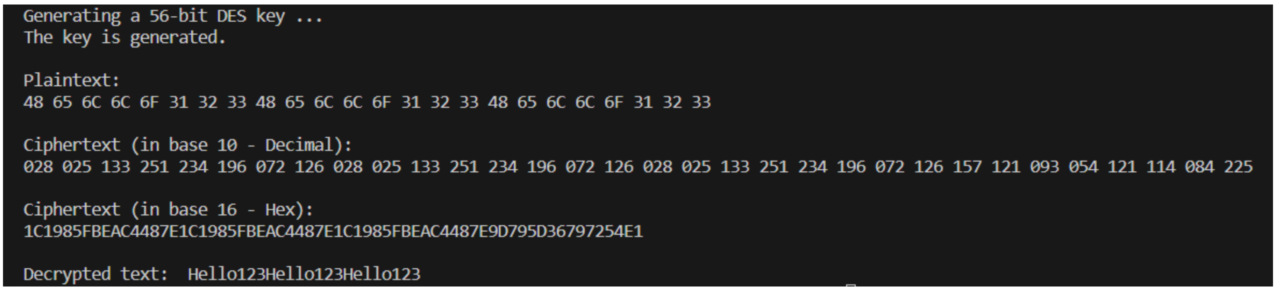
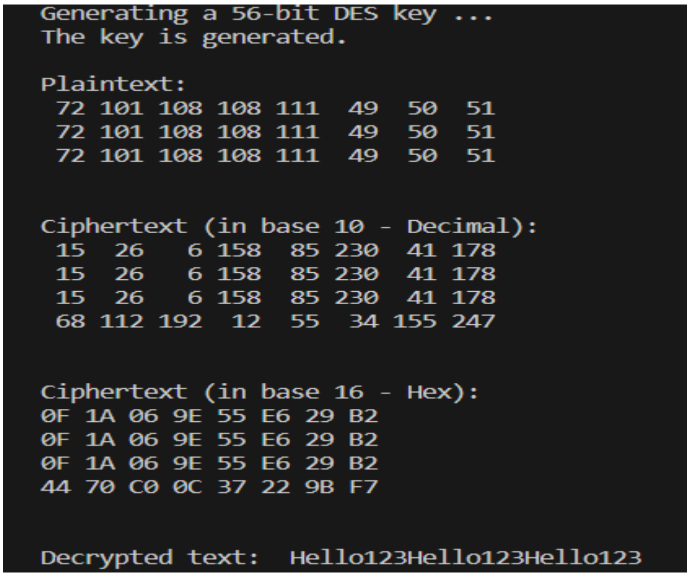
Applied Cryptography Exercises

Prac2Q1



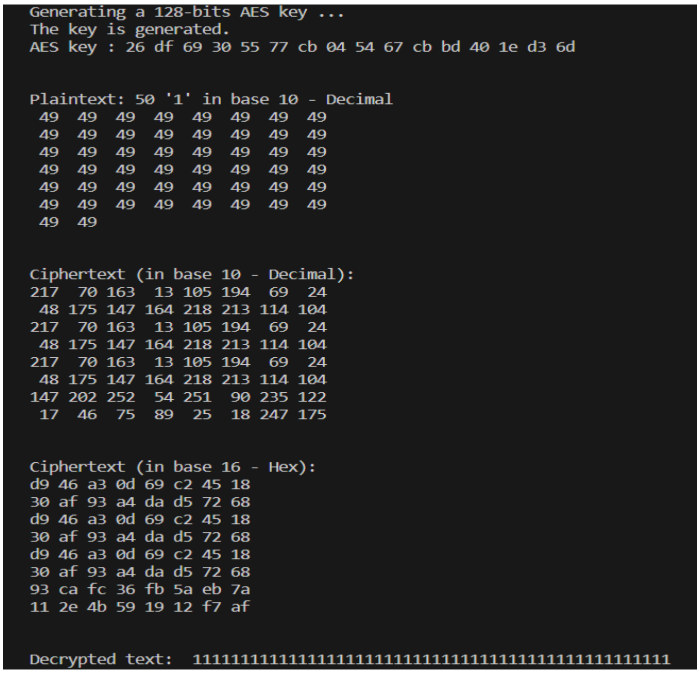
```
Generating a 56-bit DES key ...
The key is generated.

Plaintext:
48 65 6C 6C 6F 31 32 33 48 65 6C 6C 6F 31 32 33 48 65 6C 6C 6F 31 32 33

Ciphertext (in base 10 - Decimal):
028 025 133 251 234 196 072 126 028 025 133 251 234 196 072 126 028 025 133 251 234 196 072 126 157 121 093 054 121 114 084 225

Ciphertext (in base 16 - Hex):
1C1985FBEAC4487E1C1985FBEAC4487E1C1985FBEAC4487E9D795D36797254E1

Decrypted text:  Hello123Hello123Hello123
```

