BU CodeBreakers 2017

Digital System Representations: Decimal, Binary, ASCII

Today's Schedule

Find your nametags

09:00-10:15   Digital system representation

10:30 - 11:45   Worksheet 1

12:00 - 01:45   Review + Exercise IV/V

02:00 - 03:00   Exercise IV/V

Guest Speaker tomorrow morning 9am (Room: B18)

Review

```
>>> for i in [1, 3, 5, 7]:
        print (i)

>>> for c in 'CodeBreakers':
        print (c)

>>> i = 0
>>> while i<10:
        print (i)
        i = i+1

>>> range(9, -1, -1)
```
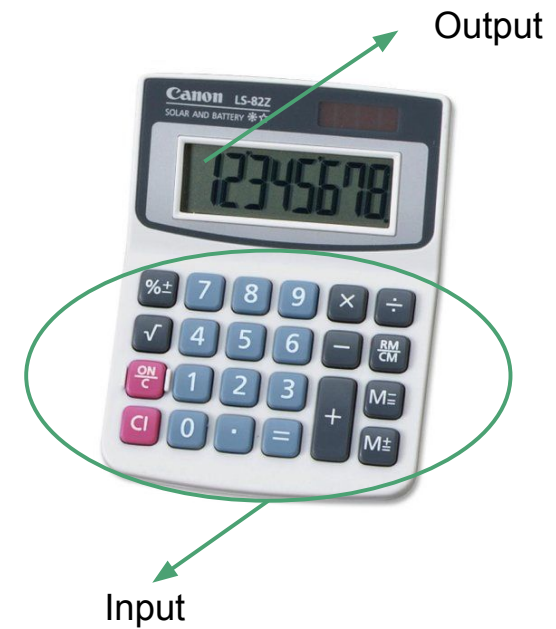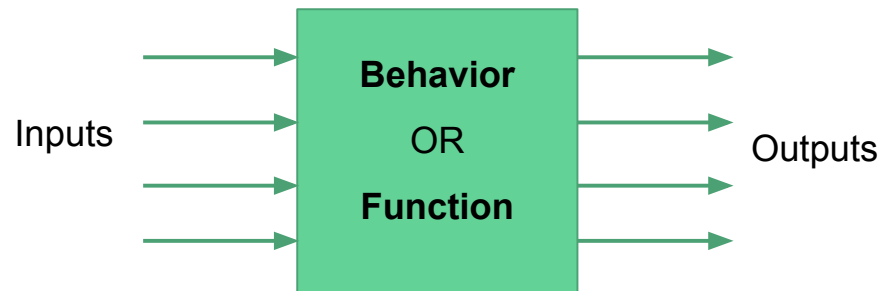
System

- Set of components that work together to achieve a certain task

Inputs

Behavior

OR

Function

Outputs

Output

Input

Digital System

- Modern computing system are digital i.e. everything is represented by **digits**

- How do I represent text "Codebreakers" with digits/numbers?

Solution I : Sequencing with integers [a ➜ 0, b ➜ 1, c ➜ 2, ..., z ➜ 25]

C: 2,   **o**: 14,   **d**: 3,   **e**: 4,

b: 1,   **r**: 17,   **e**: 4,   **a**: 0,   **k**: 10,   **e**: 4,   **r**: 17,   **s**: 18

- Modern computing systems:

    physical signals [temperature, electricity] ➜ digital conversion ➜ COMPUTE!
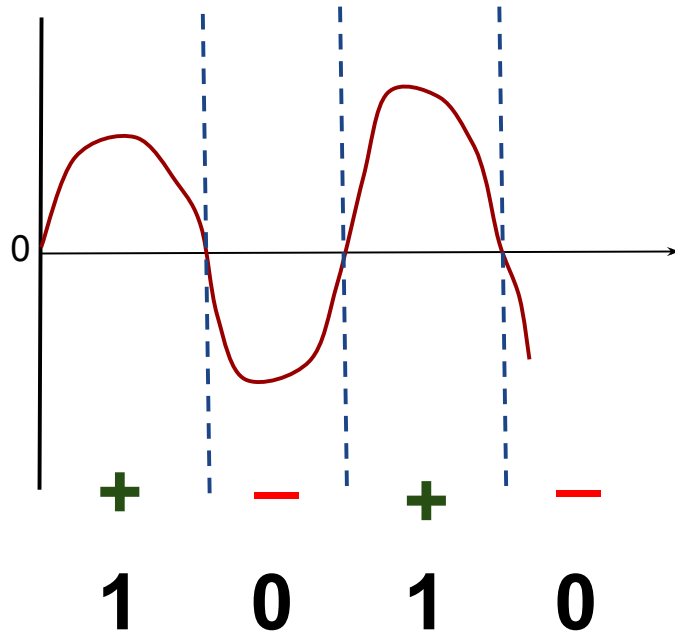
Bit & Byte

- A bit is the smallest unit of information

  - It represents one 2-way decision or a 2-way choice

  - yes / no, true / false, on / off ...

