中国科学院大学
University of Chinese Academy of Sciences

CS101

Digital Symbol Manipulation

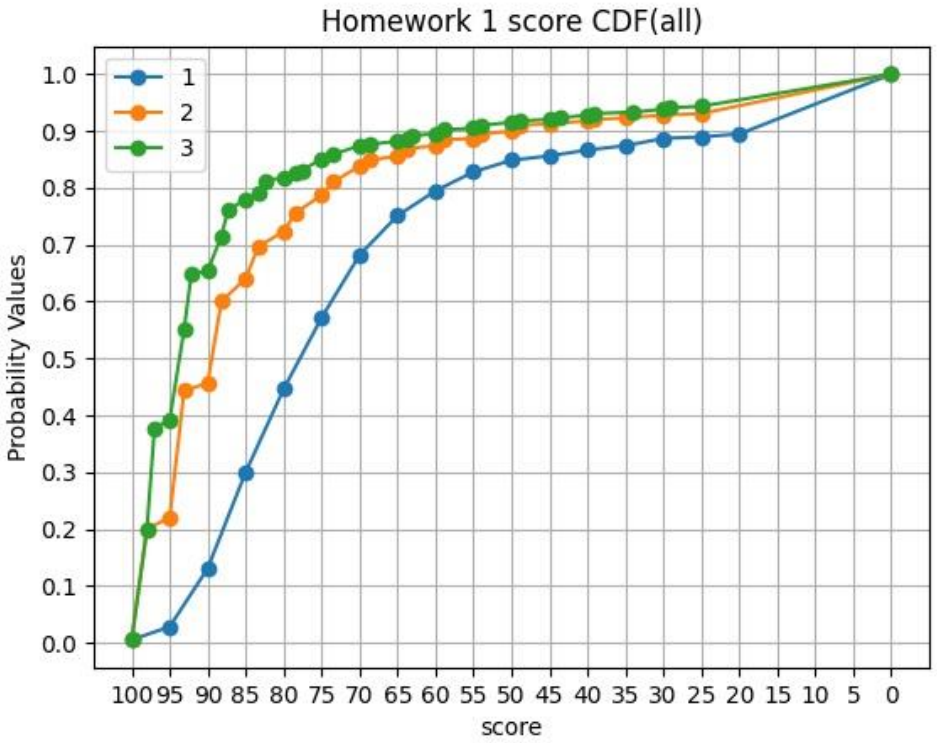
Data as Symbols

zxu@ict.ac.cn
zhangjialin@ict.ac.cn

HW-1成绩

第1、2、3次提交：350人、324人、207人

- 最终成绩 = max(第一次*1, 第二次*0.98, 第三次*0.97)
- 第一次：有同学得100分；**中位数为75分**；20.5%同学不及格
- 第二次：**中位数为88分**；12.5%同学不及格
- 最终成绩
  - **成绩中位数为93分**
  - 67%的学生得90分+
  - 10.5%同学不及格



