

자료구조 1주차 2차시

1 | 문자 자료의 표현

1 | 문자 자료의 표현

1 문자 자료의 표현이란?

컴퓨터 내부에서는 문자 자료도
1과 0의 2진수 조합으로 표현

- ▶ 문자의 표현을 위해 문자에 대한 이진수 코드를 정의하여 사용

1 | 문자 자료의 표현

1 문자 자료의 표현이란?

▶ 자주 사용하는 문자에 대한 이진수 코드표

BCD
코드

EBCDIC
코드

ASCII
코드

1 | 문자 자료의 표현

2 BCD 코드 : Binary Coded Decimal Code

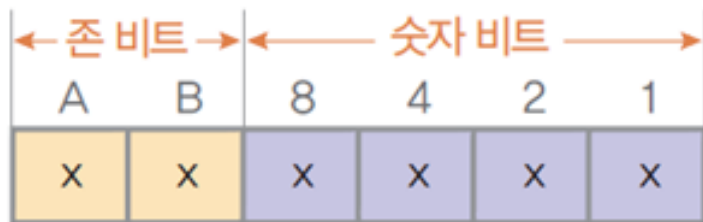
➤ 6비트를 사용하여 문자 표현

상위 2비트

존 비트

하위 4비트

2진수 비트



BCD 코드의 구성

※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

1 | 문자 자료의 표현

2 BCD 코드 : Binary Coded Decimal Code

- ▶ 존 비트와 2진수 비트를 조합하여 10진수 0~9와 영어 대문자, 특수 문자를 표현

존 비트 AB의 값 {

00	: 숫자 0, 1~9(1010, 0001~1001)
01	: 문자 A~I(0001~1001)
10	: 문자 J~R(0001~1001)
11	: 문자 S~Z(0010~1001)

1 | 문자 자료의 표현

2 BCD 코드 : Binary Coded Decimal Code

▶ BCD 코드는 존 비트와 숫자 비트를 조합한 코드

존 비트 01과 숫자 비트 0001을 연결한 010001이 A에 대한 BCD 코드

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
00	0001	1	01	0001	A	10	0001	J			
00	0010	2	01	0010	B	10	0010	K	11	0010	S
00	0011	3	01	0011	C	10	0011	L	11	0011	T
00	0100	4	01	0100	D	10	0100	M	11	0100	U
00	0101	5	01	0101	E	10	0101	N	11	0101	V
00	0110	6	01	0110	F	10	0110	O	11	0110	W
00	0111	7	01	0111	G	10	0111	P	11	0111	X
00	1000	8	01	1000	H	10	1000	Q	11	1000	Y
00	1001	9	01	1001	I	10	1001	R	11	1001	Z
00	1010	0									

※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

※ 0~9까지의 10진수 숫자는 물론 영어 대문자와 특수문자까지 표현 할 수 있음

1 | 문자 자료의 표현

2 BCD 코드 : Binary Coded Decimal Code

존 비트 01과 숫자 비트 0001을 연결한 01 0001이 A에 대한 BCD 코드

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
00	0001	1	01	0001	A	10	0001	J			
00	0010	2	01	0010	B	10	0010	K	11	0010	S
00	0011	3	01	0011	C	10	0011	L	11	0011	T
00	0100	4	01	0100	D	10	0100	M	11	0100	U
00	0101	5	01	0101	E	10	0101	N	11	0101	V
00	0110	6	01	0110	F	10	0110	O	11	0110	W
00	0111	7	01	0111	G	10	0111	P	11	0111	X
00	1000	8	01	1000	H	10	1000	Q	11	1000	Y
00	1001	9	01	1001	I	10	1001	R	11	1001	Z
00	1010	0									

