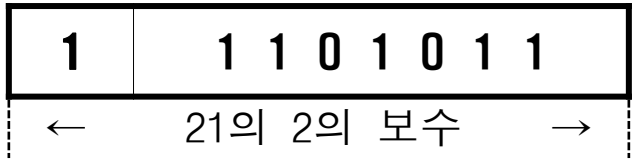
2. 자료의 표현 : 수치자료의 표현

❖2진수의 정수 표현

- 2의 보수(2' Complement) 형식
  - 1바이트를 사용하는 2진 보수 형식의 예

+21

-21



☞ 부호절대값형식의 양수 표현과 같음!

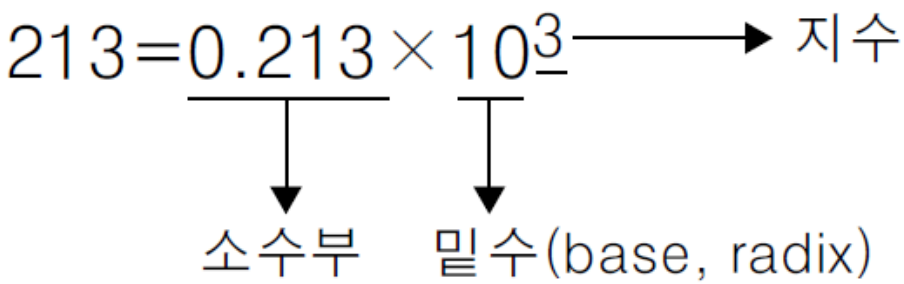
- 2진수 정수의 세 가지 표현 방법에서 양수의 표현은 같고 **음수의 표현만 다르다.**

2. 자료의 표현 : 수치자료의 표현

❖2진수의 실수 표현

- 고정 소수점 표현
  - 소수점이 항상 최상위 비트의 왼쪽 밖에 고정되어 있는 것으로 취급하는 방법
  - 고정 소수점 표현의 **00010101**은 **0.00010101**의 실수 값을 의미

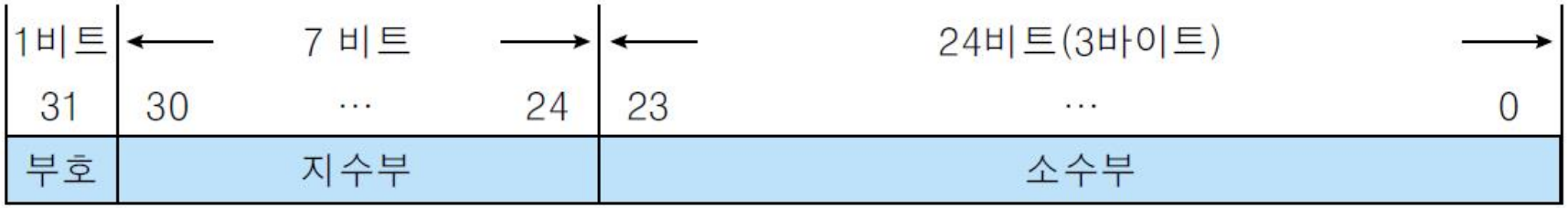
- 부동 소수점 형식의 표현
  - 고정 소수점 형식에 비해서 표현 가능한 값의 범위가 넓다
  - 실수를 구분하여 표현



2. 자료의 표현 : 수치자료의 표현

❖2진수의 실수 표현

- 부동 소수점 형식의 표현
  - 4바이트를 사용하는 부동 소수점 형식



[그림 1-12] 4바이트 부동소수점 표현 형식

2. 자료의 표현 : 문자자료의 표현

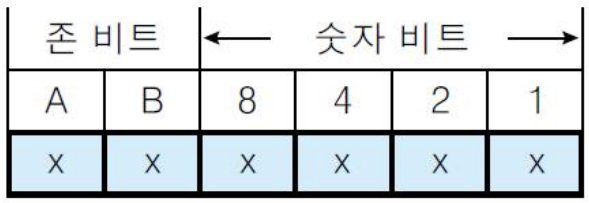
❖문자자료의 표현

- 문자에 대한 이진수 코드를 정의하여 사용
- 문자에 대한 이진수 코드표
  - BCD 코드
  - EBCDIC 코드
  - ASCII 코드

2. 자료의 표현 : 문자자료의 표현

❖BCD 코드

- 6비트를 사용하여 문자 표현
  - 상위 2비트 : 존 비트
  - 하위 4비트 : 2진수 비트
  - 존 비트와 2진수 비트를 조합하여 10진수 0~9와 영어 대문자, 특수 문자를 표현