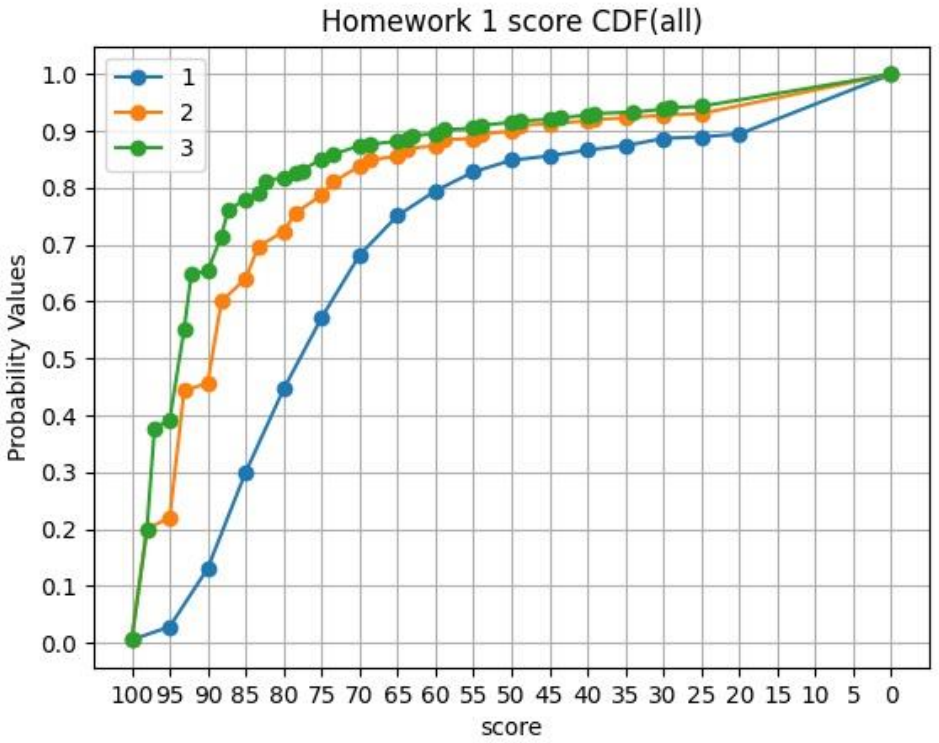
- 作业难度不大
  - 认真完成可以得高分

HW-1成绩

第1、2、3次提交：350人、324人、207人

某同学最终成绩：92.15分
第一次：70分
第二次：80分
第三次：95分

- 最终成绩 = max(第一次*1, 第二次*0.98, 第三次*0.97)
- 第一次：有同学得100分；**中位数为75分**；20.5%同学不及格
- 第二次：**中位数为88分**；12.5%同学不及格
- 最终成绩
  - **成绩中位数为93分**
  - 67%的学生得90分+
  - 10.5%同学不及格
    - **5.64%同学0分（要退课？）**
  - **约4%同学是做得不够好**
    - 请找助教帮助！



答疑助教分配与时间安排

- 答疑助教分配（按实验课小班分配）
  - 第1、5小班：裴晓坤 peixiaokun19@mails.ucas.ac.cn
  - 第2、6小班：杨泽超 yangzechao19@mails.ucas.ac.cn
  - 第3、7小班：林志达 linzhida19@mails.ucas.edu.cn
  - 第4、8小班：李思悦 lisiyue19@mails.ucas.ac.cn
- 平时答疑
  - 请发电子邮件到本班助教
  - 有延迟
- QQ群全班实时答疑
  - 周五晚上：19:00-22:00
  - 周六上午：09:00-12:00
  - 周六下午：14:00-17:00
  - 周日上午：09:00-12:00
  - 周日下午：14:00-17:00

请同学们要有准备
要上课
要看书
要学习课件
先做题

无准备找助教，
效率很低

Outline

- What are digital symbol manipulations?
  - Review of bits, digital symbols, data, and programs
  - Information transformation is digital symbol manipulation
- Data are digital symbols
  - Analog vs. digital values
  - Binary-decimal conversion
  - integers, ASCII characters
  - Characters to numbers
- Programs are digital symbols
- Computers are platforms of digital symbol manipulations

These slides acknowledge sources for additional data not cited in the textbook

1. What is a digital symbol

- A notation
  - Representable as one or more bits
    - Discrete, not continuous
    - Finite, not infinite many bits
  - Denoting any concrete or abstract entity
    - Number, character, string, image, audio, video
    - Idea, program code, human genome sequence
- Manipulation is
  - Any sequence of operation on symbols
    - Find (look up), read, write, transport (send), modify

A WeChat QR code

```
1        nnnnnnnnnn nntaccttcc caggtaacaa accaaccaac tttcgatctc ttgtagatct
61       gttctctaaa cgaactttaa aatctgtgtg gctgtcactc ggctgcatgc ttagtgcact
121      cacgcagtat aattaataac taattactgt cgttgacagg acacgagtaa ctcgtctatc
                                        ……
29821    tttagtagtg ctatccccat gtgattttaa tagnnnnnnn nnnnnnnnnn aaaaaaaaaa
29881    aaaaaaaaaa aaaaaaaaaa aaa
```

Portion of the genome sequence of a SARS-CoV-2 virus

https://www.ncbi.nlm.nih.gov/nuccore/LR991581

Symbols are carriers of civilizations

- 永乐大典
  Yongle Encyclopedia
  - Historical changes of character 兴
    - 篆书
    - 隶书
    - 真书（楷书）
    - 草书
    - 章草

- Digital symbols are carriers of modern civilizations
  - Unicode of 兴: U+5174
  - 我高兴 → U+ 6211, 9AD8, 5174
    I'm happy