Prac2Q2



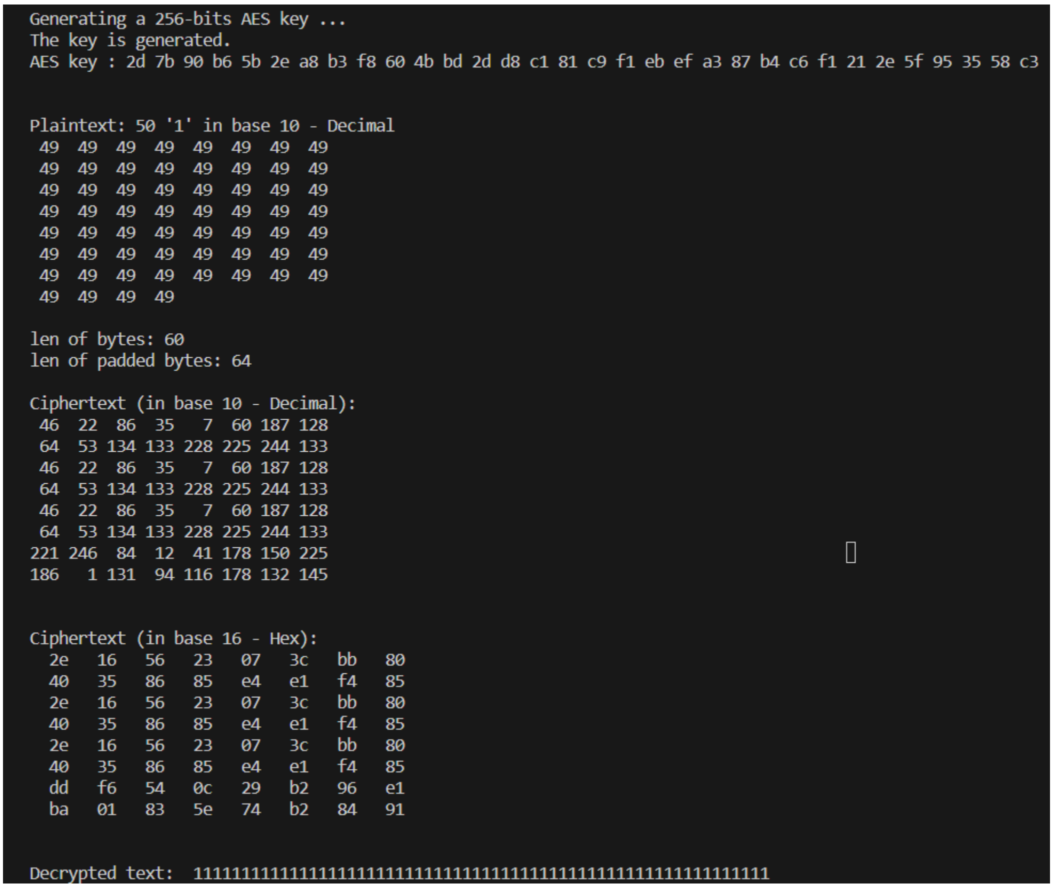
```
Generating a 56-bit DES key ...
The key is generated.

Plaintext:
 72 101 108 108 111  49  50  51
 72 101 108 108 111  49  50  51
 72 101 108 108 111  49  50  51


Ciphertext (in base 10 - Decimal):
 15  26   6 158  85 230  41 178
 15  26   6 158  85 230  41 178
 15  26   6 158  85 230  41 178
 68 112 192  12  55  34 155 247


Ciphertext (in base 16 - Hex):
0F 1A 06 9E 55 E6 29 B2
0F 1A 06 9E 55 E6 29 B2
0F 1A 06 9E 55 E6 29 B2
44 70 C0 0C 37 22 9B F7


Decrypted text:  Hello123Hello123Hello123
```

Prac3Q1



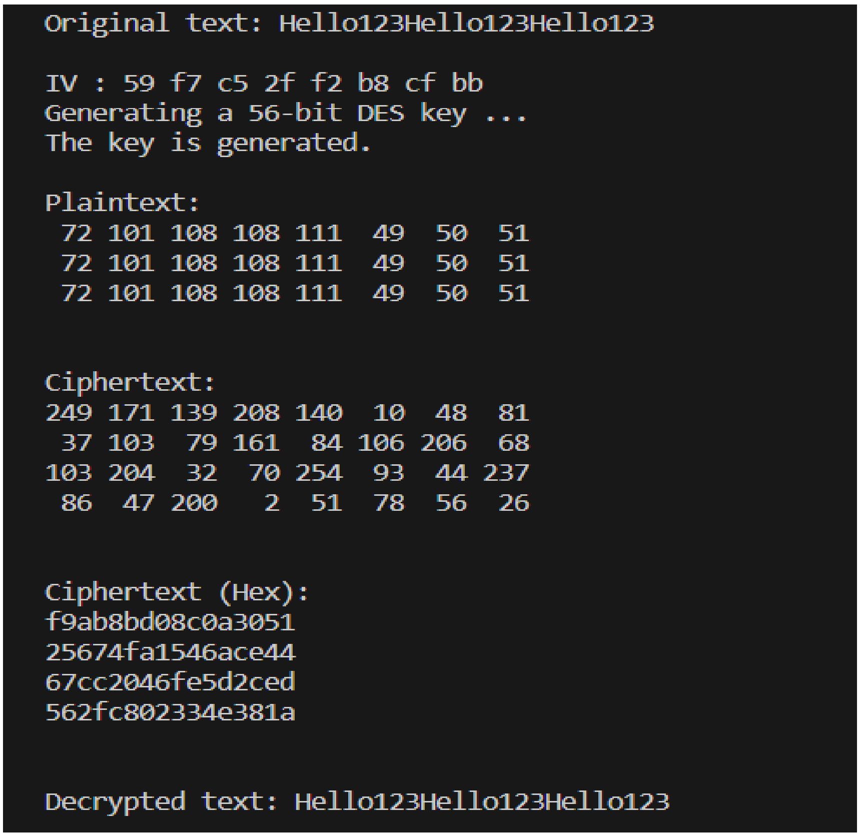
```
Generating a 128-bits AES key ...
The key is generated.
AES key : 26 df 69 30 55 77 cb 04 54 67 cb bd 40 1e d3 6d


Plaintext: 50 '1' in base 10 - Decimal
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49


Ciphertext (in base 10 - Decimal):
217  70 163  13 105 194  69  24
 48 175 147 164 218 213 114 104
217  70 163  13 105 194  69  24
 48 175 147 164 218 213 114 104
217  70 163  13 105 194  69  24
 48 175 147 164 218 213 114 104
147 202 252  54 251  90 235 122
 17  46  75  89  25  18 247 175


Ciphertext (in base 16 - Hex):
d9 46 a3 0d 69 c2 45 18
30 af 93 a4 da d5 72 68
d9 46 a3 0d 69 c2 45 18
30 af 93 a4 da d5 72 68
d9 46 a3 0d 69 c2 45 18
30 af 93 a4 da d5 72 68
93 ca fc 36 fb 5a eb 7a
11 2e 4b 59 19 12 f7 af


Decrypted text:  11111111111111111111111111111111111111111111111111
```