[그림 1-13] BCD 코드 구성

존 비트 AB의 값
- 00 : 0, 19(1010, 00011001)
- 01 : 문자 AI(00011001)
- 10 : 문자 R(00011001)
- 11 : 문자 S(00101001)

2. 자료의 표현 : 문자자료의 표현

❖BCD 코드

- BCD 코드표
  - 예) 영문자 A에 대한 BCD 코드 ☞ 010001

[표 1-2] BCD 코드표

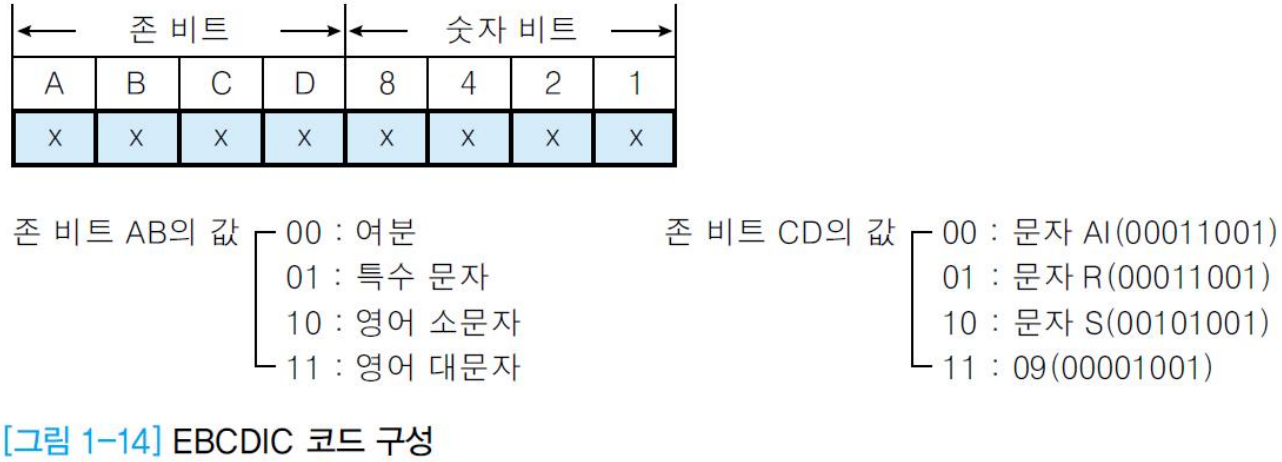
존 비트		숫자 비트				표현 문자	존 비트		숫자 비트				표현 문자	존 비트		숫자 비트				표현 문자	존 비트		숫자 비트				표현 문자
0	0	0	0	0	1	1	0	1	0	0	0	1	A	1	0	0	0	0	1	J							
0	0	0	0	1	0	2	0	1	0	0	1	0	B	1	0	0	0	1	0	K	1	1	0	0	1	0	S
0	0	0	0	1	1	3	0	1	0	0	1	1	C	1	0	0	0	1	1	L	1	1	0	0	1	1	T
0	0	0	1	0	0	4	0	1	0	1	0	0	D	1	0	0	1	0	0	M	1	1	0	1	0	0	U
0	0	0	1	0	1	5	0	1	0	1	0	1	E	1	0	0	1	0	1	N	1	1	0	1	0	1	V
0	0	0	1	1	0	6	0	1	0	1	1	0	F	1	0	0	1	1	0	O	1	1	0	1	1	0	W
0	0	0	1	1	1	7	0	1	0	1	1	1	G	1	0	0	1	1	1	P	1	1	0	1	1	1	X
0	0	1	0	0	0	8	0	1	1	0	0	0	H	1	0	1	0	0	0	Q	1	1	1	0	0	0	Y
0	0	1	0	0	1	9	0	1	1	0	0	1	I	1	0	1	0	0	1	R	1	1	1	0	0	1	Z
0	0	1	0	1	0	0																					