※ 출처 : C로 배우는 쉬운 자료구조
(개정3판), 이지영, 한빛미디어

1 | 문자 자료의 표현

2 EBCDIC 코드 : Extended Binary Coded Decimal Interchange Code

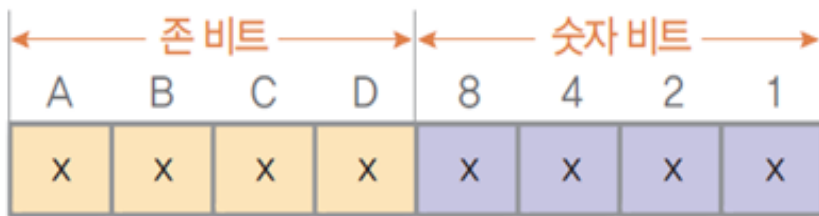
▶ 8비트를 사용하여 문자 표현

상위 4비트

존 비트

하위 4비트

2진수 비트



※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

2 EBCDIC 코드 : Extended Binary Coded Decimal Interchange Code

- ▶ 존 비트와 2진수 비트를 조합하여 10진수 0~9와 영어 대문자/소문자와 특수문자를 표현

존 비트 AB의 값

00	: 여분
01	: 특수문자
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)

1 | 문자 자료의 표현

2 EBCDIC 코드 : Extended Binary Coded Decimal Interchange Code

▶ 열은 상위 4비트의 존 비트, 행은 하위 4비트의 숫자 비트를 표현

A가 있는 자리와 열존 비트 '1100'과 행숫자 비트 '0001'을 연결한 '11000001'이 A에 대한 EBCDIC 코드

표 1-5 EBCDIC 코드 표

상위 비트	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	DLE	DS		SP	&	-						()	WV	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	IRS		DC4	<	*	%	@								
1101	CR	IGS	ENQ	NAK	()	-	'								
1110	SO	IRS	ACK		+	:	>	=								
1111	SI	IUS	BEL	SUB		~	?	*								

- EOT End Of Transmission
- ESCO ESCape
- ETB End of Transmission Block
- ETX End of Text
- FF From Feed
- FS File Separator
- HT Horizontal Tab
- IFS Interchange File Separator
- PN Punch ON
- RES Reader Stop
- RS Shift In
- SM Set Mode
- SMM Start of Manual Message
- SO Shift Out
- SOH Shift Of Heading
- SOS Interchange Group Separator
- IL ILQ
- ACK Acknowledge
- BEL Bell
- BS Backspace
- BYP Bypass
- CAN Cancel
- CC Cipher Control
- CG Storage Control
- CU1 Customer Use 1
- CU3 Customer Use 3
- IFS Interchange File Separator
- IUS Interchange Unit Separator
- LC Lower Case
- LP Line Feed
- NAK Negative Acknowledge
- NL New Line
- NUL Null
- PF Print of 1
- SOS Shift Of Significance
- DC1 Device Control 1
- DC2 Device Control 2
- DC4 Device Control 4
- DEL Delete
- DLE Data Link Escape
- CUE Customer Use 2
- DS Digit Select
- EM End of Medium
- ENG ENGQue
- SP Space
- STX Start of Text
- SUB Substitute
- SYN Synchronous
- TM Tape Mark
- UC Upper Case
- VT Vertical Tab
- WV Word Separator
- ~ Logical NOT

- DC1 Device Control 1
- DC2 Device Control 2
- DC4 Device Control 4
- DEL Delete
- DLE Data Link Escape
- CUE Customer Use 2
- DS Digit Select
- EM End of Medium
- ENG ENGQue
- SP Space
- STX Start of Text
- SUB Substitute
- SYN Synchronous
- TM Tape Mark
- UC Upper Case
- VT Vertical Tab
- WV Word Separator
- ~ Logical NOT

※ 출처 : C로 배우는 쉬운
자료구조(개정3판), 이지영,
한빛미디어

※ 열과 행의 비트를 연결하면 8비트의 EBCDIC 코드를 구할 수 있음

1 | 문자 자료의 표현

2 EBCDIC 코드 : Extended Binary Coded Decimal Interchange Code

A가 있는 자리와 열(존 비트) '1100'과 행(숫자 비트) '0001'을 연결한 '11000001'이 A에 대한 EBCDIC 코드

표 1-5 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	&	-						{	}	W/V	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		~	?	"								

