

자료구조론

1장 자료구조 개요

□ 이 장에서 다룰 내용



자료구조 개요



자료구조의 분류

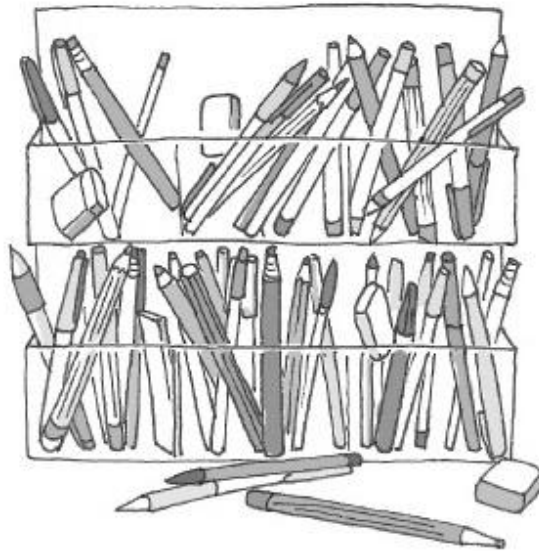


자료의 표현

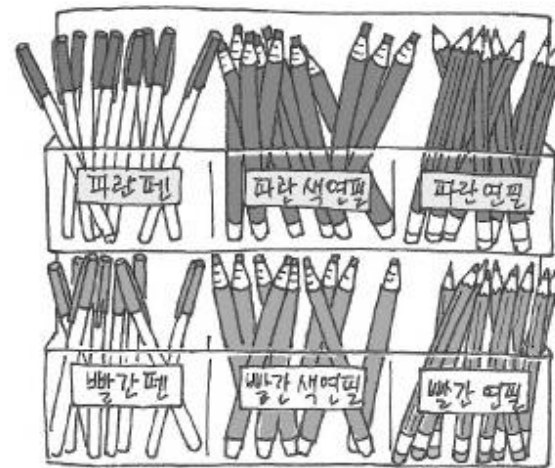
□ 자료구조 개요

❖ 자료구조란?

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



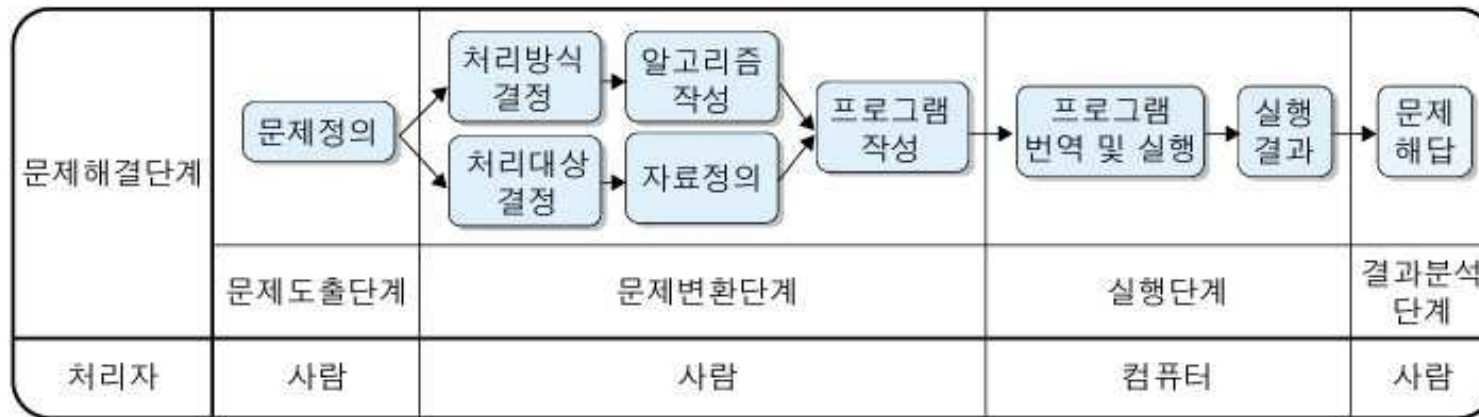
[나쁜 자료구조]



[좋은 자료구조]