Source: The National Library of China

Digital symbol manipulation is at the core of modern civilizations

- Computer science studies computational processes

- A computational process is a process of information transformation,
  which is a sequence of steps of digital symbol manipulation
  - Process = a sequence of steps
  - Information transformation = digital symbol manipulation

Process of information transformation

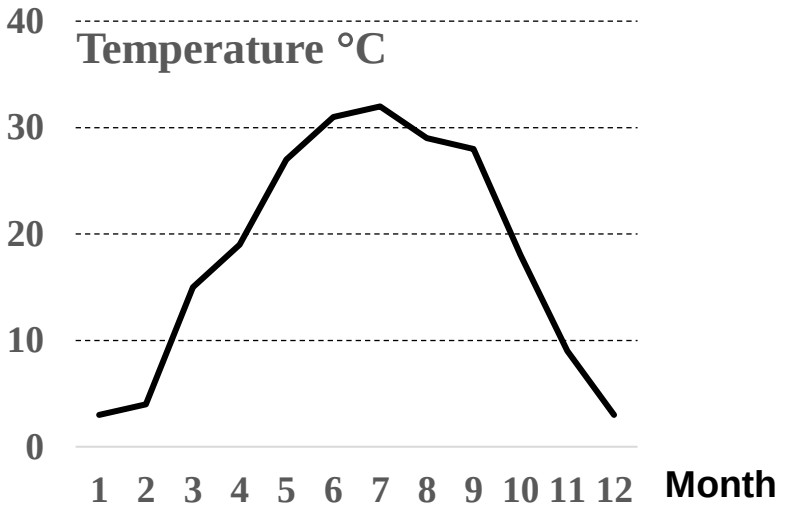
a sequence of steps digital symbol manipulation

2. Data are digital symbols

- Analog vs. digital values
- Binary-decimal conversion
- Representing integers
- Representing ASCII characters

2.1 Analog vs. digital values

- Discretization: mapping analog values to discrete values
  - No intermediate value between two consecutive discrete values
    - E.g., between 3 and 4, between 28 and 29
  - There is always a value between any two analog values
    - E.g., between 3 and 4, between 3 and 3.1
- Use a word of bits to represent a value
  - E.g., 2 bits in the following table
    - A 2-bit word can represent any value from 0 to 3 (2^2-1)

