

CS/MATH111 RSA EXTRA CREDIT ASSIGNMENT

due Tuesday, October 29

Solution 1: Solution is in c++ code:

The user is prompted for a public key -

Valid public keys tested(e , n):

23 55, 13 77, 7 143, 5 91, 7 95, 11 51, 9 33

The user is prompted to enter either e or d to encrypt or decrypt a message.

If encrypt is chosen, the message in the file "Encrypt.txt" is encrypted and the coded message is written to the file "EncryptOutput.txt".

If decrypt is chosen, the encoded message in the file "Decrypt.txt" is decrypted and the decoded message is written to the file "DecryptOutput.txt".

Encrypting and decrypting is according to the schema used previously, (A is 2, B is 3, ..., Z is 27, and space is 28).

The code was tested using four different quotes:




"EVERYTHING SHOULD BE MADE AS SIMPLE AS POSSIBLE BUT NOT SIMPLER"

"THE GREATEST GLORY IN LIVING LIES NOT IN NEVER FALLING BUT IN RISING EVERY TIME WE FALL"

"DO NOT GO WHERE THE PATH MAY LEAD GO INSTEAD WHERE THERE IS NO PATH AND LEAVE A TRAIL"

"YOU KNOW YOU ARE ON THE ROAD TO SUCCESS IF YOU WOULD DO YOUR JOB AND NOT BE PAID FOR IT"

See code below:

Files   

main.cpp

Decrypt.txt




DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

main.cpp saved

```
1  #include<iostream>
2  #include <fstream>
3  #include<math.h>
4
5  using namespace std;
6
7  void primesfinder(int& p, int& q, int n){
8  for(int i = 2; i < n; ++i){
9      if((n % i) == 0){
10         p = i;
11         q = n/p;
12     }
13 }
14 }
15
16
17 int modulo(int c, int d, int n){
18     if (c == 0){
19         c = 2;
20     }
21     long long x=1, y=c;
22     while (d > 0) {
23         if (d%2 == 1) {
24             x = (x*y) % n;
25         }
26         y = (y*y) % n;
27         d /= 2;
28     }
29     return x % n;
30 }
31
32
33 int d_finder(int phi, int e){
```

Files   


main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

main.cpp saved 

```
33 int d_finder(int phi, int e){
34     for (int k = 1; k < 100000; ++k){
35         for (int i = 1; i < 100000; ++i){
36             if ((i*e) == (1 + (k*phi))){
37                 return i;
38             }
39         }
40     }
41     return -1;
42 }
43
44 int main(){
45     ifstream inFS;
46     int filenumber;
47
48     int e;
49     int n;
50     cout << "Enter a public key: e & n with a space between:" <<endl;
51     cin >> e;
52     cin >> n;
53     int p = 0;
54     int q = 0;
55     int d;
56     int totient;
57     //int k = 4;
58     int input;
59     int m;
60     int c;
61
62     primesfinder(p,q,n);
63     totient = (p-1)*(q-1);
64
65     //d = (1 + (k*totient))/e;
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
65 //d = (1 + (k*totient))/e;
66 d = d_finder(totient, e);
67 char ch;
68 cout << "Encrypt(e) or decrypt(d)" << endl;
69 cin >> ch;
70
71 if (ch == 'e'){
72     inFS.open("Encrypt.txt");
73     ofstream outFS;
74     outFS.open("EncryptOutput.txt");
75     while (inFS.get(ch)) {
76         //inFS.get(ch);
77         if(ch == 'A'){
78             m = 2;
79         }
80         else if (ch == 'B'){
81             m = 3;
82         }
83         else if (ch == 'C'){
84             m = 4;
85         }
86         else if (ch == 'D'){
87             m = 5;
88         }
89         else if (ch == 'E'){
90             m = 6;
91         }
92         else if (ch == 'F'){
93             m = 7;
94         }
95         else if (ch == 'G'){
96             m = 8;
97         }
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
98     else if (ch == 'H'){
99         m = 9;
100     }
101     else if (ch == 'I'){
102         m = 10;
103     }
104     else if (ch == 'J'){
105         m = 11;
106     }
107     else if (ch == 'K'){
108         m = 12;
109     }
110     else if (ch == 'L'){
111         m = 13;
112     }
113     else if (ch == 'M'){
114         m = 14;
115     }
116     else if (ch == 'N'){
117         m = 15;
118     }
119     else if (ch == 'O'){
120         m = 16;
121     }
122     else if (ch == 'P'){
123         m = 17;
124     }
125     else if (ch == 'Q'){
126         m = 18;
127     }
128     else if (ch == 'R'){
129         m = 19 ;
130     }
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
130     }
131     else if (ch == 's'){
132         m = 20;
133     }
134     else if (ch == 't'){
135         m = 21;
136     }
137     else if (ch == 'u'){
138         m = 22;
139     }
140     else if (ch == 'v'){
141         m = 23;
142     }
143     else if (ch == 'w'){
144         m = 24;
145     }
146     else if (ch == 'x'){
147         m = 25;
148     }
149     else if (ch == 'y'){
150         m = 26;
151     }
152     else if (ch == 'z'){
153         m = 27;
154     }
155     else if (ch == ' '){
156         m = 28;
157     }
158     if(m == n){
159         c = 1;
160     }
161     else{
162         c = modulo(m,e,n);
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
162         c = modulo(m,e,n);
163     }
164     outFS << c << " ";
165 }
166 inFS.close();
167 outFS.close();
168 }
169 else if (ch == 'd'){
170     inFS.open("Decrypt.txt");
171     ofstream outFS;
172     outFS.open("DecryptOutput.txt");
173     while (inFS >> input) {
174         //inFS >> input;
175         m = modulo(input,d,n);
176         if(m == 2){
177             ch = 'A';
178         }
179         else if (m == 3){
180             ch = 'B';
181         }
182         else if (m == 4){
183             ch = 'C';
184         }
185         else if (m == 5){
186             ch = 'D';
187         }
188         else if (m == 6){
189             ch = 'E';
190         }
191         else if (m == 7){
192             ch = 'F';
193         }
194         else if (m == 8){
```