□ 자료구조 개요

■ 컴퓨터에 의한 문제 해결 과정



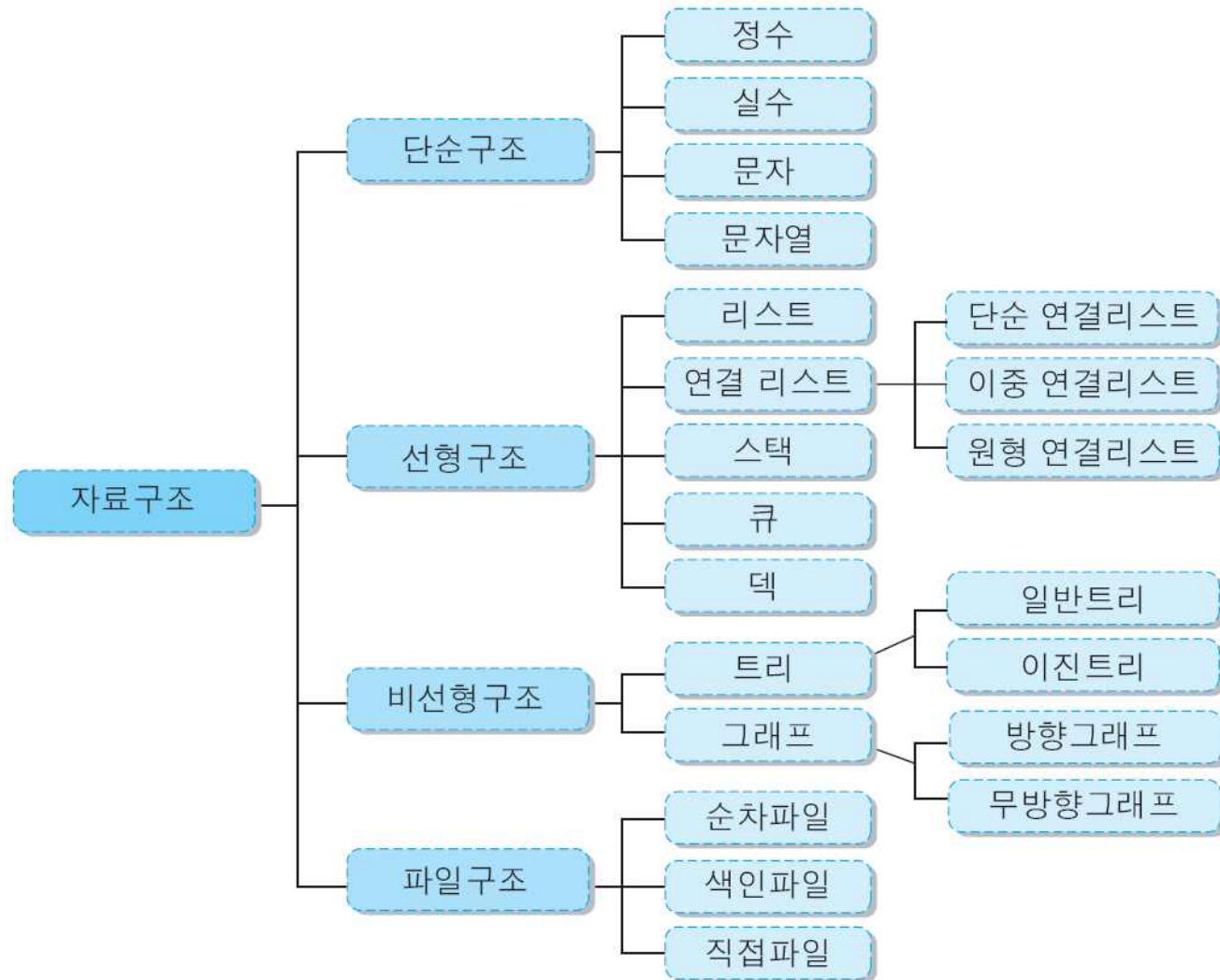
- 컴퓨터가 효율적으로 문제를 처리하기 위해서는 문제를 정의하고 분석하여 그에 대한 최적의 프로그램을 작성해야 한다.
- 프로그램은 자료를 표현하고 그 자료를 처리하여 결과를 얻음

자료구조

알고리즘

☞ 자료구조에 대한 개념과 활용 능력 필요!