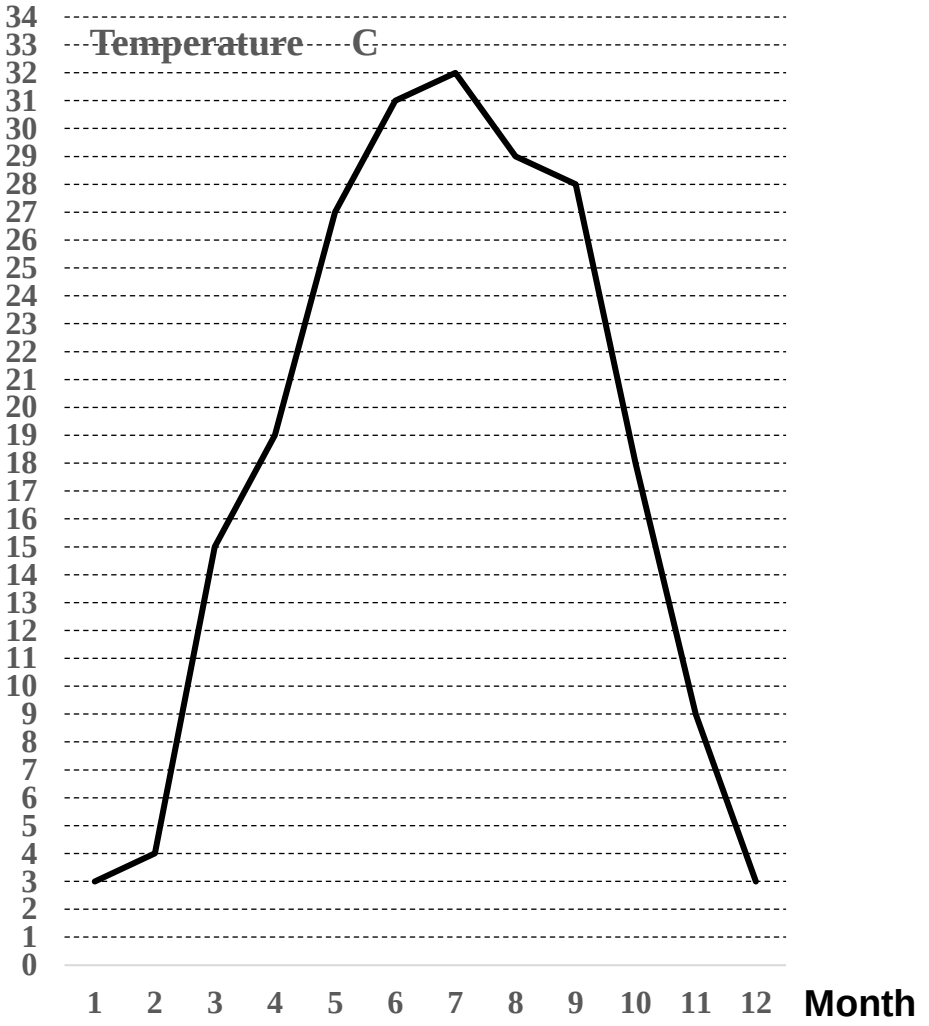


Month		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Temperature °C	Decimal	0	0	20	20	30	30	30	30	30	20	10	0
	Binary	00	00	10	10	11	11	11	11	11	10	01	00

Average monthly high temperature value in Beijing in 2019

More precise discretization

- Use a word of bits to represent a value
  - E.g., 5 bits in the following table
  - A 5-bit word can represent any value from 0 to 31 (2^5-1)
  - A 5-bit word cannot represent 32
    - Use 6 bits
    - Use 5 bits and an overflow bit
      - Output 31 and overflow



Month		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Temperature °C	Decimal	3	4	15	19	27	31	32	29	28	18	9	3
	Binary	00011	00100	01111	10011	11011	11111	?	11101	11100	10010	01001	00011

Average monthly high temperature value in Beijing in 2019

Bit, byte, word

- Bit: 1 binary digit
  - &&, ||, ! AND, OR, NOT
- Byte: a group of 8 bits
  - Most memory is byte addressable; load, store
- Word: a group of k bits
  - The processor operate on words, not part of a word
    - +, -, *, /, %; Add, subtract, multiply, divide, modulus
    - ++, -- increment, decrement
    - >>, <<; shift right, shift left
    - &, |, ^ bitwise AND, bitwise OR, bitwise NOT
  - Most modern processors have k = 64
    - They are 64-bit processors
  - Historically, we have k = 32, 16, 8, and other values

We use five types of basic symbols in the Go programming exercises

	Type	Size	Literals	Values	Operations
Logic →	bool	1 bit	true, false	true, false	&&, ||, !
Character →	byte, uint8	1 byte	63, '?'	[0, 255]	+, -, *, /, %; ++, --; >>, <<; &, |, ^
Integer natural number →	int	8 bytes	-12345, 69	$[-2^{63}, 2^{63}-1]$	
	uint64	8 bytes	12345	$[0, 2^{64}-1]$	
Real number →	float64	8 bytes	3.14159	IEEE 754	+, -, *, /

We use five types of basic symbols in the Go programming exercises

	Type	Size	Literals	Values	Operations
Logic →	bool	1 bit	true, false	true, false	&&, ||, !
Character →	byte, uint8	1 byte 8-bit	63, '?'	[0, 255] **>255**	+, -, *, /, %; ++, --; >>, <<; &, |, ^
Integer natural number	int	8 bytes 64-bit	-12345, 69	$[-2^{63}, 2^{63}-1]$ **$< -2^{63}$; $> 2^{63}-1$**	
	uint64	8 bytes 64-bit	51234	$[0, 2^{64}-1]$ **$> 2^{64}-1$**	
Real number →	float64	8 bytes 64-bit	3.14159	IEEE 754	+, -, *, /

What are the conditions for overflow?

2.2 Binary-decimal conversion

- There is no best method for every one
- Use two examples to demonstrate one method

- Example 1: converting a binary number to decimal number
- $(110.101)_2 = 1\times2^2+1\times2^1+1\times2^{-1}+1\times2^{-3} =$

$$= 4 + 2 + 0.5 + 0.125 =$$

$$= 6.625$$

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
10000.	1000.	100.	10.	1.	0.1	0.01	0.001	0.0001	0.00001
16	8	4	2	1	0.5	0.25	0.125	0.0625	0.03125

Correspondence of binary and decimal bases

Example 2: Decimal to binary conversion

- $6.625 = (110.101)_2$

- Step 1: 6-4=2;
  sufficient, write down 1(2).

