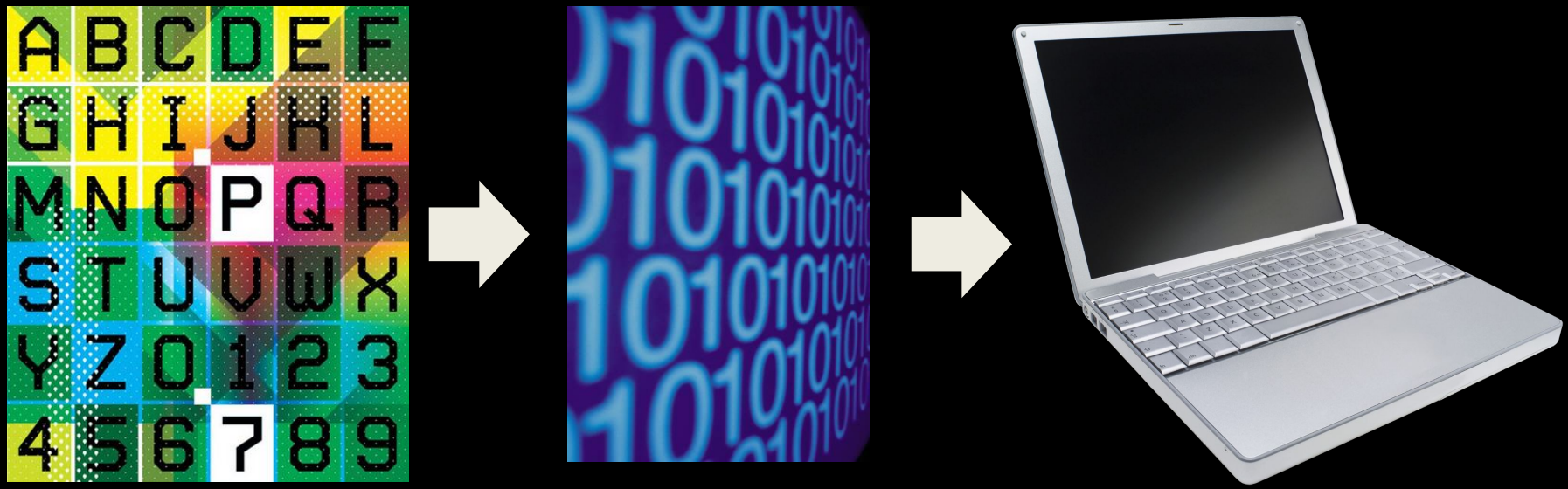
Characters must also be encoded in binary



ASCII maps characters to numbers

INT	CHAR		INT	CHAR	INT	CHAR	INT	CHAR
0	NUL	(null)	32	SPACE	64	@	96	`
1	SOH	(start of heading)	33	!	65	A	97	a
2	STX	(start of text)	34	"	66	B	98	b
3	ETX	(end of text)	35	#	67	C	99	c
4	EOT	(end of transmission)	36	$	68	D	100	d
5	ENQ	(enquiry)	37	%	69	E	101	e
6	ACK	(acknowledge)	38	&	70	F	102	f
7	BEL	(bell)	39	'	71	G	103	g
8	BS	(backspace)	40	(	72	H	104	h
9	HT	(horizontal tab)	41	)	73	I	105	i
10	LF	(line feed)	42	*	74	J	106	j
11	VT	(vertical tab)	43	+	75	K	107	k
12	FF	(form feed)	44	,	76	L	108	l
13	CR	(carriage return)	45	-	77	M	109	m
14	SO	(shift out)	46	.	78	N	110	n
15	SI	(shift in)	47	/	79	O	111	o
16	DLE	(data link escape)	48	0	80	P	112	p
17	DC1	(device control 1)	49	1	81	Q	113	q
18	DC2	(device control 2)	50	2	82	R	114	r
19	DC3	(device control 3)	51	3	83	S	115	s
20	DC4	(device control 4)	52	4	84	T	116	t
21	NAK	(negative acknowledge)	53	5	85	U	117	u
22	SYN	(synchronous idle)	54	6	86	V	118	v
23	ETB	(end of transmission block)	55	7	87	W	119	w
24	CAN	(cancel)	56	8	88	X	120	x
25	EM	(end of medium)	57	9	89	Y	121	y
26	SUB	(substitute)	58	:	90	Z	122	z
27	ESC	(escape)	59	;	91	[	123	{
28	FS	(file separator)	60	<	92	\	124	|
29	GS	(group separator)	61	=	93	]	125	}
30	RS	(record separator)	62	>	94	^	126	~
31	US	(unit separator)	63	?	95	_	127	DEL

ASCII Math

What will print?

```
printf("%d\n", 'a' - 'A');
printf("%c\n", 'B' + ('a' - 'A'));
printf("%c\n", 'b' - ('a' - 'A'));
printf("%c\n", 'B' + 1);
printf("%c\n", ('z' - 'a' + 1) % 26 + 'a');
```

Example #1
Prints Z through A

```
for (int i = 'Z'; i >= 'A'; i--)
    printf("%c\n", i);
```

Example #2
Converts a lowercase string to uppercase

```
char name[] = "milo";
for (int i = 0, j = strlen(name); i < j; i++)
    name[i] = name[i] + ('A' - 'a');
```