□ 자료구조의 분류



□ 자료구조의 분류

❖ 자료의 형태에 따른 분류

- 단순 구조
 - 정수, 실수, 문자, 문자열 등의 기본 자료형
- 선형구조(linear)
 - 자료들 간의 앞뒤 관계가 1:1의 선형 관계
 - 리스트, 연결리스트, 스택(stack), 큐(queue), 덱(deque) 등
- 비선형구조
 - 자료들 간의 앞뒤 관계가 1:多, 또는 多:多의 관계
 - 계층구조(hierarchical), 망구조(network)
 - 트리, 그래프 등
- 파일구조
 - 레코드의 집합인 파일에 대한 구조
 - 순차파일, 색인파일, 직접파일 등

□ 자료구조의 분류

❖ 자료구조의 선택

■ 자료구조 선택의 중요성

- 자료구조는 처리하고자 하는 자료의 표현과 저장 방법을 결정
- 자료구조와 알고리즘은 밀접한 관계
- 따라서 어떤 자료구조를 선택하느냐에 따라 처리 능률이 크게 달라짐

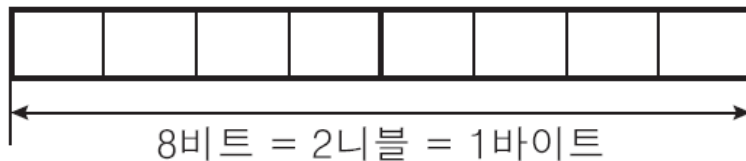
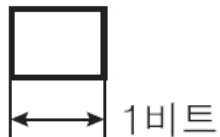
■ 자료구조의 선택의 기준

- 자료의 양
- 자료의 활용 빈도
- 자료의 갱신 정도
- 자료 처리를 위해 사용 가능한 기억장치 용량
- 자료처리 시간의 제한
- 자료처리를 위한 프로그래밍의 용이성

□ 자료의 표현

❖ 디지털 시스템에서의 자료의 표현

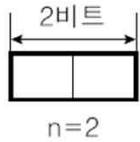
- 숫자, 문자, 그림, 소리, 기호 등 모든 형식의 자료를 2진수 코드로 표현하여 저장 및 처리
- 2진수 코드
 - 1과 0, ON과 OFF, 참(True)과 거짓(False)의 조합
- 2진수 코드의 단위