6.625

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)							

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
10000.	1000.	100.	10.	1.	0.1	0.01	0.001	0.0001	0.00001
16	8	4	2	1	0.5	0.25	0.125	0.0625	0.03125

$6.625 = (110.101)_2$

- Step 1: 6-4=2;
  sufficient, write down 1(2).
- Step 2: 2-2=0;
  sufficient, write down 1(0).
- Step 3: remainder is 0,
  stop.

6.625

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)							

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)	**1** (0)						

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)	**1** (0)	**0**					

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
10000.	1000.	100.	10.	1.	0.1	0.01	0.001	0.0001	0.00001
16	8	4	2	1	0.5	0.25	0.125	0.0625	0.03125

$6.625 = (110.101)_2$

6.625

- Step 1: 6-4=2; sufficient, write down 1(2).

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)							

- Step 2: 2-2=0; sufficient, write down 1(0).

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)	**1** (0)						

- Step 3: remainder is 0, stop.

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)	**1** (0)	**0**					

- Step 4: 0.625-0.5=0.125; sufficient, write down 1 (.125).

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)	**1** (0)	**0**	**1** (.125)				

- Step 5: 0.125-0.25; insufficient, write down 0 (.125).

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)	**1** (0)	**0**	**1** (.125)	**0** (.125)			

- Step 6: 0.125-0.125=0; sufficient, write down 1 (0).

2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (2)	**1** (0)	**0**	**1** (.125)	**0** (.125)	**1** (0)		

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
10000.	1000.	100.	10.	1.	0.1	0.01	0.001	0.0001	0.00001
16	8	4	2	1	0.5	0.25	0.125	0.0625	0.03125

Decimal to binary conversion could lead to an infinite digit sequence

- $(11.3)_{10} = (?)_2$
  $= 1011.0100\,1100\,1\ldots$

11.3

2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
8	4	2	1	0.5	0.25	0.125	0.0625	0.03125
1 (3)	0 (3)	1 (1)	1 (0)	0 (.3)	1(.05)	0 (.05)	0 (.05)	1 (.01875)

2^{-6}	2^{-7}	2^{-8}	2^{-9}
0.015625	0.0078125	0.00390625	0.001953125
1 (.003125)	0 (.003125)	0 (.003125)	1 (.001171875)

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
10000.	1000.	100.	10.	1.	0.1	0.01	0.001	0.0001	0.00001
16	8	4	2	1	0.5	0.25	0.125	0.0625	0.03125

Binary-decimal-hexadecimal conversion

- Hexadecimal notation is shorter than binary
  - Often used in programs

- 63 = 00111111
  = 0011 1111
  = $3F_{16}$

Binary	Decimal	Hexadecimal
$2^3 2^2 2^1 2^0$	$10^1 10^0$	16^0
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

2.3 Representing integers

- Use unsigned integers to represent natural numbers
  - A n-bit symbol can represent any value in [0, 2^n -1]
  - An 8-bit symbol can represent any value in [0, 2^8 -1], i.e., [0, 255]
- What if negative values are possible?
  - Use the leftmost bit to represent sign: 0 (+) and 1(-)
- A straightforward representation (simple signed integers)
  - Sign bit + 7-bit absolute value
  - However, it may lead to wrong results
    - 63 = 00111111， 64 = 01000000
    - (-63) = 10111111， (-64) = 11000000
    - 63 + 64 = 00111111 + 01000000 = 01111111 = 127 √
    - (-63) + (-64) = 10111111 + 11000000 = 11111111 = (-127) √
    - 63 + (-63) = 00111111 + 10111111 = 11111110 = (-126) X