- EOT End of Transmission
- ESC ESCape
- ETB End of Transmission Block
- ETX End of Text
- FF From Feed
- FS Field Separator
- HT Horizontal Tab
- IFS Interchange File Separator
- PN Punch on
- RES REStore
- RS Reader Stop
- S Shift In
- SM Set Mode
- SMM Start of Manual Message
- SO Shift Out
- SOH Start Of Heading
- IGS Interchange Group Separator
- IL ILdo
- ACK Acknowledge
- BEL BELl
- BS BackSpace
- BYP BYPass
- CAN CANceL
- CC Cursor Control
- CR Carriage Return
- CU1 Customer Use 1
- CU3 Customer Use 3
- IRS Interchange Record Separator
- IUS Interchange Unit Separator
- LC Lower Case
- LF Line Feed
- NAK Negative Acknowledge
- NL New Line
- NUL NULl
- PF Punch off
- SOS Start Of Significance
- DC1 Device Control1
- DC2 Device Control2
- DC4 Device Control4
- DEL DElete
- DLE DElete Link Escape
- CU2 Customer Use 2
- DS Digital Select
- EM End of Medium
- ENQ ENQuire
- SP SPace
- STX Start of Text
- SUB SUBstitute
- SYN SYNchronous
- TM Tape Mark
- UC Upper Case
- VT Vertical Tab
- ~ Cent Sign
- Logical NOT

※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

1 | 문자 자료의 표현

3 ASCII 코드 : American Standard Code for Information Interchange

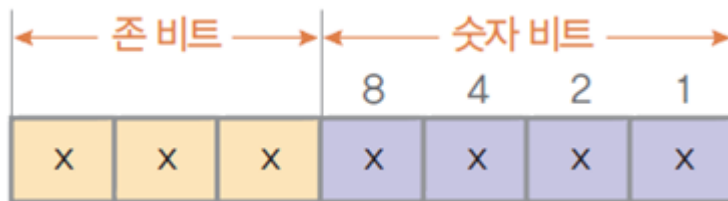
▶ 7비트를 사용하여 문자 표현

상위 3비트

존 비트

하위 4비트

2진수 비트



※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

1 | 문자 자료의 표현

3 ASCII 코드 : American Standard Code for Information Interchange

▶ 열은 상위 3비트의 존 비트, 행은 하위 4비트의 숫자 비트를 나타냄

A가 있는 자리와 열(존 비트) '100'과 행(숫자 비트) '0001'을 연결한 '1000001'이 A에 대한 ASCII 코드

표 1-6 ASCII 코드 표

상위 하위	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

- GS Group Separator
- RS Record Separator
- US Unit Separator

※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

※ 열과 행의 비트를 연결하면 7비트의 ASCII코드를 구할 수 있음

1 | 문자 자료의 표현

3 ASCII 코드 : American Standard Code for Information Interchange

A가 있는 자리의 열(존 비트) '100'과 행(숫자 비트) '0001'을 연결한 '100001'이 A에 대한 ASCII 코드

표 1-6 ASCII 코드 표

상위 하위	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	W(V	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

- GS Group Separator
- RS Record Separator
- US Unit Separator

※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

3 ASCII 코드 : American Standard Code for Information Interchange

- ▶ ASCII 코드를 데이터 통신용으로 사용할 때는 최상위 비트에 패리티비트를 추가하여 8비트 형식으로 사용하기도 함
- ▶ 대문자 A는 65, 소문자 a는 97 코드 값을 가짐
- ▶ C언어에서 문자의 자료형 표현은 char 키워드를 사용

1 | 문자 자료의 표현

4 유니코드

- ▶ EBCDIC 코드나 ASCII 코드는 최대 8비트로 숫자, 몇 가지 특수문자, 알파벳 정의하므로 문자 코드 표에 정의되어 있지 않은 문자 표현 불가능