□ 자료의 표현

■ n개의 비트로 2^n 개의 상태수 표현

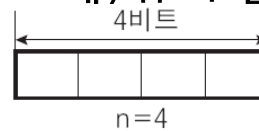
- 예) $n=2$ 인 경우



0	0
0	1
1	0
1	1

4개 ($2^n = 2^2 = 4$)의 상태값 표현

- 예) $n=4$ 인 경우

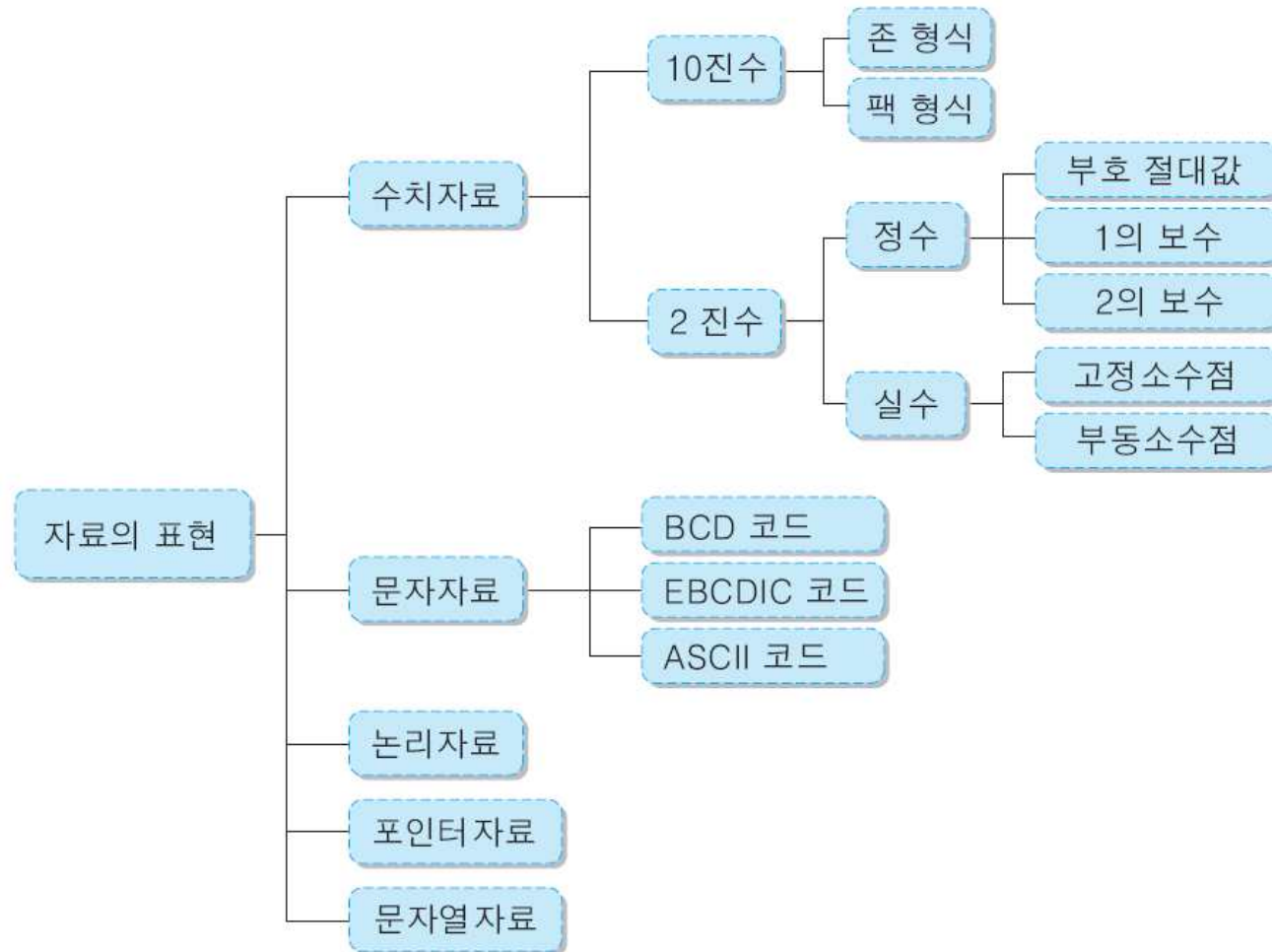


0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

16개 ($2^n = 2^4 = 16$)의 상태값 표현

□ 자료의 표현

❖ 컴퓨터 내부에서 표현할 수 있는 자료의 종류



□ 자료의 표현 - (1) 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

- + 21	➤ - 21
--------	--------

1비트	← 7 비트 →
0	0 0 1 0 1 0 1
부호	절대값 = 21

1비트	← 7 비트 →
1	0 0 1 0 1 0 1
부호	절대값 = 21

□ 자료의 표현 - [1] 수치자료의 표현

■ 1의 보수(1' Complement) 형식

- 음수의 표현에서 부호 비트를 사용하는 대신 1의 보수를 사용하는 방법

- n비트의 2진수를 1의 보수로 만드는 방법

- n비트를 1로 한 이진수에서 변환하고자 하는 이진수를 뺀다.

- 예) 10진수 21을 1의 보수로 만들기 (1바이트 사용)

$$\begin{array}{r} \text{1 1 1 1 1 1 1 1} \\ - 0 0 0 1 0 1 0 1 \\ \hline \text{1 1 1 0 1 0 1 0} \end{array}$$

21의 2진수 값

21의 1의 보수

- 1바이트를 사용하는 1의 보수 형식의 예



부호절대값형식의
양수 표현과 같음!

