Question 1

Encoding $LETSSAILFORTHESPANISHMAIN$ with key $PIECESOFEIGHT$ gave

```
AMXUWSWQJWXAATATCRAGMQIOU
```

Question 2

Decoding $ZVTVKGVBLNWYJVCLBOOHSSKFIGWYOEDNZ$ with key $GOLDCOINS$ gave

```
THISISNOTHINGTODOWITHPIRATESATALL
```

Question 3

Essentially $C2 - C1 = K_B$ (done by decrypting $C2$ with the key being $C1$), allowing me to decrypt $C3$ with key K_B to get $M =$ the following

```
ASSOONASWESTARTEDPROGRAMMINGWEFOUNDTOOURSURPRISETHATITWASNT
ASEASYTOGETPROGRAMSRIGHTASWEHADTHOUGHTDEBUGGINGHADTOBE
DISCOVEREDICANREMEMBERTHEEXACTINSTANTWHENIREALIZEDTHATALARGE
PARTOFMYLIFEFROMTHENONWASGOINGTOBESPENTINFINDINGMISTAKESINMY
OWNPROGRAMSMAURICEWILKESDISCOVERSDEBUGGING
```

Please note, line breaks were added for readability, and are **not** part of the original message, nor part of the alphabet

Question 4

Calculations done by code, showing intermediate steps exactly as output by RSA.py

$17^{54}\ mod\ 139 = \mathbf{125}$

```
17^1 = 17 mod 139
17^2 = 11 mod 139
17^4 = 121 mod 139
17^8 = 46 mod 139
```

```
17^16 = 31 mod 139
17^32 = 127 mod 139

Starting with 17^32 mod 139
Multiplying by 17^16, to reach 17^48 mod 139
Multiplying by 17^4, to reach 17^52 mod 139
Multiplying by 17^2, to reach 17^54 mod 139
Calculated 17^54 mod 139 = 125
```

2345^{65531} mod 265189 = **32548**

```
2345^1 = 2345 mod 265189
2345^2 = 195245 mod 265189
2345^4 = 221653 mod 265189
2345^8 = 77513 mod 265189
2345^16 = 143185 mod 265189
2345^32 = 182635 mod 265189
2345^64 = 70805 mod 265189
2345^128 = 215169 mod 265189
2345^256 = 207374 mod 265189
2345^512 = 132069 mod 265189
2345^1024 = 209853 mod 265189
2345^2048 = 200702 mod 265189
2345^4096 = 144460 mod 265189
2345^8192 = 173623 mod 265189
2345^16384 = 116932 mod 265189
2345^32768 = 212973 mod 265189

Starting with 2345^32768 mod 265189
Multiplying by 2345^16384, to reach 2345^49152 mod 265189
Multiplying by 2345^8192, to reach 2345^57344 mod 265189
Multiplying by 2345^4096, to reach 2345^61440 mod 265189
Multiplying by 2345^2048, to reach 2345^63488 mod 265189
Multiplying by 2345^1024, to reach 2345^64512 mod 265189
Multiplying by 2345^512, to reach 2345^65024 mod 265189
Multiplying by 2345^256, to reach 2345^65280 mod 265189
Multiplying by 2345^128, to reach 2345^65408 mod 265189
Multiplying by 2345^64, to reach 2345^65472 mod 265189
Multiplying by 2345^32, to reach 2345^65504 mod 265189
Multiplying by 2345^16, to reach 2345^65520 mod 265189
Multiplying by 2345^8, to reach 2345^65528 mod 265189
```

```
Multiplying by 2345^2, to reach 2345^65530 mod 265189
Multiplying by 2345^1, to reach 2345^65531 mod 265189
Calculated 2345^65531 mod 265189 = 32548
```

4733459^{65537} *mod* $75968647 = \mathbf{621879}$

```
4733459^1 = 4733459 mod 75968647
4733459^2 = 49107677 mod 75968647
4733459^4 = 16238929 mod 75968647
4733459^8 = 67757406 mod 75968647
4733459^16 = 25488171 mod 75968647
4733459^32 = 64480977 mod 75968647
4733459^64 = 57889554 mod 75968647
4733459^128 = 19358089 mod 75968647
4733459^256 = 50744319 mod 75968647
4733459^512 = 56497489 mod 75968647
4733459^1024 = 54825938 mod 75968647
4733459^2048 = 38930457 mod 75968647
4733459^4096 = 49024383 mod 75968647
4733459^8192 = 51007254 mod 75968647
4733459^16384 = 24313 mod 75968647
4733459^32768 = 59341440 mod 75968647
4733459^65536 = 51988154 mod 75968647

Starting with 4733459^65536 mod 75968647
Multiplying by 4733459^1, to reach 4733459^65537 mod 75968647
Calculated 4733459^65537 mod 75968647 = 621879
```

