

Project 2 Pseudocode

Aim:

- Create a single script that can encrypt and decrypt a secret message with a key
- Use the auto-key cipher to generate an output

Outline:

- * Initialise variables such as message (user input), keyword (user input), alphabet, autokey (used to create matrix of different shifts), and arrays to generate the output eg. Encrypt1, Encrypt2, Decrypt1, Decrypt2.
- * Message needs to be fixed to remove any special characters, punctuation, and spaces from the message. All lowercase need to be converted to uppercase as well.

```

for [each letter in the message]
  if [letter = space, punctuation, special characters]
    message [position of letter] = [clear]
  end
end

```

```

message = upper (message)
key = upper (key)

```

*key needs to be positioned right.
create a for loop that goes through the rows to position the key first.
[length of message - length of key]

- * Create a matrix to look at while encrypting/ decrypting [it is called the Tabula Recta]

```

for 1: 26 % 26 shifts
  autokey = [autokey autokey(1)]
  autokey(1) = [];
end

```

switch (instruction)
case 'Encrypt' % carry out encryption
case 'Decrypt' % carry out decryption
end

- * Then we need to create a switch case to determine if the user wants to encrypt a message or decrypt a message. A menu can also be created.

* Now we need to determine the location of each letter in the message as compared to the original alphabet

```
for [go through the entire message]
    for [go through all the alphabets]
        if [message(m) == alphabet(i)]
            match
            position = alphabet(position)
            location = [location match]
```

end

end

```
end
location2 = [location1; location2]
location1 = [];
```

disp(location2)

* The trickiest part is to use the Tabula Recta and the positions of the message to determine the new output (depends if user wants to encrypt/decrypt the message). "Find" ^{function} key is not allowed!!

for 1:26 % number of shifts

```
for [n=1: message.length]
```

```
    if [letter = key(n)]
```

convert that letter to the correct letter in the alphabet that corresponds to the key

```
    else [continue going through to the next letter until m is done]
```

end

end

end

disp(Encrypt/Decrypt 2)