- Typical value: either 0 or 1

- All information in a computer is **stored** and **processed** as bits

- A byte is 8 bits that are treated as a unit

| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | → Byte |

Bits
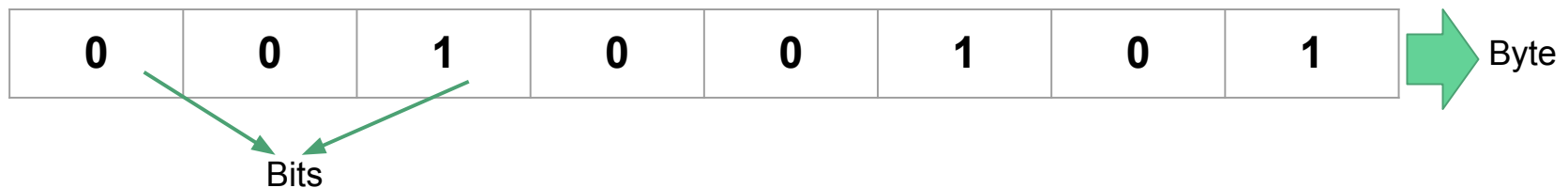
Why Binary?

- Simplicity

+ − + −

1   0   1   0

4-bit binary representation

Now we can pretty much represent anything by:

1. Real-world signals ➔ Binary
2. Binary ➔ Decimal Numbers?
3. Binary ➔ Characters?

Decimal Numbers Review

- How many digits are there?
  - Ten: 0, 1, ..., 9
- **42** ➡   **4** tens + **2** ones
- **234** ➡   **2** hundreds + **3** tens + **4** ones
- **1234?**

  1 thousands   + **2** hundreds + **3** tens   + **4** ones

  1 * 1000   + **2** * 100   + **3** * 10   + **4** * 1

  1 * 10^3   + **2** * 10^2   + **3** * 10^1   + **4** * 10^0

- **Decimal numbers are sums of powers of 10**

Binary Number System

- Uses bits to represent any number

- Only uses 2 digits: 0, 1

- Recall: Decimal numbers are sums of powers of 10. *What about binary numbers?*

- Binary numbers are sums of powers of 2

- **5** ➜ 4 + 1   ➜ **1** four + **0** two + **1** one

- **22** ➜ 16 + 4 + 2   ➜ **1** sixteen   + **0** eight   + **1** four   + **1** two   + **0** one

                                        ➜ **1** * 2^4   + **0** * 2^3   + **1** * 2^2 + **1** * 2^1 + **0** * 2^0

⇒ **22** decimal   ➜ **1 0 1 1 0** binary

Decimal ➜ Binary

1. Start with a number (n)
2. Divide n by 2
3. Remainders: 0/1 ➜ append to Binary number
4. Quotient ➜ new n
5. Continue from Step 2 until quotient becomes zero
6. Final result is the reverse of your current Binary number

Decimal ➜ Binary

- Convert n = 97 to binary

---

n ÷ 2	quotient	remainder

---

97 ÷ 2	48	1
48 ÷ 2	24	0
24 ÷ 2	12	0
12 ÷ 2	6	0
6 ÷ 2	3	0
3 ÷ 2	1	1
1 ÷ 2	0 (Stop)	1

Least Significant Bit (LSB)