Prac3Q2



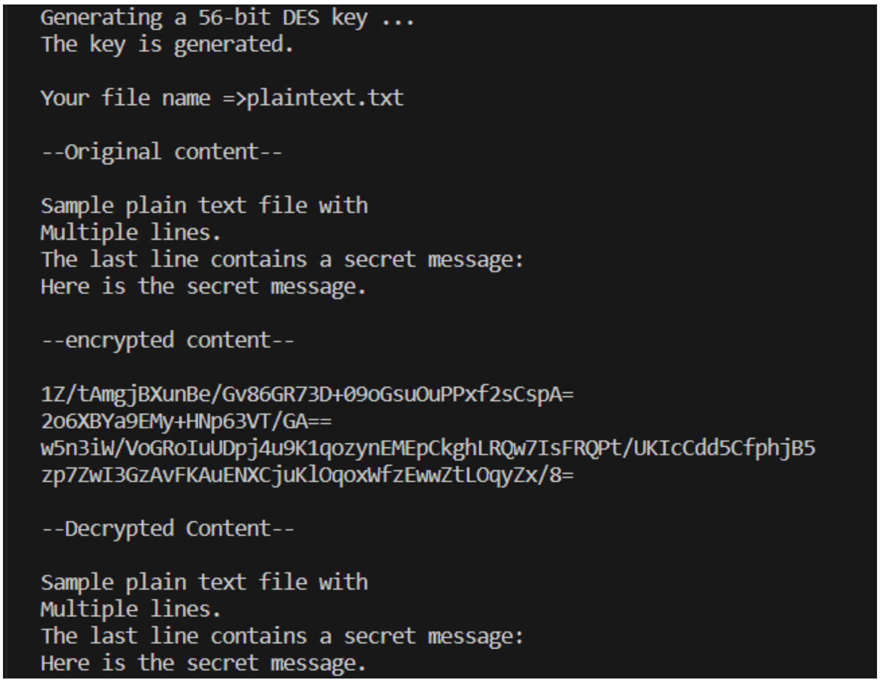
```
Generating a 256-bits AES key ...
The key is generated.
AES key : 2d 7b 90 b6 5b 2e a8 b3 f8 60 4b bd 2d d8 c1 81 c9 f1 eb ef a3 87 b4 c6 f1 21 2e 5f 95 35 58 c3


Plaintext: 50 '1' in base 10 - Decimal
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49  49  49  49  49
 49  49  49  49

len of bytes: 60
len of padded bytes: 64

Ciphertext (in base 10 - Decimal):
 46  22  86  35   7  60 187 128
 64  53 134 133 228 225 244 133
 46  22  86  35   7  60 187 128
 64  53 134 133 228 225 244 133
 46  22  86  35   7  60 187 128
 64  53 134 133 228 225 244 133
221 246  84  12  41 178 150 225
186   1 131  94 116 178 132 145


Ciphertext (in base 16 - Hex):
  2e    16    56    23    07    3c    bb    80
  40    35    86    85    e4    e1    f4    85
  2e    16    56    23    07    3c    bb    80
  40    35    86    85    e4    e1    f4    85
  2e    16    56    23    07    3c    bb    80
  40    35    86    85    e4    e1    f4    85
  dd    f6    54    0c    29    b2    96    e1
  ba    01    83    5e    74    b2    84    91


Decrypted text:  111111111111111111111111111111111111111111111111111111111111
```

Prac4Q1



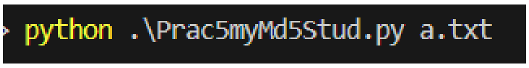
```
Original text: Hello123Hello123Hello123

IV : 59 f7 c5 2f f2 b8 cf bb
Generating a 56-bit DES key ...
The key is generated.

Plaintext:
 72 101 108 108 111  49  50  51
 72 101 108 108 111  49  50  51
 72 101 108 108 111  49  50  51


Ciphertext:
249 171 139 208 140  10  48  81
 37 103  79 161  84 106 206  68
103 204  32  70 254  93  44 237
 86  47 200   2  51  78  56  26


Ciphertext (Hex):
f9ab8bd08c0a3051
25674fa1546ace44
67cc2046fe5d2ced
562fc802334e381a


Decrypted text: Hello123Hello123Hello123
```

Prac4Q2

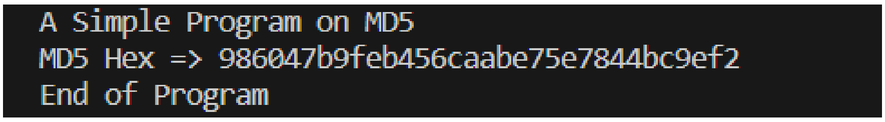


```
Generating a 56-bit DES key ...
The key is generated.

Your file name =>plaintext.txt

--Original content--

Sample plain text file with
Multiple lines.
The last line contains a secret message:
Here is the secret message.

--encrypted content--

1Z/tAmgjBXunBe/Gv86GR73D+09oGsuOuPPxf2sCspA=
2o6XBYa9EMy+HNp63VT/GA==
w5n3iW/VoGRoIuUDpj4u9K1qozynEMEpCkghLRQw7IsFRQPt/UKIcCdd5CfphjB5
zp7ZwI3GzAvFKAuENXCjuKlOqoxWfzEwwZtLOqyZx/8=

--Decrypted Content--

Sample plain text file with
Multiple lines.
The last line contains a secret message:
Here is the secret message.
```

Prac5Q1

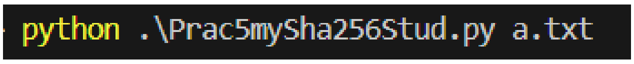


```
python .\Prac5myMd5Stud.py a.txt
```