□ 자료의 표현 - [1] 수치자료의 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)

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ -\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1 \\ \hline 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ +1 \\ \hline 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1 \end{array}$$

☞ 21의 2진수 값

☞ 21의 1의 보수

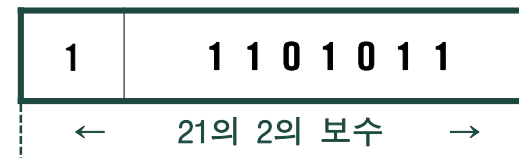
☞ 21의 2의 보수

- 1바이트를 사용하는 2의 보수 형식의 예

- + 21



➤ - 21



☞ 부호절대값형식의
양수 표현과 같음!

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

□ 자료의 표현 - (1) 수치자료의 표현

❖ 2진수의 실수 표현

■ 고정 소수점 표현

- 소수점이 항상 최상위 비트의 왼쪽 밖에 고정되어있는 것으로 취급하는 방법
- 고정 소수점 표현의 00010101 은 0.00010101 의 실수 값을 의미
- 표현 가능한 범위가 작음
 - 예를 들어 8비트를 사용한다고 한다면
 - » 0.00000000011101을 나타내지 못함
 - » 0.00000011011은 0.00000011 로 표현됨

- #### ■ 부동 소수점
- $213 = 0.213 \times 10^3$ —————> 지수
- ↓ ↓
- 소수부 밑수 (base, radix)
- 고정 소수
 - 실수를 부
- | 범위가 넓다.
| 큰하여 표현

□ 자료의 표현 - [1] 수치자료의 표현

❖ 2진수의 실수 표현

■ 부동 소수점 형식의 표현

- 고정 소수점 형식에 비해서 표현 가능한 값의 범위가 넓다.
- 실수를 부호와 지수, 소수의 세 부분으로 구분하여 표현

$$213 = \underbrace{0.213}_{\text{소수부}} \times \underbrace{10^3}_{\text{밑수 (base, radix)}} \longrightarrow \text{지수}$$

□ 자료의 표현 - [1] 수치자료의 표현

- 4바이트를 사용하는 부동 소수점 형식



- 고정 소수점 형식에 비해서 표현 가능한 값의 범위가 넓다.

□ 자료의 표현 - [1] 수치자료의 표현

❖ 10진수의 표현

■ 존(Zone) 형식의 표현

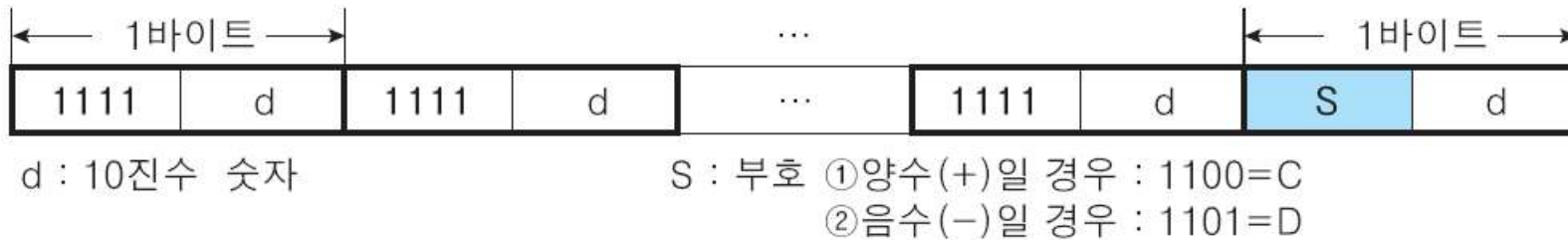
- 10진수 한자리를 표현하기 위해서 1바이트(8비트)를 사용하는 형식
- 존 영역
 - 상위 4비트 : 항상 1111로 표시
- 수치 영역
 - 하위 4비트 : 10진수 한자리 값을 2진수로 표시

• 존 형식의 구조

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

□ 자료의 표현 - [1] 수치자료의 표현

- 여러 자리의 10진수를 표현하는 방법
 - 10진수의 자릿수 만큼 존 형식을 연결하여 사용
 - 마지막 자리의 존 영역에 부호를 표시
 - » 양수(+) : 1100
 - » 음수(-) : 1101
- 존 형식의 10진수 표현 형식



