Lab: Modular Arithmetic and Cryptography

In this lab we will practice modular arithmetic and encoding/decoding messages using Octave, which you can access at octave-online.net.

To complete the entire lab, you will need to sign in to Octave Online with Google or by using an email address. Once you sign in (under the Menu button), you will need to upload the files `numbers.m` and `letters.m`. The button to upload files is on the left-hand side of the screen. Once uploaded, you will see the list of files, and by clicking on each file name individually you can read its contents.

1. Try these exercises from the Modular Arithmetic lecture in Octave. To compute $a \pmod{n}$ in Octave type `mod(a,n)`. For example, to solve the first exercise you can type `mod(3+8,9)`.

   - $3 + 8 \mod 9$
   - $21 + 20 \mod 35$
   - $5 - 9 \mod 11$
   - $5 \cdot 5 \mod 24$
   - $2 \cdot 3 \mod 6$
   - $7 \cdot 3 \cdot 5 \mod 10$
   - $2^3 \mod 5$
   - $3^6 \mod 8$
   - $(5 + 7)^3 \mod 10$

2. In the Cryptography lecture we saw we can encode messages by shifting each letter in our message a fixed number of letters down the alphabet. We also can do this quickly by translating our letters to numbers and working modulo 26.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

   Let's try encoding "MATH" using a shift of 10. We can do this by hand, but computers can make this much easier.

- From our chart M=12, A=0, T=19, H=7. We can shift these in Octave, try entering the following (hit enter between each).
  ```
  12+10
  0+10
  19+10
  7+10
  ```

- We have some numbers that are out of our range of 0-25, so we need to "loop back around". Compute the value modulo 26 using the syntax you learned above.
- once we have all the numbers in the range 0-25 we can use the chart to convert back to letters. You should get the encoded message:"WKDR".

3. In the last exercise, we really just used Octave to do something we could do by hand or using a calculator, now we will learn how we can use Octave to do this more quickly using arrays (row vectors) and the scripts letters.m and number.m you imported earlier.

   - First input our message in Octave (with the name plaintext) by typing
     ```
     plaintext='MATH'
     ```
     (It is very important to use capital letters and the single quotation marks here.)
   - Now we need to switch our letters to numbers. We do this and give this data the name plaintextnum by typing
     ```
     plaintextnum = numbers(plaintext)
     ```
     Note, our plaintextnum is a list of the numbers representing our original messge.
   - Now it is time to encode the message. We need to add 10 to each number and then reduce mod 26. We can do this is two steps:
     ```
     shifted = plaintextnum+10
     ciphertextnum = mod(shifted, 26)
     ```
     or all in one step like this:
     ```
     ciphertextnum = mod(plaintextnum+10, 26).
     ```
   - Finally, we need to switch back to letters by typing
     ```
     letters(ciphertextnum)
     ```

4. Try it yourself! Use the commands you learned above to encrypt the text "GIRLS GET MATH" with a shift of 19.

    - We can't deal with the spaces so we just leave those out. Start by typing
      ```
      plaintextnum=numbers('GIRLSGETMATH')
      ```
    - Fill out the following table as you go along. Remember to refer back the last exercise for the syntax.

Plaintext in numbers:	
Added and reduced:	
Ciphertext:	

5. Now let's try decoding a message. Caesar sent the following message to his generals: "QHHG PRUH VDODG GUHVVLQJ". The message was encrypted with a shift of 3. Can you figure out what it says?

    - First try it by hand in the blanks provided.

Ciphertext:	
Ciphertext in numbers:	
Subtract 3 :	
Reduce mod 26 :	
Plaintext:	

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

    - Now try it all in Octave! Check that your answer matches the one you found above.

6. Challenge Problem: (Eavesdropping) Suppose you intercept the following ciphertext that you know was encoded using a Caesar cipher. Can you decode the message?

   "LRO ZLAB EXP YBBK YOLHBK"

7. Diffie-Hellman: Suppose you are Alice, you and your friend Barb want to share a secret key. You agree to use Diffie-Hellman Key Exchange with the publicly shared prime $p = 263$ and base $g = 5$.

   Barb sends you the number $B = 45$. You choose the secret integer $a = 52$ (which you don't share with anyone).

   - Fill out the table below with the information you have so far.

Alice	Eve	Barb
$a =$ _____	$p =$ _____, $g =$ _____	$b =$ _____
$A \equiv g^a \mod p$	$A =$ _____, $B =$ _____	$B \equiv g^b \mod p$
$s \equiv B^a \mod p \equiv$ _____	???	$s \equiv A^b \mod p$

   - Store these as variables in Octave. For example type `p=263`.
   - Use Octave to calculate $A \equiv g^a \mod p$: we can't just use the `mod` command we learned earlier because g^a is too big for Octave to store properly. Instead to compute $g^a \mod p$ we use

     ```
     powermod(g,a,p)
     ```

   - Use Octave to raise Barb's number B to the a-th power to get your shared secret s. (Note you don't need to know Barb's number b to find s!)
   - Check that Barb will find the same secret number as you: Barb's secret integer was $b = 101$. Use Barb's secret integer and your number A to check that you get the same shared secret.