4 유니코드

▶ 이런 문제 해결 위해 세계 여러 나라의 언어를 통일된 방법으로 표현할 수 있도록 정의한

국제 표준 코드(ISO/IEC 10646)

- 2바이트를 조합하여 하나의 글자를 표현하기 때문에 1바이트 코드로 표현할 수 없었던 다양한 언어를 표현
- 유니코드 표는 <http://www.unicode.org/>에서 확인 가능

4 유니코드

- ▶ 초기 IBM 컴퓨터 시스템에서는 BCD 코드를 사용하다가 더 많은 문자 코드를 표현할 수 있는 EBCDIC 코드로 대체



그러다 미국 표준 코드인 ASCII 코드 일반화



현재는 표현의 한계를 극복한 유니코드가 일반화

(※ XML, Java, CORBA 3.0, WML 등 인터넷 기반 프로그램과 제품에 사용)

2 | 논리 자료의 표현

2 | 논리 자료의 표현

1 논리 자료

논리값을 표현하기 위한 자료 형식

▶ 논리값

- 참(True)와 거짓(False), 1과 0
- 1비트로 표현 할 수도 있지만 일반적으로 컴퓨터 내부에서는 1바이트나 1워드를 논리값을 표현하는 단위로 사용

2 | 논리 자료의 표현

1 논리 자료

논리값을 표현하기 위한 자료 형식

▶ 논리값

- C언어에서 논리 표현 자료형은 bool 키워드를 사용하고, 0이 아닌 값은 True를 0값을 False로 처리함

2 | 논리 자료의 표현

1 논리 자료

▶ 1바이트를 사용하여 논리자료를 표현하는 방법

구분	방법 1	방법 2	방법 3
참	최하위 비트를 1로 표시 예시) 00000001	전체 비트를 1로 표시 예시) 11111111	하나 이상의 비트를 1로 표시 예시) 00000001 or 00000100
거짓	전체 비트를 0으로 표시 예시) 00000000	전체 비트를 0으로 표시 예시) 00000000	전체 비트를 0으로 표시 예시) 00000000

3 | 포인터 자료의 표현

3 | 포인터 자료의 표현

1 포인터 자료

메모리의 주소를 표현하기 위한 자료 형식

- ▶ 포인터 자료는 자료를 저장하고 있는 변수의 주소나 메모리의 특정 위치에 대한 주소를 저장하고 주소 연산을 할 때 사용
- ▶ 포인터를 사용하면 복잡한 자료구조 연산을 메모리에서의 주소 연산만으로 처리할 수 있음

3 | 포인터 자료의 표현

1 포인터 자료

- ▶ 포인터는 자료형에 상관없이 메모리 주소 한 개를 저장하는 변수이므로 2바이트를 사용
- ▶ C언어에서의 포인터 표현방법으로는 * 표시를 사용하여 일반 값을 저장하는 변수와 구별을 하게 됨
- ▶ 포인터를 사용하면 메모리를 직접 엑세스 할 수 있음

☞ 포인터의 자료표현은 3주차에서 좀더 자세히 학습함

4 | 문자열 자료의 표현

4 | 문자열 자료의 표현

1 문자열(String) 자료

한 글자로만 표현할 수 있는 문자 자료와 달리
여러 문자로 이루어진 **문자의 그룹**을 하나의
자료로 취급하여 메모리에 **연속적**으로
저장하는 자료 형식

- ▶ C언어에서 문자열의 표현방법은 포인터를
사용하거나 배열을 이용

1 문자열(String) 자료

▶ 하나의 문자열 자료에 포함된 부분문자열을 표현하는 방법

방법 1	방법 2	방법 3
<ul style="list-style-type: none">■ 부분 문자열 사이에 구분자를 사용하여 저장	<ul style="list-style-type: none">■ 가장 긴 문자열의 길이에 맞춰 고정 길이로 저장	<ul style="list-style-type: none">■ 부분 문자열을 연속하여 저장하고 각 부분 문자열에 대한 포인터를 사용

