Alejandro Sherman SID 862062898

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 the 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

```cpp
#include<iostream>
#include <fstream>
#include<math.h>

using namespace std;

void primesfinder(int& p, int& q, int n){
for(int i = 2; i < n; ++i){
  if((n % i) == 0){
    p = i;
    q = n/p;
  }
  }
}


int modulo(int c, int d, int n){
  if (c == 0){
    c = 2;
  }
    long long x=1, y=c;
    while (d > 0) {
        if (d%2 == 1) {
            x = (x*y) % n;
        }
        y = (y*y) % n;
        d /= 2;
    }
    return x % n;
}


int d_finder(int phi, int e){
```

Files

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

main.cpp saved

```cpp
int d_finder(int phi, int e){
  for (int k = 1; k < 100000; ++k){
  for (int i = 1; i < 100000; ++i){
    if ((i*e) == (1 + (k*phi))){
      return i;
    }
  }
  }
  return -1;
}

int main(){
ifstream inFS;
int filenumber;

int e;
int n;
    cout << "Enter a public key: e & n with a space between:" <<endl;
    cin >> e;
    cin >> n;
int p = 0;
int q = 0;
int d;
int totient;
//int k = 4;
int input;
int m;
int c;

primesfinder(p,q,n);
totient = (p-1)*(q-1);

//d = (1 + (k*totient))/e;
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```cpp
//d = (1 + (k*totient))/e;
d = d_finder(totient, e);
char ch;
cout << "Encrypt(e) or decrypt(d)" << endl;
cin >> ch;

if (ch == 'e'){
inFS.open("Encrypt.txt");
ofstream outFS;
outFS.open("EncryptOutput.txt");
while (inFS.get(ch)) {
        //inFS.get(ch);
      if(ch == 'A'){
        m = 2;
      }
      else if (ch == 'B'){
        m = 3;
      }
      else if (ch == 'C'){
        m = 4;
      }
      else if (ch == 'D'){
        m = 5;
      }
      else if (ch == 'E'){
        m = 6;
      }
      else if (ch == 'F'){
        m = 7;
      }
      else if (ch == 'G'){
        m = 8;
      }
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```cpp
else if (ch == 'H'){
  m = 9;
}
else if (ch == 'I'){
  m = 10;
}
else if (ch == 'J'){
  m = 11;
}
else if (ch == 'K'){
  m = 12;
}
else if (ch == 'L'){
  m = 13;
}
else if (ch == 'M'){
  m = 14;
}
else if (ch == 'N'){
  m = 15;
}
else if (ch == 'O'){
  m = 16;
}
else if (ch == 'P'){
  m = 17;
}
else if (ch == 'Q'){
  m = 18;
}
else if (ch == 'R'){
 m = 19 ;
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
        }
        else if (ch == 'S'){
            m = 20;
        }
        else if (ch == 'T'){
            m = 21;
        }
        else if (ch == 'U'){
            m = 22;
        }
        else if (ch == 'V'){
            m = 23;
        }
        else if (ch == 'W'){
            m = 24;
        }
        else if (ch == 'X'){
            m = 25;
        }
        else if (ch == 'Y'){
            m = 26;
        }
        else if (ch == 'Z'){
            m = 27;
        }
        else if (ch == ' '){
            m = 28;
        }
    if(m == n){
        c = 1;
    }
    else{
        c = modulo(m,e,n);
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
            c = modulo(m,e,n);
          }
          outFS << c << " ";
        }
    inFS.close();
    outFS.close();
    }
    else if (ch == 'd'){
    inFS.open("Decrypt.txt");
    ofstream outFS;
    outFS.open("DecryptOutput.txt");
    while (inFS >> input) {
            //inFS >> input;
            m = modulo(input,d,n);
          if(m == 2){
            ch = 'A';
          }
          else if (m == 3){
            ch = 'B';
          }
          else if (m == 4){
            ch = 'C';
          }
          else if (m == 5){
            ch = 'D';
          }
          else if (m == 6){
            ch = 'E';
          }
          else if (m == 7){
            ch = 'F';
          }
          else if (m == 8){
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
        ch = 'G';
    }
    else if (m == 9){
        ch = 'H';
    }
    else if (m == 10){
        ch = 'I';
    }
    else if (m == 11){
        ch = 'J';
    }
    else if (m == 12){
        ch = 'K';
    }
    else if (m == 13){
        ch = 'L';
    }
    else if (m == 14){
        ch = 'M';
    }
    else if (m == 15){
        ch = 'N';
    }
    else if (m == 16){
        ch = 'O';
    }
    else if (m == 17){
        ch = 'P';
    }
    else if (m == 18){
        ch = 'Q';
    }
```

main.cpp

Decrypt.txt

DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

```
        else if (m == 19){
          ch = 'R';
        }
        else if (m == 20){
          ch = 'S';
        }
        else if (m == 21){
          ch = 'T';
        }
        else if (m == 22){
          ch = 'U';
        }
        else if (m == 23){
          ch = 'V';
        }
        else if (m == 24){
          ch = 'W';
        }
        else if (m == 25){
          ch = 'X';
        }
        else if (m == 26){
          ch = 'Y';
        }
        else if (m == 27){
          ch = 'Z';
        }
        else if (m == 28){
          ch = ' ';
        }
      outFS << ch;
    }
  inFS.close();
  outFS.close();
  }
  return 0;
  }
```

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

```
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

```
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

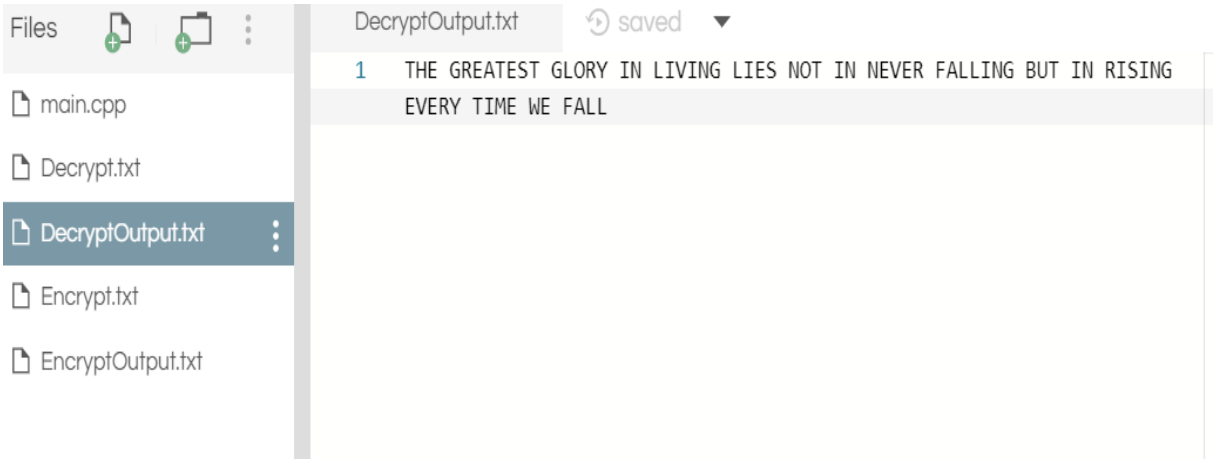
DecryptOutput.txt

Encrypt.txt

EncryptOutput.txt

Decrypt.txt saved

```
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
```




```
> 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
```