main.cpp

Decrypt.txt

DecryptOutput.txt




Encrypt.txt

EncryptOutput.txt

```
195     ch = 'G';
196 }
197 else if (m == 9){
198     ch = 'H';
199 }
200 else if (m == 10){
201     ch = 'I';
202 }
203 else if (m == 11){
204     ch = 'J';
205 }
206 else if (m == 12){
207     ch = 'K';
208 }
209 else if (m == 13){
210     ch = 'L';
211 }
212 else if (m == 14){
213     ch = 'M';
214 }
215 else if (m == 15){
216     ch = 'N';
217 }
218 else if (m == 16){
219     ch = 'O';
220 }
221 else if (m == 17){
222     ch = 'P';
223 }
224 else if (m == 18){
225     ch = 'Q';
226 }
```


| | |
|--|--|
| <div style="background-color: #4a7c95; color: white; padding: 2px 5px; margin-bottom: 5px;">main.cpp</div> <div style="margin-top: 5px;"> <div>Decrypt.txt</div> <div>DecryptOutput.txt</div> <div>Encrypt.txt</div> <div>EncryptOutput.txt</div> </div> | <pre> 227 else if (m == 19){ 228 ch = 'R'; 229 } 230 else if (m == 20){ 231 ch = 'S'; 232 } 233 else if (m == 21){ 234 ch = 'T'; 235 } 236 else if (m == 22){ 237 ch = 'U'; 238 } 239 else if (m == 23){ 240 ch = 'V'; 241 } 242 else if (m == 24){ 243 ch = 'W'; 244 } 245 else if (m == 25){ 246 ch = 'X'; 247 } 248 else if (m == 26){ 249 ch = 'Y'; 250 } 251 else if (m == 27){ 252 ch = 'Z'; 253 } 254 else if (m == 28){ 255 ch = ' '; 256 } 257 outFS << ch; 258 } 259 inFS.close(); 260 outFS.close(); 261 } 262 return 0; 263 } </pre> |
|--|--|


The following shows an example with the second listed quote, and the public key (11, 51). Please see below:

Files   

main.cpp


Decrypt.txt

DecryptOutput.txt

Encrypt.txt 




EncryptOutput.txt

Encrypt.txt

 saved ▼

1

THE GREATEST GLORY IN LIVING LIES NOT IN NEVER FALLING BUT IN RISING
EVERY TIME WE FALL


Files   

main.cpp


Decrypt.txt

DecryptOutput.txt

Encrypt.txt




EncryptOutput.txt 

EncryptOutput.txt


 saved ▼

1

30 15 39 46 2 25 39 8 30 39 41 30 46 2 4 16 25 32 46 37 9 46 4 37 5
37 9 2 46 4 37 39 41 46 9 16 30 46 37 9 46 9 39 5 39 25 46 31 8 4 4
37 9 2 46 24 28 30 46 37 9 46 25 37 41 37 9 2 46 39 5 39 25 32 46 30
37 44 39 46 48 39 46 31 8 4 4