□ 자료의 표현 - (1) 수치자료의 표현

- 존 형식으로 10진수를 표현하는 예
+213

1111	0010	1111	0001	1100	0011
F	2	F	1	C(+)	3
2		1		3	

-213

1111	0010	1111	0001	1101	0011
F	2	F	1	D(-)	3
2		1		3	

□ 자료의 표현 - (1) 수치자료의 표현

■ 팩(Pack) 형식의 표현

- 10진수 한자리를 표현하기 위해서 존 영역 없이 4비트를 사용하는 형식
- 최하위 4비트에 부호를 표시
 - 양수(+) : 1100
 - 음수(-) : 1101

- 팩 형식의 10진수 표현 형식



□ 자료의 표현 - (1) 수치자료의 표현

- 팩 형식으로 10진수를 표현하는 예
+ 213

0010	0001	0011	1100
2	1	3	C(+)

- 213

0010	0001	0011	1101
2	1	3	D(-)

□ 자료의 표현 - (2) 문자자료의 표현

❖ 문자자료의 표현

- 문자에 대한 이진수 코드를 정의하여 사용

- 각 문자마다 서로 다른 값을 부여하기로 약속하고 그 약속에 따라 특정 값이 특정 문자를 나타내는 것으로 본다.

- 문자에 대한 이진수 코드표

- BCD 코드(binary-coded decimal)
- EBCDIC 코드 (extended binary-coded decimal interchange code)
- ASCII 코드(American Standard Code for Information Interchange)
- 유니코드(unicode)

□ 자료의 표현 - (2) 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/소문자와 특수문자를 표현
- ASCII 코드의 구성

← 존 비트 →			← 숫자 비트 →			
			8	4	2	1
x	x	x	x	x	x	x

□ 자료의 표현 - (2) 문자자료의 표현

■ ASCII 코드 표

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
----	-----------------	----	------------------	----	----------------

- 예) 영문자 A에 대한 ASCII 코드 ➡ 1000001

□ 자료의 표현 - (2) 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

← 존 비트 →				← 숫자 비트 →			
A	B	C	D	8	4	2	1
x	x	x	x	x	x	x	x

존 비트 AB의 값

00	: 여분
01	: 특수 문자
10	: 영어 소문자
11	: 영어 대문자

존 비트 CD의 값

00	: 문자 A~I(0001~1001)
01	: 문자 J~R(0001~1001)
10	: 문자 S~Z(0010~1001)
11	: 숫자 0~9(0000~1001)

□ 자료의 표현 - (2) 문자자료의 표현

■ EBCDIC 코드 표

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		┘	?	"								

- 예) 영문자 A에 대한 EBCDIC 코드 ➡ 11000001

□ 자료의 표현 - (2) 문자자료의 표현

■ 유니 코드 표

상위16진수 하위16진수	000	001	002	003	004	005	006	007
0	NUL	DEL	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	

□ 자료의 표현 - (2) 문자자료의 표현

- 영문자 A에 대한 유니 코드 ☞ 0041 ➡ 0000 0000 0100 0001

← 16 비트 →															
상위 16진수												하위 16진수			
0				0				4				1			
상위 2진수												하위 2진수			
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

□ 자료의 표현 - (2) 문자자료의 표현

- 한글에 대한 유니코드
 - 초성 19개 × 중성 21개 × 종성 28개 = 11,172개

[표 1-6] 유니코드표: 한글에 대한 유니코드 중 일부

상위16진수 하위16진수	AC0	AC1	AC2	AC3	AC4	AC5	AC6	AC7
0	가	감	갸	갹	갈	각	갸	거
1	각	갑	갸	갹	갈	갈	갸	격
2	감	갸	갸	갹	갈	갸	갸	귀
3	갸	갸	갸	갹	갈	갸	갸	귀
4	간	갸	갸	갹	갈	개	갸	건
5	갸	강	갸	갸	갈	객	갸	견
6	갸	갸	갸	갹	갈	객	갸	견
7	갈	갸	갸	갹	갈	갸	갸	견
8	갈	각	갸	가	감	갸	갸	걸
9	갸	갈	갸	각	갑	갸	갸	검
A	갸	갸	갸	각	갸	갸	갸	검
B	갸	갸	갸	갸	갸	갸	갸	검
C	갸	개	갸	갸	갸	갸	객	견
D	갸	객	갸	갸	강	갸	갸	견
E	갸	객	갸	갸	갸	갸	갸	견
F	갸	갸	갸	갸	갸	갸	갸	견

