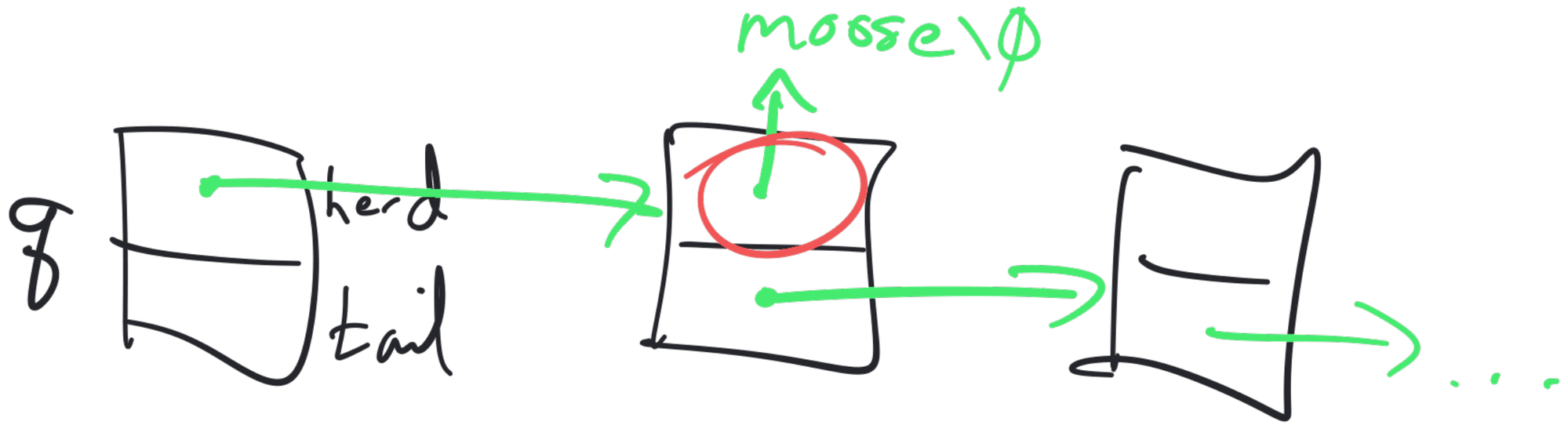
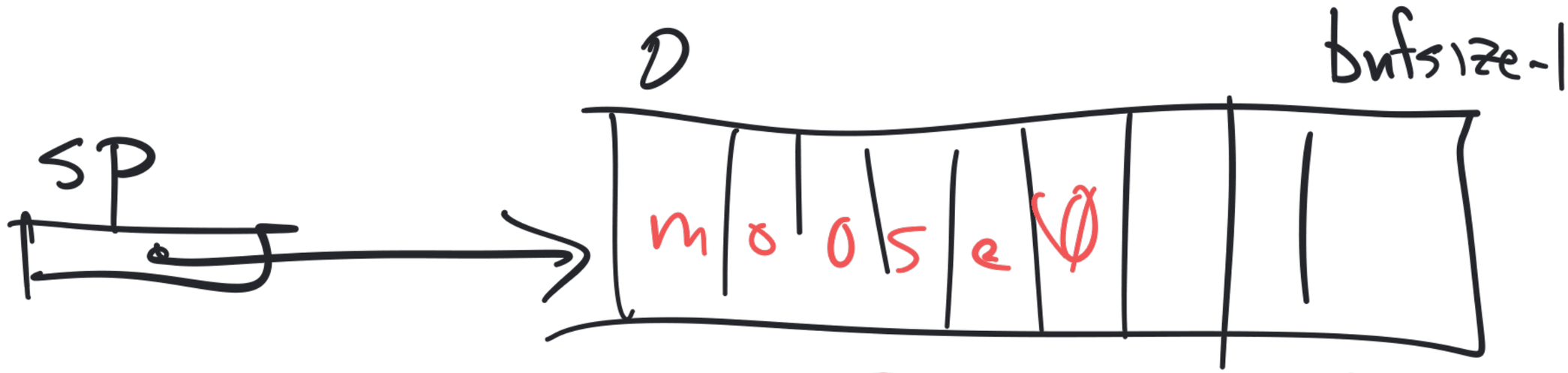
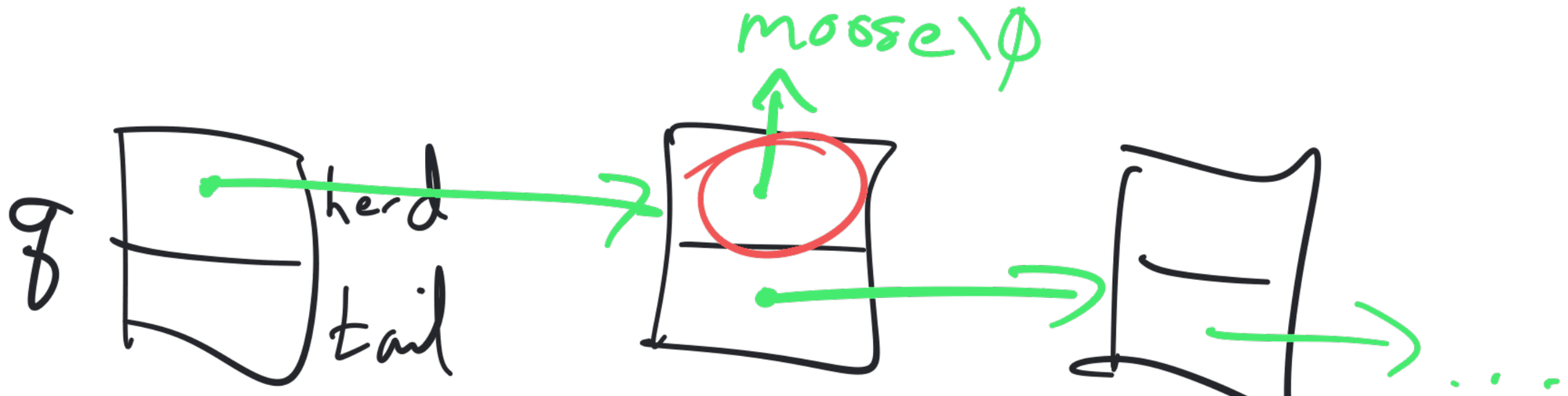
CS 208

