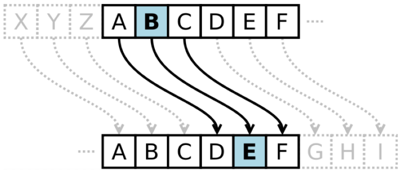
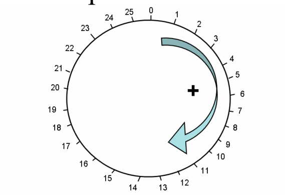
CS125 Section 6 *" Your secret is safe me! You can trust me... I'm a computer scientist!"*

Q1 Caesar Cipher Review!
Here's a pictorial explanation of Caesar Cipher[1].





An example Caesar Cipher code snippet is below. The expression at line 6 looks complicated. We will break it down into its individual ideas.

```
1.  String s = "thepersonwhosaysitcannotbedoneshouldnotinterruptthepersondoingit";
2.  char[]  mesg = s.toCharArray();
3.  String result = "";
4.  int shift = 3;
5.  for(int i = 0; i < mesg.length; i++)
6.     result += (char) ('a' +  (mesg[i] - 'a' + shift) % 26 );
```

Circle and label (1, 2, 3, 4) the exact part of line 6's expression that ...
1. Transforms the current letters 'A' to 'Z' to integers 0 to 25

2. Performs modulo 26 arithmetic (e.g. 27 mod 26 is 1)

3. Performs the shift right by 3.

4. Transforms the integers 0 to 25 back to characters 'A' to 'Z'

Now we'll look at some alternative implementations of line 6:

1. In this example (see below), which letters in *mesg* would be encoded incorrectly? Why?

```
result += (char) ('a' +  (mesg[i] - 'a' + shift) );
```

2. Here' another implementation. Why is the result now "1191071041151041171118114...?

```
result += ('a' +  (mesg[i] - 'a' + shift) );
```

3. In the original program, explain why using shift=-1 and shift=25 produce different results:
sgdodqrnmvgnr`xrhsb`mmnsadcnmdrgntkcmnshmsdqqtossgdodqrnmcnhmfhs *shift = -1*
sgdodqrnmvgnrzxrhsbzmmnsadcnmdrgntkcmnshmsdqqtossgdodqrnmcnhmfhs *shift = 25*

4. How would you extend the program to pass spaces and punctuation unchanged to *result*?

[1] Image attribution:
Cipher map: http://en.wikipedia.org/wiki/Caesar_cipher
mod26 circle: http://www.cs.virginia.edu/~evans/dragoncrypto/day1.html

Q2. Convert the following to use while loops. How many *'s will be printed?

```
int a = 0;
int c = 0;
for(int x = 15; x > 0; x--) {
   for(int y = 1; y < x; y++)
      TextIO.put('*');
   x = x/2;
}
```

Q3. Complete the following program. **Do not create additional variables or print the same letter in two different places in your code. Ask someone else to look for errors once you have a solution.**

```
/** Print "FEDEX" , "UPS", "DHL" or "PRIVATE" ("F" "U" "D" or "P") according to the following rules:
  * Domestic Non-urgent packages under 25 lbs are shipped UPS.
  * Domestic Non-urgent packages 82 lbs or greater are shipped by DHL courier
  * Urgent packages are always shipped using FEDEX
  * All other packages are shipped using PRIVATE courier */
int weight = TextIO.getlnInt();
char urgent = TextIO.getlnChar(); // either 'Y' or 'N'
boolean international = TextIO.getlnBoolean();
```

Q4. Assembly Program Review

What is the following assembly program doing (mathematically)? Find it by determining the output.

r0		a0	3	a100	zero_reg r1	
r1		a4	18	a102	zero_reg r2	
r2		a8	1	a104	add r2 + 1 -> r2	
r3		a12	5	a106	sub r2 - 10 -> r3	
r4		a16	42	a108	br.p 5	#jumps to 120
r5		a20	9	a110	load r4 <- [r1 + 0]	
r6		a24	0	a112	add r4 + r2 -> r4	
r7		a28	74	a114	store r4 -> [r1 + 0]	
		a32	3	a116	add r1 + 4 -> r1	
PC	100	a36	36	a118	br.pnz -8	#jumps to 104
				a120	//done	

N Z P

Q5. Complete the following program to print out in lowercase the first word after the first period on each line. Stop processing after 3 words have been printed. If there is no period on the line ignore the entire line. A word is defined as the characters 'a'-'z'; consider all other characters including digits and punctuation as non-letters. For the following input, you code would print "*you go now*". Use TextIO.getln() to read each line.

Stop and think. You can learn
(I think)
to type well. Go to any
keyboard. Now and then. I
do not practice but today
it will be great!

```
class Extract {
 public static void main(String[] args) {
    TextIO.readFile("textFile.txt");
```