□ 자료의 표현 - (2) 문자자료의 표현

- 한글 '가'에 대한 유니 코드  AC00  1010 1100 0000 0000

16 비트

상위 16진수								하위 16진수							
A				C				0				0			
상위 2진수								하위 2진수							
1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0

□ 자료의 표현 - (3) 논리자료의 표현

❖ 논리자료

- 논리값을 표현하기 위한 자료 형식
- 논리값
 - 참(true)과 거짓(false), 1과 0
- 1바이트를 사용하여 논리자료를 표현하는 방법
 - 방법1)
 - 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
 - 거짓 – 전체 비트를 0으로 표시. 00000000

□ 자료의 표현 - (4) 포인터 자료의 표현

❖ 포인터 자료

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

□ 자료의 표현 - [5] 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
 - 방법1) 부분문자열 사이에 구분자를 두고 연속 저장하는 방법
 - 방법2) 가장 긴 부분문자열의 길이에 맞추어 고정 길이로 연속 저장하는 방법
 - 방법3) 부분문자열을 연속 저장하고 각 부분문자열에 대한 포인터를 사용하는 방법

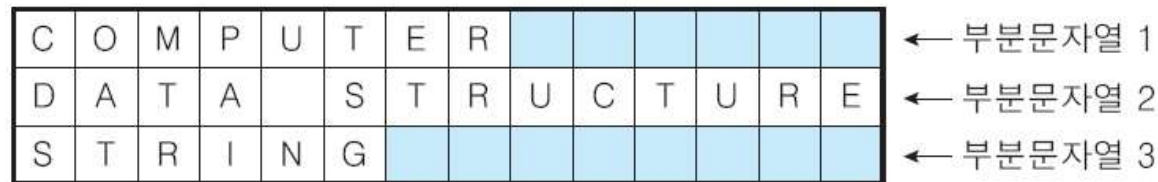
□ 자료의 표현 - (5) 문자열 자료의 표현

■ 부분문자열의 표현 예

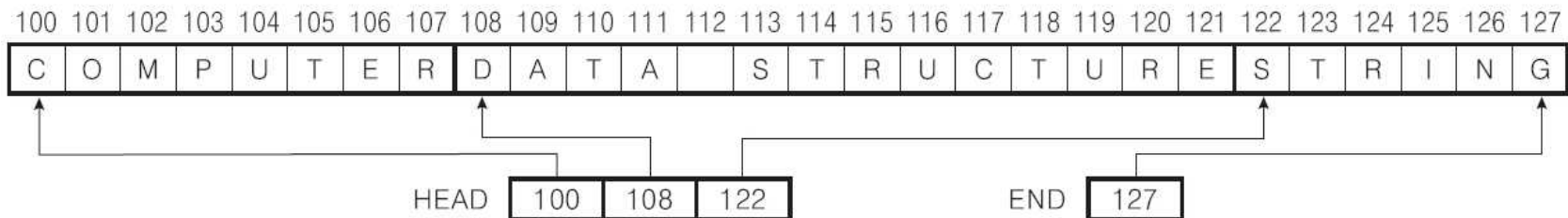
- 표현할 문자열 : {COMPUTER, DATA STRUCTURE, STRING}
- 방법1) 구분자를 사용하는 표현 : 구분자로 세미콜론(;) 사용



방법2) 고정길이를 사용하는 표현



메모리 주소



□ 자료의 표현 - (5) 문자열 자료의 표현

■ 부분 문자열 표현 방법의 비교

비교항목 \ 방법	구분자를 사용하는 방법	고정길이를 저장하는 방법	포인터를 사용하는 방법
메모리 이용률	문자열 길이 + 구분자 길이 ☞ 효율적	가장 긴 부분문자열 길이 × 부분문자열의 개수 ☞ 비효율적	문자열 길이 + 포인터 저장공간 ☞ 효율적
부분문자열 탐색시간	문자 비교연산시간 + 구분자 식별시간 ☞ 비효율적	문자 비교연산시간 ☞ 효율적	문자 비교연산시간 + 포인터 주소연산시간 ☞ 효율적