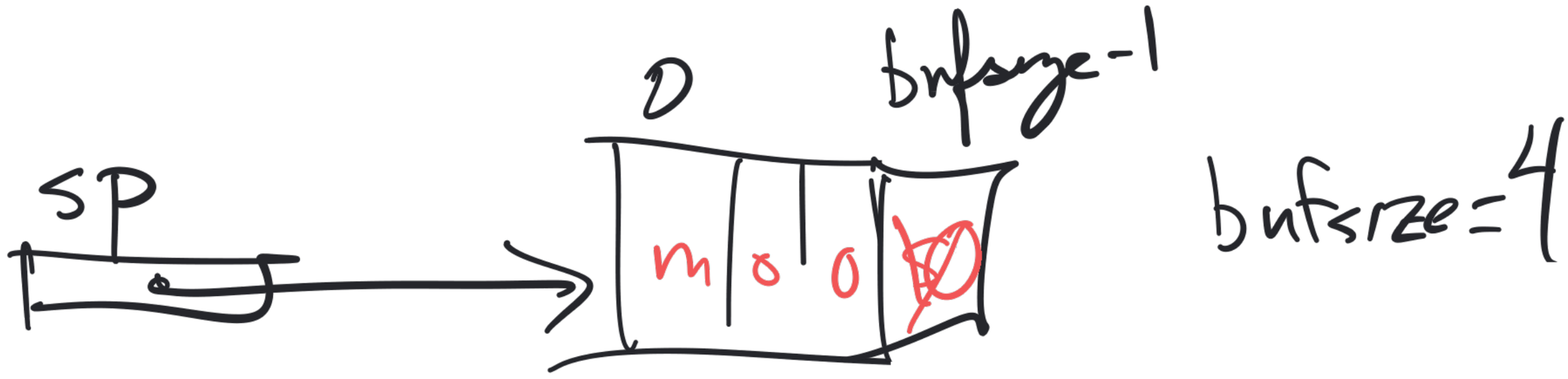
Fri 13 Jan 2023





strncpy(sp, [illegible], bufsize)






```
strncpy(sp, ____, bufsize)
sp[bufsize-1] = '\0';
```