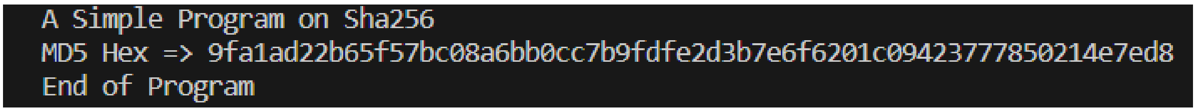


```
A Simple Program on MD5
MD5 Hex => 986047b9feb456caabe75e7844bc9ef2
End of Program
```

Prac5Q2

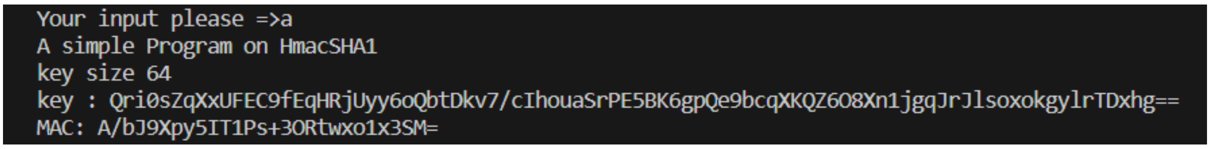


```
python .\Prac5mySha256Stud.py a.txt
```



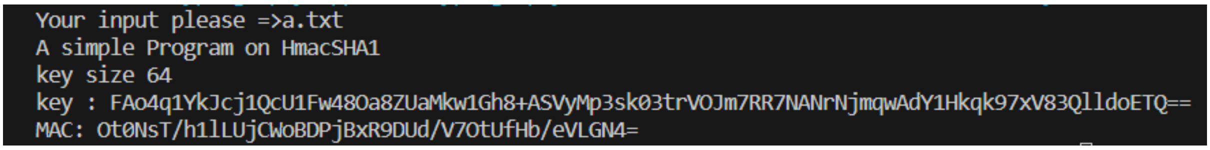
```
A Simple Program on Sha256
MD5 Hex => 9fa1ad22b65f57bc08a6bb0cc7b9fdfe2d3b7e6f6201c09423777850214e7ed8
End of Program
```

Prac6Q1



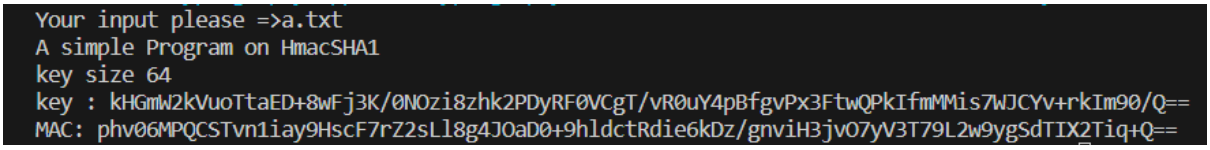
```
Your input please =>a
A simple Program on HmacSHA1
key size 64
key : Qri0sZqXxUFEC9fEqHRjUyy6oQbtDkv7/cIhouaSrPE5BK6gpQe9bcqXKQZ6O8Xn1jgqJrJlsoxokgylrTDxhg==
MAC: A/bJ9Xpy5IT1Ps+3ORtwxo1x3SM=
```

Prac6Q2



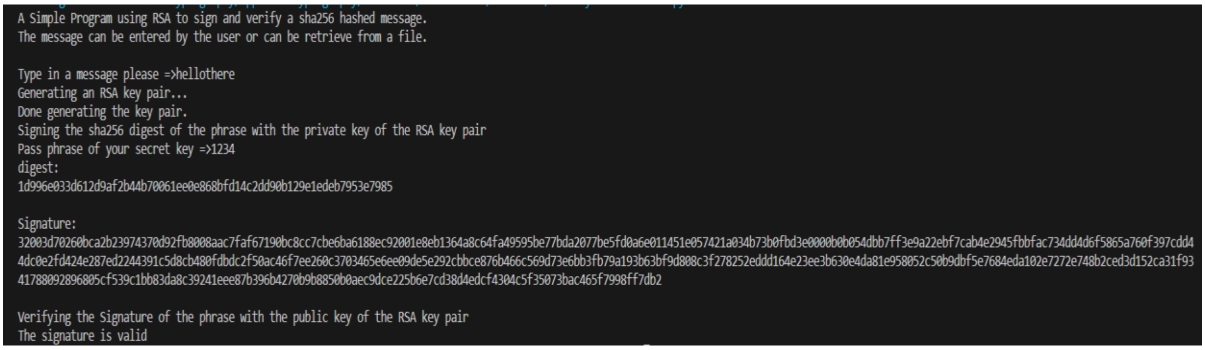
```
Your input please =>a.txt
A simple Program on HmacSHA1
key size 64
key : FAo4q1YkJcj1QcU1Fw48Oa8ZUaMkw1Gh8+ASVyMp3sk03trVOJm7RR7NANrNjmqwAdY1Hkqk97xV83QlldoETQ==
MAC: Ot0NsT/h1lLUjCWoBDPjBxR9DUd/V7OtUfHb/eVLGN4=
```

Prac6Q3



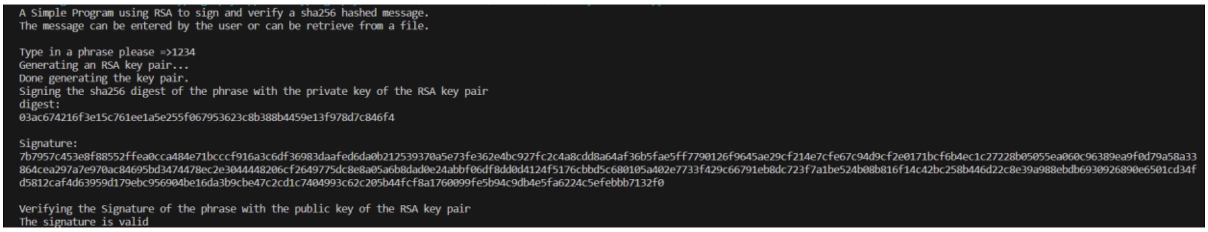
```
Your input please =>a.txt
A simple Program on HmacSHA1
key size 64
key : kHGmW2kVuoTtaED+8wFj3K/0NOzi8zhk2PDyRF0VCgT/vR0uY4pBfgvPx3FtwQPkIfmMMis7WJCYv+rkIm90/Q==
MAC: phv06MPQCSTvn1iay9HscF7rZ2sLl8g4JOaD0+9hldctRdie6kDz/gnviH3jvO7yV3T79L2w9ygSdTIX2Tiq+Q==
```