Two's complement representation

- Examples: 8-bit numbers and additions, [-127,127]
- Sign bit + 7-bit absolute value
  - 63 = 00111111， 64 = 01000000
  - (-63) = 10111111， (-64) = 11000000
  - 63 + 64 = 00111111 + 01000000 = 01111111 = 127 √
  - (-63) + (-64) = 10111111 + 11000000 = 11111111 = (-127) √
  - 63 + (-63) = 00111111 + 10111111 = 11111110 = (-126) X
- A better choice: Two's complement representation
  - Representing a negative number N
    - Bitwise negate absolute(N), and add 00000001
    - (-63) = negate(00111111) + 00000001
      = 11000000 + 00000001 = 11000001
  - Addition: normal addition and ignore overflowing
  - 63 + (-63) = 00111111 + 11000001 = 00000000 = 0 √

Detailed steps of addition

- 63+63 = 00111111+00111111

  00111111

  01111110 = 126

- 63+(-63) = 00111111+11000001

  11000001

  100000000 = 00000000 = 0 (ignore overflowing 1)

- (-63)+(-63)= 11000001+11000001

  11000001

  110000010 = 10000010 = -126 (ignore overflowing 1)

- Exercise: Determine how to do subtraction
  - 63 – 63 = 0
  - 63 – (-63) = 126
  - (-63) – (-63) = 0

2.4 Representing characters

ASCII encodings for "Alan Turing" = [65, 108, 97, 110, 32, 84, 117, 114, 105, 110, 103]

$D_6D_5D_4$ / $D_3D_2D_1D_0$	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	ENQ	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

Control Characters

Printable Characters

$D_3D_2D_1D_0$ \ $D_6D_5D_4$	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	ENQ	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

ASCII control characters (code 0-31)

Decimal	Hexadecimal	Symbol	Description	Chinese
0	0x00	NUL	Null Character	空字符
1	0x01	SOH	Start of Heading	标题开始
2	0x02	STX	Start of Text	正文开始
3	0x03	ETX	End of Text	正文结束
4	0x04	EOT	End of Transmission	传输结束
5	0x05	ENQ	Enquiry	请求
6	0x06	ACK	Acknowledgment	收到通知
7	0x07	BEL	Bell	响铃
8	0x08	BS	Back Space	退格
9	0x09	HT	Horizontal Tab	水平制表符
10	0x0A	LF, NL	Line Feed, New Line	换行键
11	0x0B	VT	Vertical Tab	垂直制表符
12	0x0C	FF, NP	Form Feed, New Page	换页键
13	0x0D	CR	Carriage Return	回车键
14	0x0E	SO	Shift Out	不用切换
15	0x0F	SI	Shift In	启用切换
16	0x10	DLE	Data Line Escape	数据链路转义
17	0x11	DC1	Device Control 1	设备控制1
18	0x12	DC2	Device Control 2	设备控制2
19	0x13	DC3	Device Control 3	设备控制3
20	0x14	DC4	Device Control 4	设备控制4
21	0x15	NAK	Negative Acknowledgement	拒绝接收
22	0x16	SYN	Synchronous Idle	同步空闲
23	0x17	ETB	End of Transmit Block	结束传输块
24	0x18	CAN	Cancel	取消
25	0x19	EM	End of Medium	媒介结束
26	0x1A	SUB	Substitute	代替
27	0x1B	ESC	Escape	换码(溢出)
28	0x1C	FS	File Separator	文件分隔符
29	0x1D	GS	Group Separator	分组符
30	0x1E	RS	Record Separator	记录分隔符
31	0x1F	US	Unit Separator	单元分隔符
127	0x7F	DEL	Delete	删除

ASCII printable characters (code 32-127)

Decimal	Hexadecimal	Symbol	Description	Chinese
32	0x20	SP	Space	空格
33	0x21	!	Exclamation mark	叹号
34	0x22	"	Double quotes	双引号
35	0x23	#	Number, sharp	井号
36	0x24	$	Dollar sign	美元符
37	0x25	%	Percent sign	百分号
38	0x26	&	Ampersand	和号
39	0x27	'	Single quote	闭单引号
40	0x28	(	Open parenthesis (or open bracket)	开括号
41	0x29	)	Close parenthesis (or close bracket)	闭括号
42	0x2A	*	Asterisk, multiply	星号
43	0x2B	+	Plus	加号
44	0x2C	,	Comma	逗号
45	0x2D	-	Hyphen	减号/破折号
46	0x2E	.	Period, dot	句号
47	0x2F	/	Slash, divide	斜杠
48	0x30	0	Zero	字符0
49	0x31	1	One	字符1
57	0x39	9	Nine	字符9
58	0x3A	:	Colon	冒号
59	0x3B	;	Semicolon	分号
60	0x3C	<	Less than (or open angled bracket)	小于
61	0x3D	=	Equals	等号
62	0x3E	>	Greater than (or close angled bracket)	大于
63	0x3F	?	Question mark	问号
64	0x40	@	At symbol	电子邮件符号