2. 자료의 표현 : 문자자료의 표현

❖EBCDIC 코드

- 8비트를 사용하여 문자 표현
  - 상위 4비트 : 존 비트
  - 하위 4비트 : 2진수 비트
  - 존 비트와 2진수 비트를 조합하여 10진수 0~9와 영어 대문자/소문자와 특수문자를 표현

- EBCDIC 코드의 구성



[그림 1-14] EBCDIC 코드 구성

2. 자료의 표현 : 문자자료의 표현

❖EBCDIC 코드

- EBCDIC 코드 표
  - 예) 영문자 A에 대한 EBCDIC 코드 ☞ 11000001

[표 1-3] EBCDIC 코드표

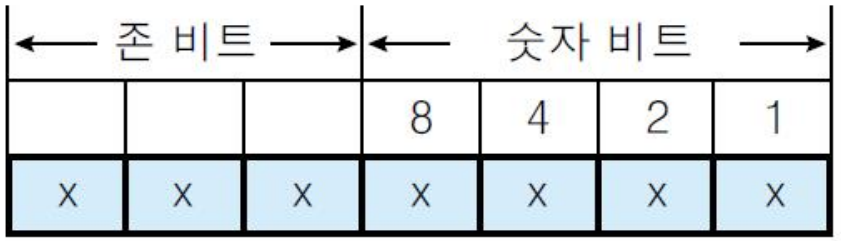
상위 / 하위	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	DLE	DS		SP	&	-						{	}	₩(\)	0
0001	SOH	DC1	SOS				/		a	j	~		A	J		1
0010	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011	ETX	TM							c	l	t		C	L	T	3
0100	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	LC	BS	ETB	UC					f	o	w		F	O	W	6
0111	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
1000	GE	CAN							h	q	y		H	Q	Y	8
1001	RLF	EM							i	r	z		I	R	Z	9
1010	SMM	CC	SM		¢	!		:								
1011	VT	CU1	CU2	CU3	.	$	,	#								
1100	FF	IFS		DC4	<	*	%	@								
1101	CR	IGS	ENQ	NAK	(	)	_	'								
1110	SO	IRS	ACK		+	;	>	=								
1111	SI	IUS	BEL	SUB	|	¬	?	"								

2. 자료의 표현 : 문자자료의 표현

❖ASCII 코드

- 7비트를 사용하여 문자 표현
  - 상위 3비트 : 존 비트
  - 하위 4비트 : 2진수 비트
  - 존 비트와 2진수 비트를 조합하여 10진수 0~9와 영어 대문자/소문자, 특수문자를 표현

- ASCII 코드의 구성



[그림 1-15] ASCII 코드 구성

2. 자료의 표현 : 문자자료의 표현

❖ASCII 코드

- ASCII 코드표
  - 예) 영문자 A에 대한 ASCII 코드 ☞ 1000001

[표 1-4] ASCII 코드표

하위 \ 상위	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	END	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	₩ (\)	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

※ 코드의 의미

GS	Group Separator	RS	Record Separator	US	Unit Separator

2. 자료의 표현 : 논리자료의 표현

❖논리자료

- 논리값을 표현하기 위한 자료 형식
- 논리값
  - 참(True)와 거짓(False), 1과 0
- 1바이트를 사용하여 논리자료를 표현하는 방법
  - 방법 1)
    - 참 : 최하위 비트를 1로 표시 **00000001**
    - 거짓 : 전체 비트를 0으로 표시. **00000000**
  - 방법2)
    - 참 : 전체 비트를 1로 표시. **11111111**
    - 거짓 : 전체 비트를 0으로 표시. **00000000**
  - 방법3)
    - 참 : 하나 이상의 비트를 1로 표시 **00000001 or 00000100 .....**
    - 거짓 : 전체 비트를 0으로 표시. **00000000**