Prac7Q1



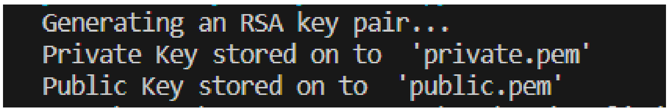
```
A Simple Program using RSA to sign and verify a sha256 hashed message.
The message can be entered by the user or can be retrieve from a file.

Type in a message please =>hellothere
Generating an RSA key pair...
Done generating the key pair.
Signing the sha256 digest of the phrase with the private key of the RSA key pair
Pass phrase of your secret key =>1234
digest:
1d996e033d612d9af2b44b70061ee0e868bfd14c2dd90b129e1edeb7953e7985

Signature:
32003d70260bca2b23974370d92fb8008aac7faf67190bc8cc7cbe6ba6188ec92001e8eb1364a8c64fa49595be77bda2077be5fd0a6e011451e057421a034b73b0fbd3e0000b0b054dbb7ff3e9a22ebf7cab4e2945fbbfac734dd4d6f5865a760f397cdd4
4dc0e2fd424e287ed2244391c5d8cb480fdbdc2f50ac46f7ee260c3703465e6ee09de5e292cbbce876b466c569d73e6bb3fb79a193b63bf9d808c3f278252eddd164e23ee3b630e4da81e958052c50b9dbf5e7684eda102e7272e748b2ced3d152ca31f93
41788092896805cf539c1bb83da8c39241eee87b396b4270b9b8850b0aec9dce225b6e7cd38d4edcf4304c5f35073bac465f7998ff7db2

Verifying the Signature of the phrase with the public key of the RSA key pair
The signature is valid
```

Prac7Q2

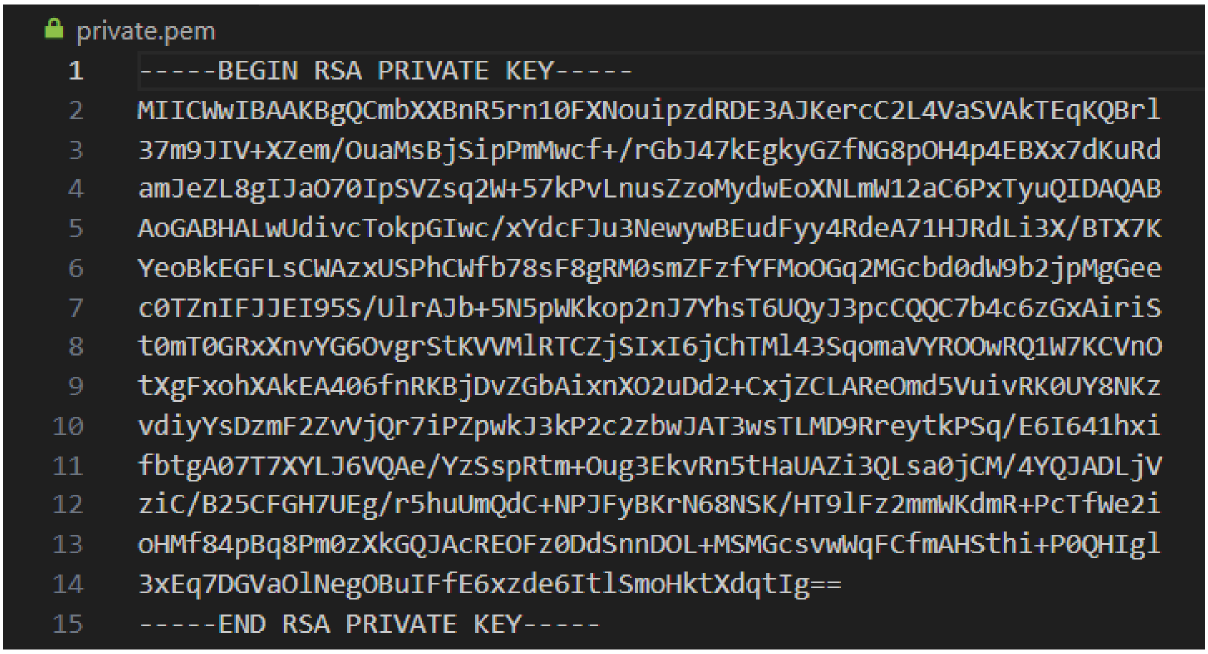


```
A Simple Program using RSA to sign and verify a sha256 hashed message.
The message can be entered by the user or can be retrieve from a file.

Type in a phrase please =>1234
Generating an RSA key pair...
Done generating the key pair.
Signing the sha256 digest of the phrase with the private key of the RSA key pair
digest:
03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4

Signature:
7b7957c453e8f88552ffea0cca484e71bcccf916a3c6df36983daafed6da0b212539370a5e73fe362e4bc927fc2c4a8cdd8a64af36b5fae5ff7790126f9645ae29cf214e7cfe67c94d9cf2e0171bcf6b4ec1c27228b05055ea060c96389ea9f0d79a58a33
864cea297a7e970ac84695bd3474478ec2e3044448206cf2649775dc8e8a05a6b8dad0e24abbf06df8dd0d4124f5176cbbd5c680105a402e7733f429c66791eb8dc723f7a1be524b08b816f14c42bc258b446d22c8e39a988ebdb6930926890e6501cd34f
d5812caf4d63959d179ebc956904be16da3b9cbe47c2cd1c7404993c62c205b44fcf8a1760099fe5b94c9db4e5fa6224c5efebbb7132f0

Verifying the Signature of the phrase with the public key of the RSA key pair
The signature is valid
```