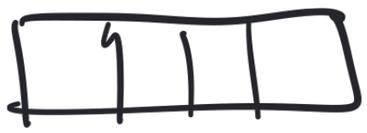
Negative integers

Assume fixed-size integers

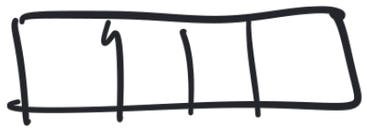
eg. 32 bits (4 bytes)

For now, pretend

4-bit ints



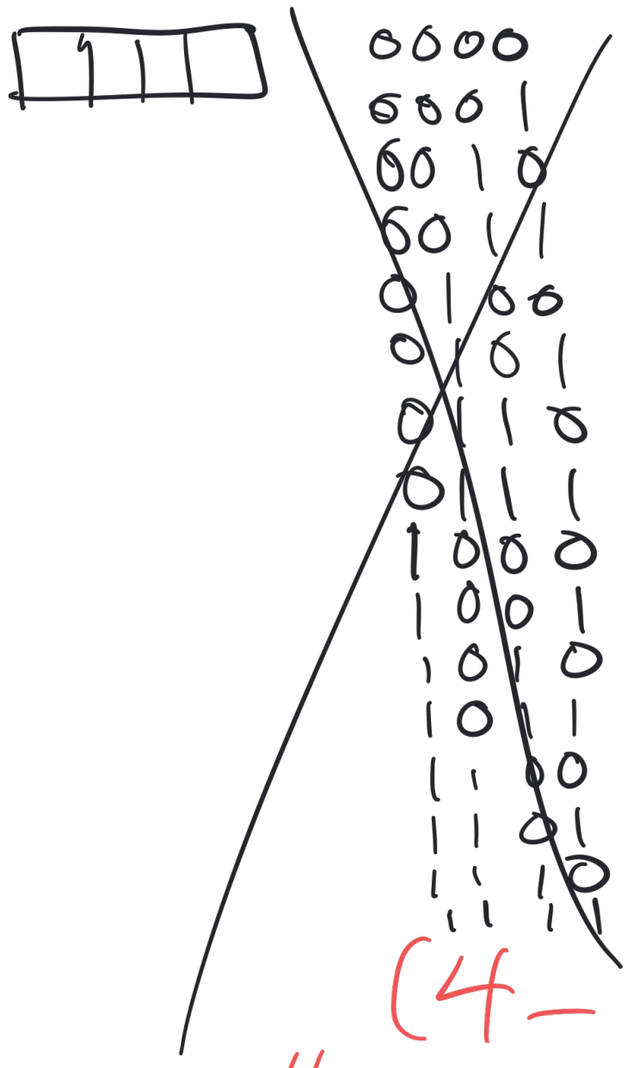
	Interp#1	Interp#2
0000	0	Truth
0001	1	Reality
0010	2	Netflix
0011	3	Ursula Leguin
0100		
0101		
0110		
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111	15	Mars



0000	0
0001	1
0010	2
0011	.
0100	.
0101	.
0110	
0111	7
1000	-0
1001	-1
1010	-2
1011	.
1100	.
1101	.
1110	
1111	-7

"sign bit"

"signed magnitude representation of integers"



0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

(4-bit)

"two's complement representation"

$4 + -6 = -2$

$0100 + 1010 = 1110$

$6 + -3 = 3$

$0110 + 1101 = 0011$

Two's complement

- Given a base ten int, represent it in 2's comp.
- Given a 2's comp int N, show 2's Comp. of $-N$ ("Negate N")
- Given 2's comp N, translate to decimal

Given 4-bit 1110 (-2)

Negate

1110

Step1 ↓ complement

0001

Step2 ↓ +1

0010

-2

+1

1101

complement

+2

16-bit two's comp.

-34_{ten}

① Represent 34

0000 0000 0010 0010

② Negate

1111 1111 1101 1101

+ 1

1111 1111 1101 1110 ← -34