Decimal	Hexadecimal	Symbol	Description	Chinese
65	0x41	A	Uppercase A	大写字母A
66	0x42	B	Uppercase B	大写字母B
90	0x5A	Z	Uppercase Z	大写字母Z
91	0x5B	[	Opening bracket	开方括号
92	0x5C	\	Backslash	反斜杠
93	0x5D	]	Closing bracket	闭方括号
94	0x5E	^	Caret	脱字符
95	0x5F	_	Underscore	下划线
96	0x60	`	Grave accent	开单引号
97	0x61	a	Lowercase a	小写字母a
98	0x62	b	Lowercase b	小写字母b
122	0x7A	z	Lowercase z	小写字母z
123	0x7B	{	Opening brace	开花括号
124	0x7C	|	Vertical bar	垂线
125	0x7D	}	Closing brace	闭花括号
126	0x7E	~	Tilde	波浪号

ASCII encodings of English characters

SP= 00100000_2 $=32_{10}$

SP is the ASCII character

00100000 or 32 is its ASCII encoding or ASCII value or ASCII code

D_7 is 0

$D_6D_5D_4$ / $D_3D_2D_1D_0$	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	ENQ	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

Representing ASCII characters

SP= 00100000_2 $=32_{10}$

SP is the ASCII character

00100000 or 32 Is its ASCII encoding or ASCII value

D_7 is 0

$D_3D_2D_1D_0$ \ $D_6D_5D_4$	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	ENQ	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

What is the ASCII value of +

‘+’ 00101011_2 43_{10}

Representing ASCII characters

$D_3D_2D_1D_0$ \ $D_6D_5D_4$	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	ENQ	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

‘+’
00101011_2
43_{10}

Esc
00011011_2
27_{10}

Representing ASCII characters

$D_3D_2D_1D_0$ \ $D_6D_5D_4$	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	ENQ	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

‘+’
00101011_2
43_{10}

Esc
00011011_2
27_{10}

NUL
00000000_2
0_{10}

The three symbols “of emptiness” NUL, SP, 0 sometimes cause confusion

SP = 00100000_2 = 32_{10}

SP is the ASCII character

00100000 or 32 Is its ASCII encoding or ASCII value

$D_3D_2D_1D_0$ \ $D_6D_5D_4$	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	ENQ	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

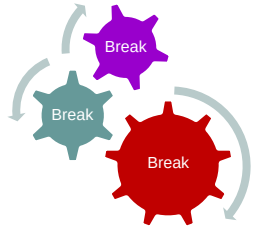
NUL 00000000_2 0_{10}

‘0’ 00110000_2 48_{10}

How to represent the world's text?

- ASCII uses 8 bits to encode English characters
  - 8 bits = 1 byte
- The world's text contains over 100000 characters spanning more than 157 languages
  - Over 70000 characters in Chinese alone
- How to encode all these characters?
  - How many bits should be used?
    - Should the answer be 17 bits, because $2^{17} > 100000$?
    - What happens in reality?
- Unicode!
  - To be discussed

Take-Home Messages



- How to do binary-decimal-hex conversion?
- How to represent numbers?
  - Use unsigned integers to represent natural numbers
  - Use two's complement to represent integers
  - Use floating-point numbers to represent real numbers
- How to represent characters?
  - Use ASCII to represent English characters
  - Use Unicode to represent the world's text
- The same symbol can have different representations
  - E.g., the ASCII code of character '?' Is represented as
    - Binary 00111111
    - Decimal 63
    - Hexadecimal 3F

Quiz: define the following terms

- A bit is_____.
- A digital symbol is _____.
- Manipulation is_____.
- Data are_____.
- A program is _____.
- Code is_____.

Quiz: define the following terms

- A bit is a binary digit with a value of 0 or 1 .
- A digital symbol is any notation that is representable as one or more bits, to denote any concrete or abstract entity.
- Manipulation is a sequence of operation steps on digital symbols, where the length can be one or many.
- Data are a group of bits .
- A program is an algorithm written in a computer language .
- Code is a fraction (segment) of a program .