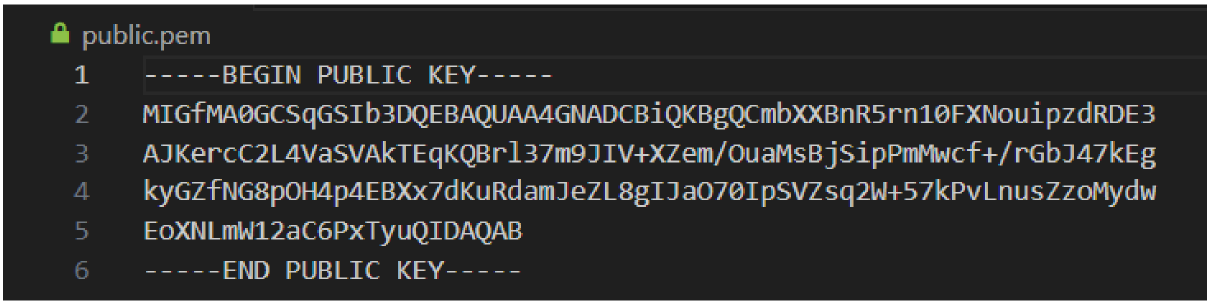
Prac8Q1 – Generate RSA Key Pair



```
Generating an RSA key pair...
Private Key stored on to  'private.pem'
Public Key stored on to  'public.pem'
```

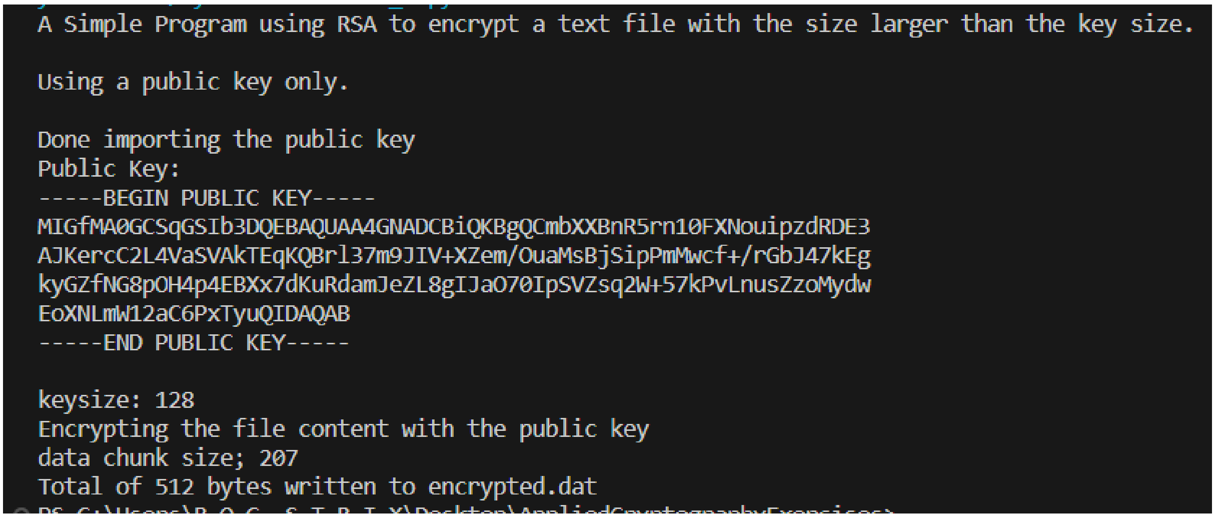


```
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBgQCmbXXBnR5rn10FXNouipzdRDE3AJKercC2L4VaSVAkTEqKQBrl
37m9JIV+XZem/OuaMsBjSipPmMwcf+/rGbJ47kEgkyGZfNG8pOH4p4EBXx7dKuRd
amJeZL8gIJaO70IpSVZsq2W+57kPvLnusZzoMydwEoXNLmW12aC6PxTyuQIDAQAB
AoGABHALwUdivcTokpGIwc/xYdcFJu3NewywBEudFyy4RdeA71HJRdLi3X/BTX7K
YeoBkEGFLsCWAzxUSPhCWfb78sF8gRM0smZFzfYFMoOGq2MGcbd0dW9b2jpMgGee
c0TZnIFJJEI95S/UlrAJb+5N5pWKkop2nJ7YhsT6UQyJ3pcCQQC7b4c6zGxAiriS
t0mT0GRxXnvYG6OvgrStKVVMlRTCZjSIxI6jChTMl43SqomaVYROOwRQ1W7KCVnO
tXgFxohXAkEA406fnRKBjDvZGbAixnXO2uDd2+CxjZCLAReOmd5VuivRK0UY8NKz
vdiyYsDzmF2ZvVjQr7iPZpwkJ3kP2c2zbwJAT3wsTLMD9RreytkPSq/E6I641hxi
fbtgA07T7XYLJ6VQAe/YzSspRtm+Oug3EkvRn5tHaUAZi3QLsa0jCM/4YQJADLjV
ziC/B25CFGH7UEg/r5huUmQdC+NPJFyBKrN68NSK/HT9lFz2mmWKdmR+PcTfWe2i
oHMf84pBq8Pm0zXkGQJAcREOFz0DdSnnDOL+MSMGcsvwWqFCfmAHSthi+P0QHIgl
3xEq7DGVaOlNegOBuIFfE6xzde6ItlSmoHktXdqtIg==
-----END RSA PRIVATE KEY-----
```