4 | 문자열 자료의 표현

2 문자열(String) 자료의 표현 예

▶ {COMPUTER, DATA STRUCTURE, STRING} 저장

방법 1 | 구분자를 사용하는 표현

: 구분자로 세미콜론(;) 사용



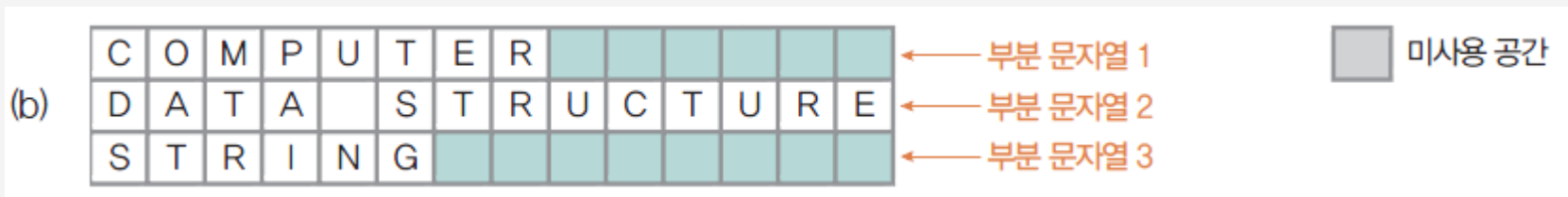
※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

4 | 문자열 자료의 표현

2 문자열(String) 자료의 표현 예

▶ {COMPUTER, DATA STRUCTURE, STRING} 저장

방법 2 | 고정길이를 사용하는 표현



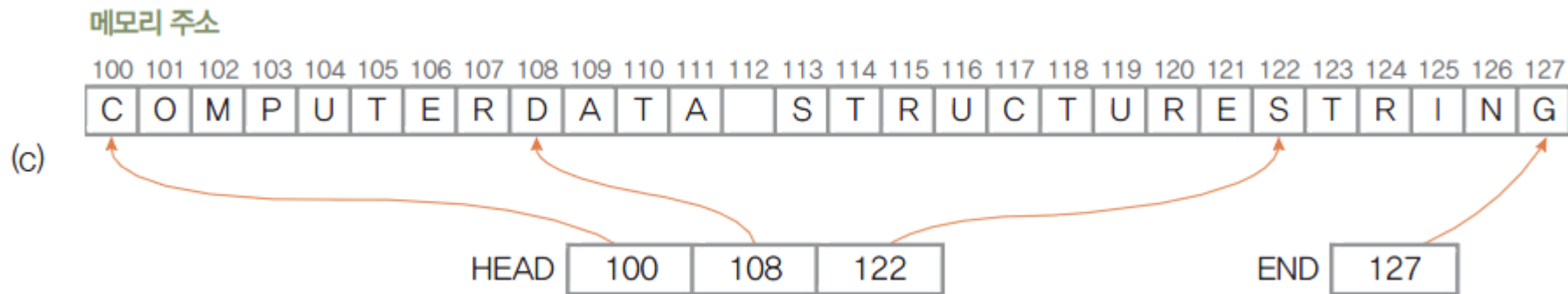
※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

4 | 문자열 자료의 표현

2 문자열(String) 자료의 표현 예

▶ {COMPUTER, DATA STRUCTURE, STRING} 저장

방법 3 | 포인터를 사용하는 표현



※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어



2 문자열(String) 자료의 표현 예

▶ 문자열 자료의 세가지 표현방법의 메모리 이용률과
부분 문자열 탐색시간을 비교하면 아래 표와 같음

※ 출처 : C로 배우는 쉬운 자료구조(개정3판), 이지영, 한빛미디어

문자열 표현 방법 비교

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