Binary number: 1 1 0 0 0 0 1

Most Significant Bit (MSB)
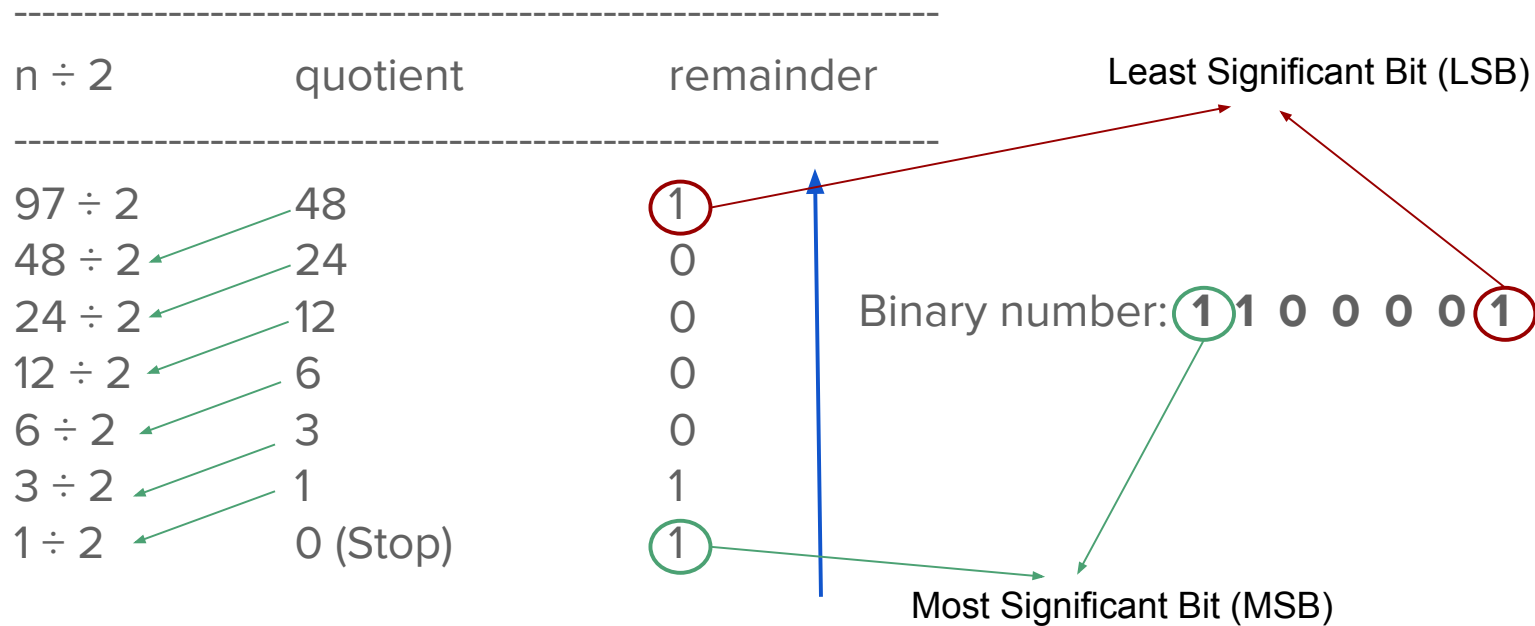
Binary ➜ Decimal

Convert from Binary to Decimal

←—————————————

(MSB)   (LSB)