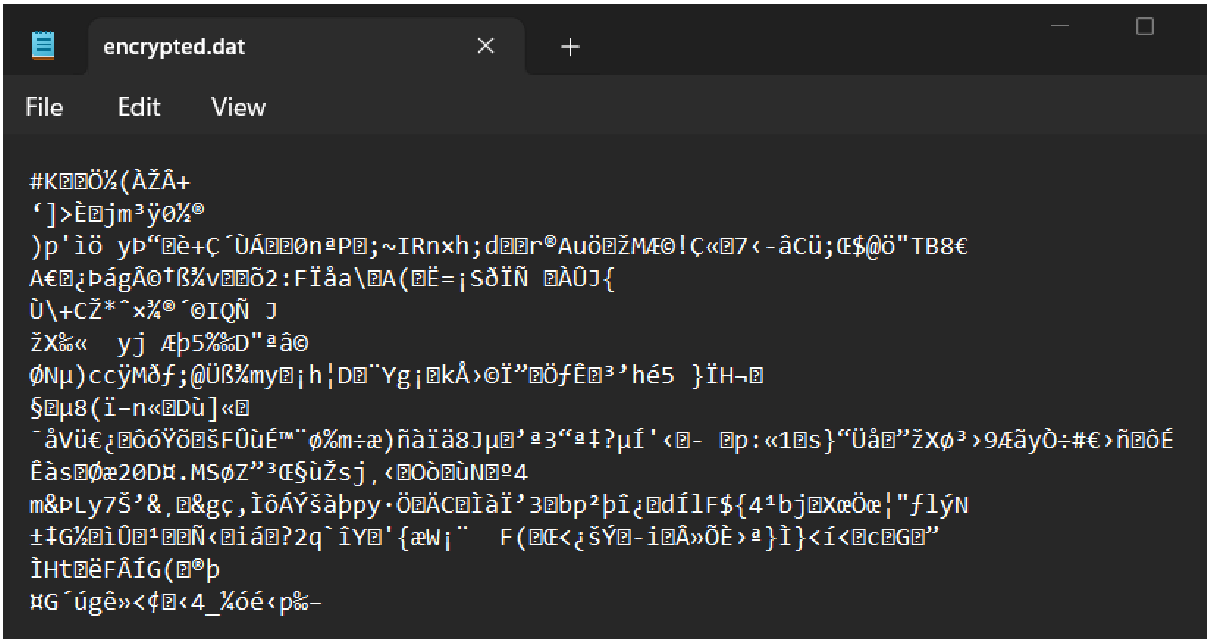


```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmbXXBnR5rn10FXNouipzdRDE3
AJKercC2L4VaSVAkTEqKQBrl37m9JIV+XZem/OuaMsBjSipPmMwcf+/rGbJ47kEg
kyGZfNG8pOH4p4EBXx7dKuRdamJeZL8gIJaO70IpSVZsq2W+57kPvLnusZzoMydw
EoXNLmW12aC6PxTyuQIDAQAB
-----END PUBLIC KEY-----
```

Prac8Q2 – Encrypt the file.

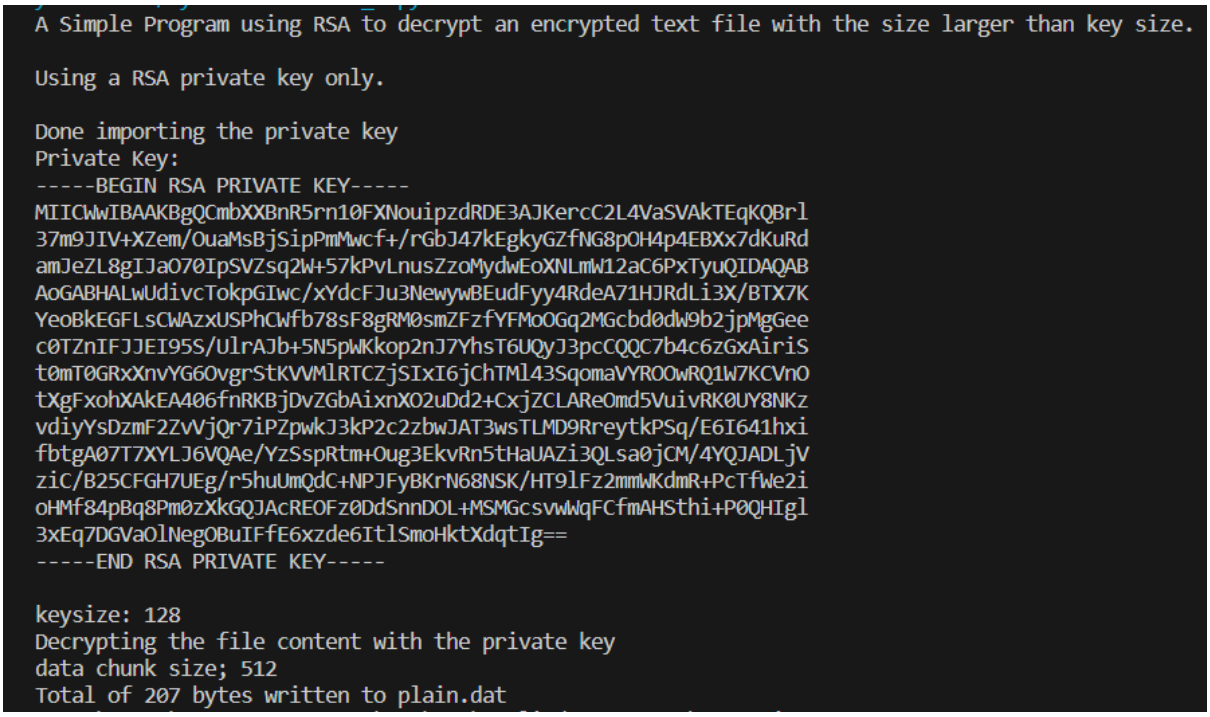


```
A Simple Program using RSA to encrypt a text file with the size larger than the key size.

Using a public key only.

Done importing the public key
Public Key:
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmbXXBnR5rn10FXNouipzdRDE3
AJKercC2L4VaSVAkTEqKQBrl37m9JIV+XZem/OuaMsBjSipPmMwcf+/rGbJ47kEg
kyGZfNG8pOH4p4EBXx7dKuRdamJeZL8gIJaO70IpSVZsq2W+57kPvLnusZzoMydw
EoXNLmW12aC6PxTyuQIDAQAB
-----END PUBLIC KEY-----

keysize: 128
Encrypting the file content with the public key
data chunk size; 207
Total of 512 bytes written to encrypted.dat
```