Question 5

You wish to securely send the message M=654733 to the bank

i) Calculation used to encrypt this message for sending to the bank

$C = M^{e_{bank}} \ mod \ n_{bank}$

In this case,
$C = 654733^{65537} \ mod \ 76282747$

ii) The encrypted value, calculated by my code

```
39964485
```

Question 6

The bank sends you an encrypted message 1684446

i) Calculation used in decryption

$C^{d_{mine}} = M^{e_{mine}d_{mine}} = M\ mod\ n_{mine}$

In this case,
$C^{3497603} = M^{1676267 \cdot 3497603} = M\ mod\ 9436709$

ii) The decrypted value in this case

```
1101011
```

Question 7

The bank requests a signed encrypted message from you so that they can verify that you are the sender and the message is secure in transmission to them. You should encrypt the message and signature as two separate blocks. They already know your public key.

i) The calculation to sign and encrypt the message 337722

Sign the message
$S = M^{d_{mine}}\ mod\ n_{mine}$

Encrypt the message
$C_M = M^{e_{bank}}\ mod\ n_{bank}$

Encrypt the signature
$C_S = S^{e_{bank}}\ mod\ n_{bank}$

In this case,
$S = 337722^{3497603}\ mod\ 9436709$

Encrypt the message
$C_M = 337722^{65537}\ mod\ 76282747$

Encrypt the signature
$C_S = S^{65537}\ mod\ 76282747$

ii) The transmission made for the message 337722 when it has been signed and encrypted

Intermediate step, signing the message

```
S = 7218665
```

Actual transmission made below

```
C(S) = 59821766
C(M) = 33191197
```

Question 8

They return the following signed and encrypted message to you:
(C(M),C(S)) = (4647068,526345)

i) The calculations required to decrypt and verify the message

To decrypt the message
$C(M)^{d_{mine}} = M^{e_{mine}d_{mine}} = M\ mod\ n_{mine}$

To decrypt the signature
$C(S)^{d_{mine}} = S^{e_{mine}d_{mine}} = S\ mod\ n_{mine}$

To 'un-sign' the signature (retrieve the message)
$M_2 = S^{e_{bank}}\ mod\ n_{bank}$

In this case,
To decrypt the message
$4647068^{3497603} = M^{1676267 \cdot 3497603} = M \ mod \ 9436709$

To decrypt the signature
$526345^{3497603} = S^{1676267 \cdot 3497603} = S \ mod \ 9436709$

To 'un-sign' the signature (retrieve the message)
$M_2 = S^{65537} \ mod \ 76282747$

ii) The values in this case

```
Decrypted M = 7406060
Decrypted S = 8180219
'Un-signed' S = 64026314
```

iii) Signature validity

It is not valid, as after being 'un-signed', it does not match the decrypted M.

Question 9

You wish to demonstrate to the bank that you know a third party (Bob), by showing a signed message that appears to come from Bob. Bobs public key is (122269479, 53407), but you do not know his private key. Construct a valid signed message from Bob, and show that the check calculations confirm that signature is valid.

```
Chosen signature = 54321246
Calculated message = 36464280
```

We check that the chosen signature is valid for this message by 'un-signing' it, using Bob's public key, to retrieve the message it represents.

Verify $M = S^{e_{bob}} \ mod \ n_{bob}$
Verify $36464280 = 54321246^{53407} \ mod \ 122269479$
The steps carried out to verify this result are shown below

```
54321246^1 = 54321246 mod 122269479
54321246^2 = 81646755 mod 122269479
54321246^4 = 83557920 mod 122269479
```

```
54321246^8 = 57369570 mod 122269479
54321246^16 = 119457924 mod 122269479
54321246^32 = 119700675 mod 122269479
54321246^64 = 114747744 mod 122269479
54321246^128 = 86356824 mod 122269479
54321246^256 = 27923511 mod 122269479
54321246^512 = 95156322 mod 122269479
54321246^1024 = 48525369 mod 122269479
54321246^2048 = 81219015 mod 122269479
54321246^4096 = 15183681 mod 122269479
54321246^8192 = 53006622 mod 122269479
54321246^16384 = 90264669 mod 122269479
54321246^32768 = 71323281 mod 122269479

Starting with 54321246^32768 mod 122269479
Multiplying by 54321246^16384, to reach 54321246^49152 mod 122269479
Multiplying by 54321246^4096, to reach 54321246^53248 mod 122269479
Multiplying by 54321246^128, to reach 54321246^53376 mod 122269479
Multiplying by 54321246^16, to reach 54321246^53392 mod 122269479
Multiplying by 54321246^8, to reach 54321246^53400 mod 122269479
Multiplying by 54321246^4, to reach 54321246^53404 mod 122269479
Multiplying by 54321246^2, to reach 54321246^53406 mod 122269479
Multiplying by 54321246^1, to reach 54321246^53407 mod 122269479
Signature 54321246 found to represent message: 36464280
```

Question 10

You intercept a message from Bob to the bank, which says: My new 3-digit PIN code is in the encrypted attachment. Yours Bob. and comes with the attachment (58621765). How can you crack such a message and what is Bobs new PIN?

Being only three digits, there are only 1000 possible PINs. As we know the Bank's public key, we can brute force values $000, 001, ..., 999$, encrypting each of them with the Bank's public key, until one of them matches 58621765, at which point we have found Bob's new PIN. As this is such a small range of possiblities, the inefficiency of brute force is negligible in this case.

My brute force, at the bottom of RSA.py, found the following

```
Bob's new pin = 777
```