Algorithms: Day Three

Real world examples: Transforming data, codes, and ciphers

These examples will use:

- Looping
- Transforming
- Conditionals

First Example: Substitution Cipher

- Transforms one input into another
- Attempts to make it difficult to decode the "ciphertext" back into "plaintext"

Simple substitution

- Consider only the uppercase letters A through Z
- For each letter, we have another unique letter to transform into

S D G L I F E A B C H J K M N O P Q R T U V W X Y Z

Z D R T S E P O G A V K C H W F Q X N B L Y U I M J

Take an example.
Encode the phrase
CODES

```
S D G L I F E A B C H J K M N O P Q R T U V W X Y Z
                  |
                  v
Z D R T S E P O G A V K C H W F Q X N B L Y U I M J
```

CODES

A

```
S D G L I F E A B C H J K M N O P Q R T U V W X Y Z
                              |
                              v
Z D R T S E P O G A V K C H W F Q X N B L Y U I M J
```

CODES

AF

```
S D G L I F E A B C H J K M N O P Q R T U V W X Y Z
  |
  v
Z D R T S E P O G A V K C H W F Q X N B L Y U I M J
```

CODES

AFD

```
S D G L I F E A B C H J K M N O P Q R T U V W X Y Z
            |
            v
Z D R T S E P O G A V K C H W F Q X N B L Y U I M J
```

CODES

AFDP

```
S D G L I F E A B C H J K M N O P Q R T U V W X Y Z
|
v
Z D R T S E P O G A V K C H W F Q X N B L Y U I M J
```

CODES

AFDPZ

What is our algorithm?

What is our algorithm?

1. Start at the first letter of our plaintext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext
6. ...

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext
6. ...
7. If the letter is "X" put a "I" in the ciphertext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext
6. ...
7. If the letter is "X" put a "I" in the ciphertext
8. If the letter is "Y" put a "M" in the ciphertext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext
6. ...
7. If the letter is "X" put a "I" in the ciphertext
8. If the letter is "Y" put a "M" in the ciphertext
9. If the letter is "Z" put a "J" in the ciphertext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext
6. ...
7. If the letter is "X" put a "I" in the ciphertext
8. If the letter is "Y" put a "M" in the ciphertext
9. If the letter is "Z" put a "J" in the ciphertext
10. If that was the last letter in the plaintext: STOP

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext
6. ...
7. If the letter is "X" put a "I" in the ciphertext
8. If the letter is "Y" put a "M" in the ciphertext
9. If the letter is "Z" put a "J" in the ciphertext
10. If that was the last letter in the plaintext: STOP
11. Go to the next letter of our plaintext

What is our algorithm?

1. Start at the first letter of our plaintext
2. If the letter is "S" put a "Z" in the ciphertext
3. If the letter is "D" put a "D" in the ciphertext
4. If the letter is "G" put a "R" in the ciphertext
5. If the letter is "L" put a "T" in the ciphertext
6. ...
7. If the letter is "X" put a "I" in the ciphertext
8. If the letter is "Y" put a "M" in the ciphertext
9. If the letter is "Z" put a "J" in the ciphertext
10. If that was the last letter in the plaintext: STOP
11. Go to the next letter of our plaintext
12. Go back to step 2

We'll learn more efficient ways of doing things other than a bunch of "if"s

What do you think the algorithm is for decoding?

Caesar cipher

This is a "shift" cipher

Takes each letter and "shifts" it down the alphabet, circling back to the start if we go off the end.

A .. Z, with a shift number

Take the example of a shift of 3.

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

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

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

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

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
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	A

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

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 A

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

C D E F G H I J K L M N O P Q R S T U V W X Y Z A B

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

C D E F G H I J K L M N O P Q R S T U V W X Y Z A B

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
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Encoding the phrase CODES

It works just like when using a substitution cipher.

What is the ciphertext?

Is there another way?

Can we use the position information?

What if there was a shift of 0?

```
1. Start with the first letter in the plain text
2. For the given letter, figure out the index in the alphabet (e.g., A is 0, B is 1, ..., Y is 24, Z is 25)
3. Go to that index in the shifted alphabet and copy that letter to the ciphertext
4. If this is the last letter in the plain text: STOP
5. Consider the next letter in the plain text
6. Go to step 2
```

Let's encode `SDGDAY`

Walk through each step of the algorithm to get the encoded text (... which is the not very secure text of `SDGDAY`)

What if the shift was more than 0?

How do we handle the "index in the shifted alphabet"?

First attempt, consider a shift of 3

```
1. Start with the first letter in the plain text
2. For the given letter, figure out the index in the alphabet (e.g., A is 0, B is 1, ..., Y is 24, Z is 25)
3. Add 3 to that index
4. If the index is more than 25 (the index of Z), subtract 26
5. Go to that index in the alphabet and copy that letter to the ciphertext
6. If this is the last letter in the plain text: STOP
7. Consider the next letter in the plain text
8. Go to step 2
```

Let's encode `SDGDAY`

Walk through each step of the algorithm to get the encoded text (what do you get?)

Only needs *ONE* alphabet!

There is no need to keep that second alphabet order.

Another way to deal with "... went off the end of the alphabet."

Modulus

Division

Consider integer division.

19 divided by 8 is 2 with a remainder of 3

or

19 / 3 is 2 with a remainder of 3

That remainder is considered the "modulus."

We would say:

`19 modulus 8` is `3`

The symbol used in most programming languages is the `%`

`19 % 8` is `3`

How does this help us?

Consider the index divided by 26, but only the remainder.

```
 0 % 26 = 0
 1 % 26 = 1
 2 % 26 = 2
 3 % 26 = 3
 4 % 26 = 4
 5 % 26 = 5
 6 % 26 = 6
 7 % 26 = 7
 8 % 26 = 8
 9 % 26 = 9
10 % 26 = 10
11 % 26 = 11
12 % 26 = 12
13 % 26 = 13
14 % 26 = 14
15 % 26 = 15
16 % 26 = 16
17 % 26 = 17
18 % 26 = 18
19 % 26 = 19
20 % 26 = 20
21 % 26 = 21
22 % 26 = 22
23 % 26 = 23
24 % 26 = 24
25 % 26 = 25
```

… and when we run off the end …

```
20 % 26 = 20
21 % 26 = 21
22 % 26 = 22
23 % 26 = 23
24 % 26 = 24
25 % 26 = 25
26 % 26 = 0   <== Aha, we "wrap" around
27 % 26 = 1
28 % 26 = 2
29 % 26 = 3
30 % 26 = 4
31 % 26 = 5
32 % 26 = 6
33 % 26 = 7
34 % 26 = 8
35 % 26 = 9
```

So another way to write our algorithm

```
1. Start with the first letter in the plain text
2. For the given letter, figure out the index in the alphabet
   (e.g., A is 0, B is 1, ..., Y is 24, Z is 25)
3. Go to the index in the alphabet given by the formula

   (current letter index + shift number) % 26

   and copy that letter to the ciphertext
4. If this is the last letter in the plain text: STOP
5. Consider the next letter in the plain text
6. Go to step 2
```

- One alphabet
- Works for any shift number

How would you work with *DECODING* ?

Example: ROT13

- Simple Caeser Cipher with offset 13 (half the alphabet)
- No need to use a *negative* code to decipher (since `26 - 13` is `13`)
- Try here
- It was used for a while during the early days of the internet to avoid spoilers. Everything was plaintext.
- Example: "Wow, I just saw Empire Strikes Back! I can't believe `Qnegu Inqre vf Yhxr'f sngure!`