2. 자료의 표현 : 포인터 자료의 표현

❖포인터 자료

- 메모리의 주소를 표현하기 위한 자료 형식
- 변수의 주소나 메모리의 특정 위치에 대한 주소를 저장하고 주소연산하기 위해 사용

2. 자료의 표현 : 문자열 자료의 표현

❖문자열(String) 자료

- 여러 문자로 이루어진 문자의 그룹을 하나의 자료로 취급하여 메모리에 연속적으로 저장하는 자료 형식

- 하나의 문자열 자료에 포함된 부분문자열을 표현하는 방법
  - 방법 1 : 부분문자열 사이에 구분자를 두고 연속 저장하는 방법
  - 방법 2 : 가장 긴 부분문자열의 길이에 맞추어 고정 길이로 연속 저장하는 방법
  - 방법 3 :부분문자열을 연속 저장하고 각 부분문자열에 대한 포인터를 사용하는 방법

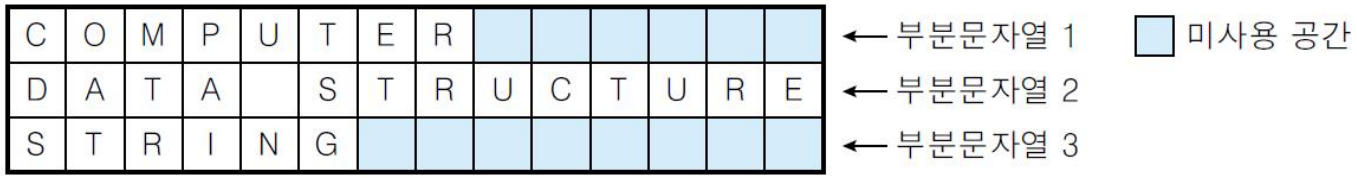
2. 자료의 표현 : 문자열 자료의 표현

❖문자열(String) 자료

- 부분 문자열의 표현 예 {COMPUTER, DATA STRUCTURE, STRING}
  - 방법 1. 구분자를 사용하는 표현 : 구분자로 세미콜론(;) 사용

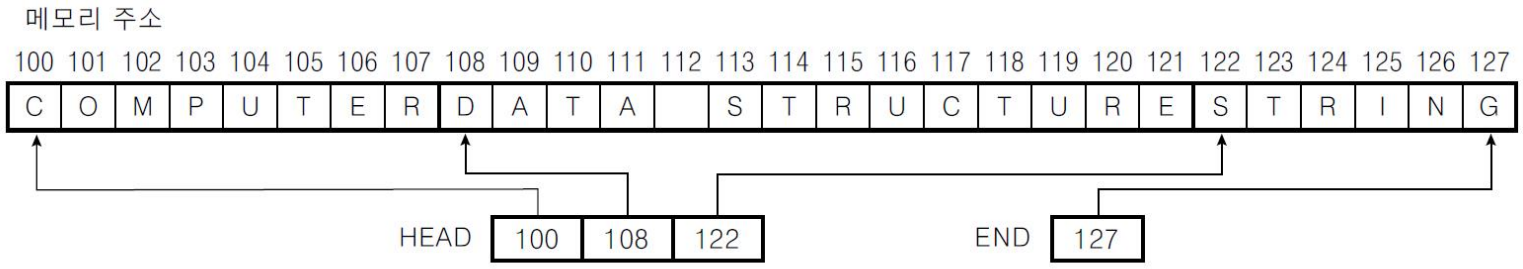
[그림 1-16] 구분자를 사용하여 문자열 저장하는 방법(방법 1)

  - 방법 2. 고정길이를 사용하는 표현



[그림 1-17] 고정 길이로 문자열 저장하는 방법(방법 2)

  - 방법 3. 포인터를 사용하는 표현



[그림 1-18] 포인터를 사용하여 문자열 저장하는 방법(방법 3)

2. 자료의 표현 : 문자열 자료의 표현

❖문자열(String) 자료

- 부분 문자열 표현 방법과 비교

[표 1-5] 문자열 표현 방법 비교

비교 항목 \ 방법	구분자를 사용하는 방법	고정 길이로 저장하는 방법	포인터를 사용하는 방법
메모리 이용률	문자열 길이 + 구분자 길이 ☞ 효율적	가장 긴 부분문자열 길이 × 부분문자열의 개수 ☞ 비효율적	문자열 길이 + 포인터 저장 공간 ☞ 효율적
부분문자열 탐색 시간	문자 비교 연산 시간 + 구분자 식별 시간 ☞ 비효율적	문자 비교 연산 시간 ☞ 효율적	문자 비교 연산 시간 + 포인터 주소 연산 시간 ☞ 효율적