Files   

main.cpp


Decrypt.txt 

DecryptOutput.txt

Encrypt.txt

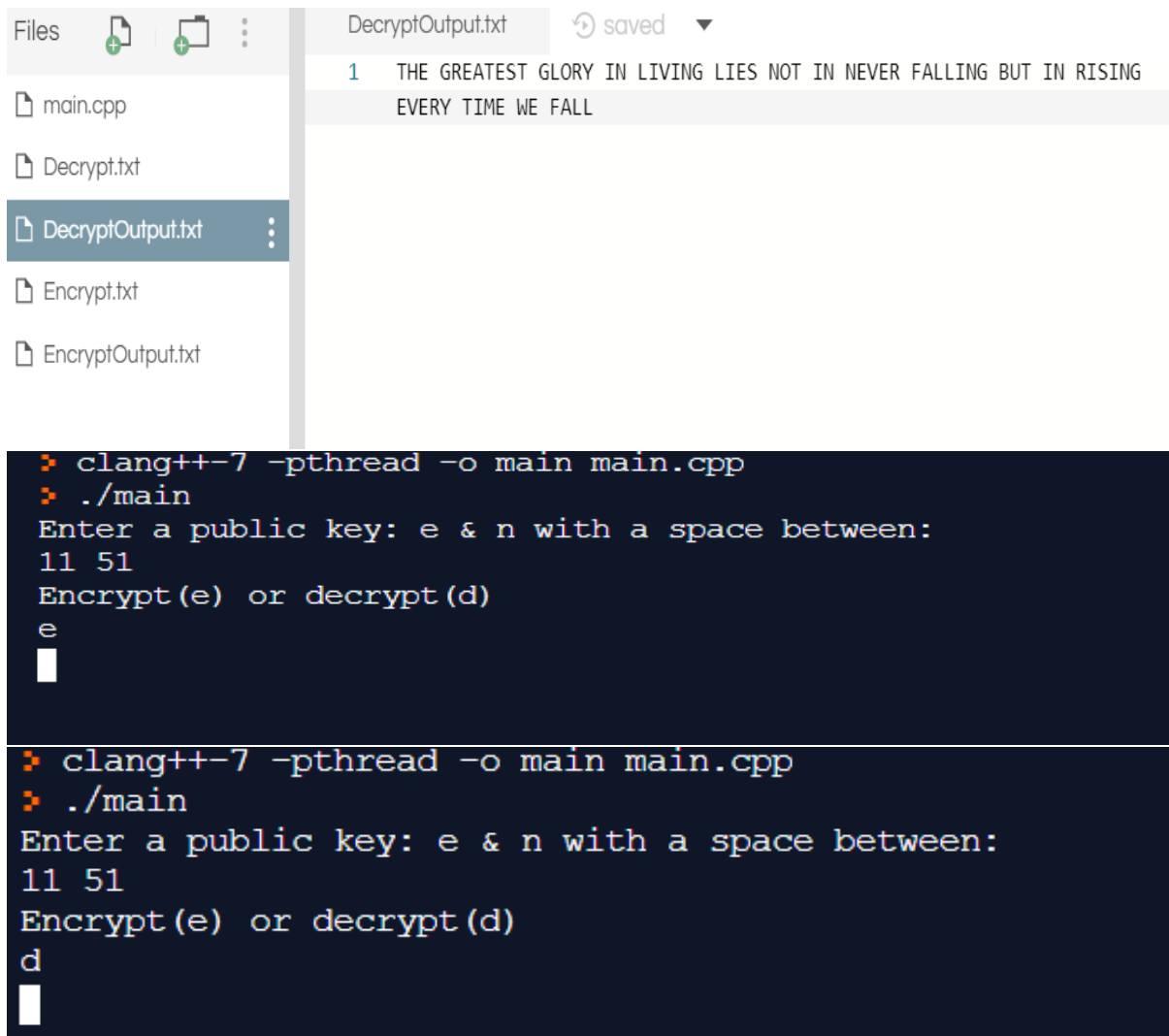
EncryptOutput.txt





Decrypt.txt

 saved ▼

1

30 15 39 46 2 25 39 8 30 39 41 30 46 2 4 16 25 32 46 37 9 46 4 37 5
37 9 2 46 4 37 39 41 46 9 16 30 46 37 9 46 9 39 5 39 25 46 31 8 4 4
37 9 2 46 24 28 30 46 37 9 46 25 37 41 37 9 2 46 39 5 39 25 32 46 30
37 44 39 46 48 39 46 31 8 4 4



```
Files    DecryptOutput.txt  saved ▼
```

1 THE GREATEST GLORY IN LIVING LIES NOT IN NEVER FALLING BUT IN RISING
EVERY TIME WE FALL

```
❏ clang++-7 -pthread -o main main.cpp
❏ ./main
Enter a public key: e & n with a space between:
11 51
Encrypt (e) or decrypt (d)
e
█
```

```
❏ clang++-7 -pthread -o main main.cpp
❏ ./main
Enter a public key: e & n with a space between:
11 51
Encrypt (e) or decrypt (d)
d
█
```