1   **0**   **1**   **1**

1 * 2^3 + **0** * 2^2 + **1** * 2^1 + **1** * 2^0

1 * 8 + **0** * 4 + **1** * 2^1 + **1** * 1

8 + 0 + 2 + 1

11
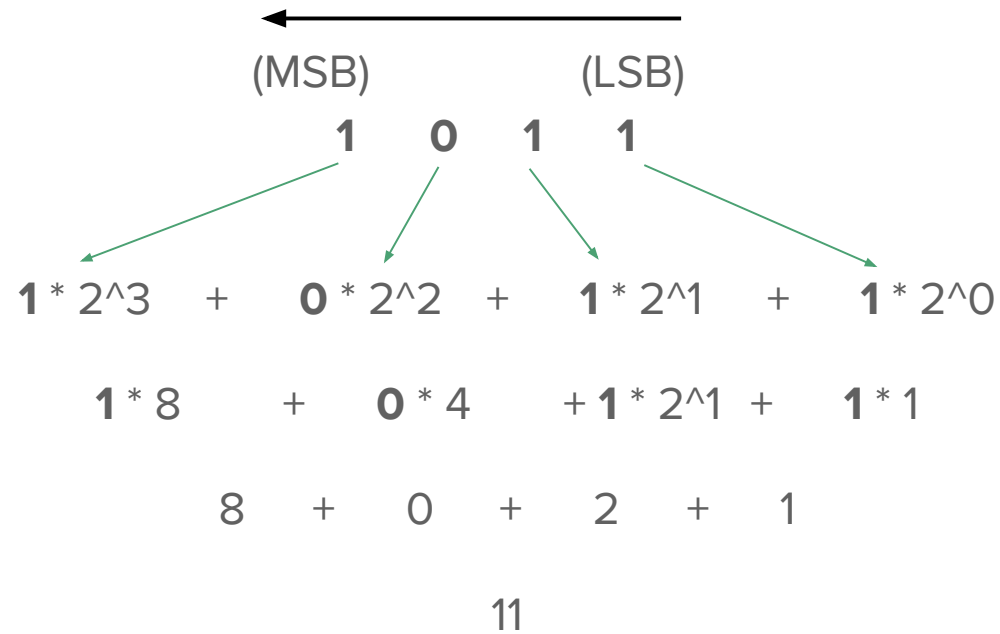
Bitwise Operations

AND (**&**)

```
0   0   1   1
1   0   1   0
----------------------
0   0   1   0
```

OR (I)

```
0   0   1   1
1   0   1   0
----------------------
1   0   1   1
```

XOR (^)

```
0   0   1   1
1   0   1   0
----------------------
1   0   0   1
```

NOT (~)

```
0   1
-----------
1   0
```

&, I, ^ and ~ are bit-wise operators in Python

Summary

Binary ➜ Language of Computers

Decimal ➜ Language of Humans

- Base conversions to communicate back and forth with computers
  - Decimal ➜ Binary
  - Binary ➜ Decimal

What about characters?

Recall

- Digital Systems

- How do I represent text "Codebreakers" with digits/numbers?

Solution I : Sequencing with integers [a➜ 0, b➜ 1, c➜ 2, ..., z➜ 25]

C: 2,   **o**: 14,   **d**: 3,   **e**: 4,

b: 1,   **r**: 17,   **e**: 4,   **a**: 0,   **k**: 10,   **e**: 4,   **r**: 17,   **s**: 18
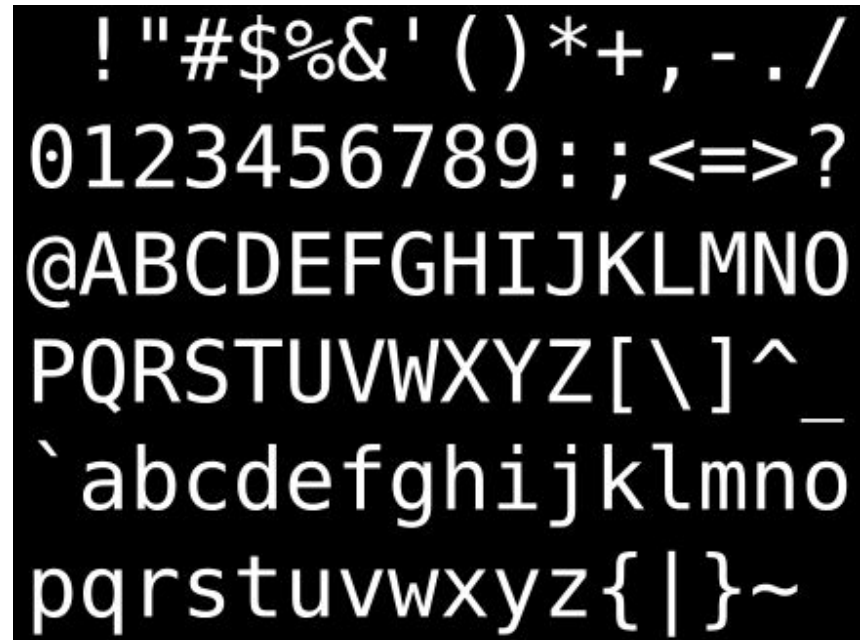
How to represent characters via digits?

ASCII

- American Standard Code for Information Interchange
- Number representation of a character
- Arbitrary but agreed upon representation in US and rest of the world

Character ➜ ASCII

Note that UPPER and lower letters are considered to be different characters

What's the ASCII representation of

H E L L O ?

72 69 76 76 79 63

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124		
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

ASCII ➜ Character

Note that UPPER and lower letters are considered to be different characters

What does this spell?

72 69 76 76 79 33

H E L L O !

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124		
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

Character <=> Binary?

Character ➤ Decimal (ASCII) ➤ Binary

Binary ➤ Decimal (ASCII) ➤ Character

WORKSHEET I
EXERCISE V/IV