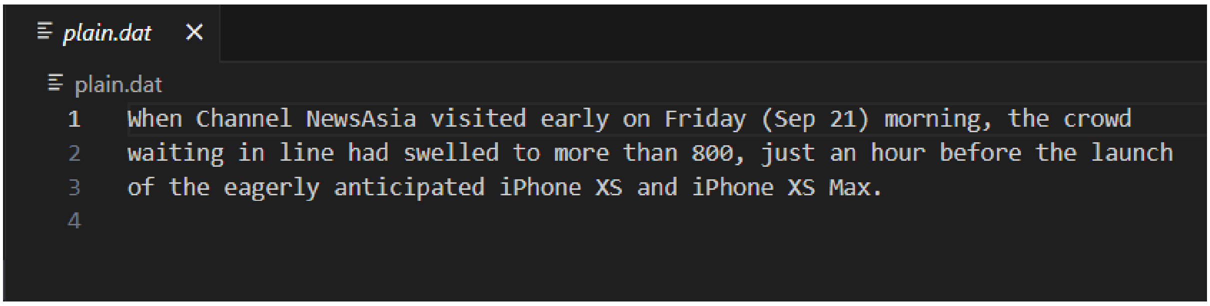


```
#K□□Ö½(ÀŽÂ+
‘]>È□jm³ÿ0½®
)p'ìö yÞ“□è+Ç´ÙÁ□□0nªP□;~IRn×h;d□□r®AuÖ□ŽMÆ©!Ç«□7‹-âCü;Œ$@ö"TB8€
A€□¿ÞágÂ©†ß¾v□□õ2:FÏåa\□A(□Ë=¡SðÏÑ □ÀÛJ{
Ù\+CŽ*ˆx¾®´©IQÑ J
žX‰«  yj Æþ5%‰D"ªâ©
ØNµ)ccÿMðƒ;@ÜßÆmy□¡h¦D□¨Yg¡□kÅ›©Ï”□ÖƒÊ□³’hé5 }ÏH¬□
§□µ8(ï–n«□Dù]«□
¯åVü€¿□ôóŸõ□šFÛùÉ™¨ø%m÷æ)ñàïä8Jµ□’ª3“ª‡?µÍ'‹□- □p:«1□s}“Üå□”žXø³›9ÆãyÒ÷#€›ñ□ôÉ
Êàs□Øæ20D¤.MSøZ”³Œ§ùŽsj‚‹□Oò□ùN□º4
m&ÞLy7Š’&‚□&gç,ÌôÁÝšàþpy·Ö□ÄC□ÌàÏ’3□bp²þî¿□dÍlF$ {4¹bj□XœÖœ¦"ƒlýN
±‡G½□ìÛ□¹□□Ñ‹□iá□?2q`îY□'{æW¡¨  F(□Œ<¿šÝ□-i□Â»ÕÈ›ª}Ì}<í<□c□G□”
ÌHt□ëFÂÍG(□®þ
¤G´úgê»<¢□‹4_¼óé‹p‰–
```

Prac8Q3 - Decrypt the file.



```
A Simple Program using RSA to decrypt an encrypted text file with the size larger than key size.

Using a RSA private key only.

Done importing the private key
Private Key:
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBgQCmbXXBnR5rn10FXNouipzdRDE3AJKercC2L4VaSVAkTEqKQBrl
37m9JIV+XZem/OuaMsBjSipPmMwcf+/rGbJ47kEgkyGZfNG8pOH4p4EBXx7dKuRd
amJeZL8gIJaO70IpSVZsq2W+57kPvLnusZzoMydwEoXNLmW12aC6PxTyuQIDAQAB
AoGABHALwUdivcTokpGIwc/xYdcFJu3NewywBEudFyy4RdeA71HJRdLi3X/BTX7K
YeoBkEGFLsCWAzxUSPhCWfb78sF8gRM0smZFzfYFMoOGq2MGcbd0dW9b2jpMgGee
c0TZnIFJJEI95S/UlrAJb+5N5pWKkop2nJ7YhsT6UQyJ3pcCQQC7b4c6zGxAiriS
t0mT0GRxXnvYG6OvgrStKVVMlRTCZjSIxI6jChTMl43SqomaVYROOwRQ1W7KCVnO
tXgFxohXAkEA406fnRKBjDvZGbAixnXO2uDd2+CxjZCLAReOmd5VuivRK0UY8NKz
vdiyYsDzmF2ZvVjQr7iPZpwkJ3kP2c2zbwJAT3wsTLMD9RreytkPSq/E6I641hxi
fbtgA07T7XYLJ6VQAe/YzSspRtm+Oug3EkvRn5tHaUAZi3QLsa0jCM/4YQJADLjV
ziC/B25CFGH7UEg/r5huUmQdC+NPJFyBKrN68NSK/HT9lFz2mmWKdmR+PcTfWe2i
oHMf84pBq8Pm0zXkGQJAcREOFz0DdSnnDOL+MSMGcsvwWqFCfmAHSthi+P0QHIgl
3xEq7DGVaOlNegOBuIFfE6xzde6ItlSmoHktXdqtIg==
-----END RSA PRIVATE KEY-----

keysize: 128
Decrypting the file content with the private key
data chunk size; 512
Total of 207 bytes written to plain.dat
```



```
When Channel NewsAsia visited early on Friday (Sep 21) morning, the crowd
waiting in line had swelled to more than 800, just an hour before the launch
of the eagerly anticipated iPhone XS and iPhone XS Max.

```