Relatório Efolio A

No efolio A, foi-nos pedidos que fizéssemos a implementação de um algoritmo de cifra simétrica e outro assimétrica. De minha escolha, optei por fazer a implementação das cifras em projetos separados, sendo mais fácil a organização dos mesmos códigos.

Uma escolha também minha, foi implementar a hipótese de realizar o processo de encriptação e posteriormente, a desencriptação através de um ficheiro de texto.

Para a execução do código, preferi a linguagem Python, com a qual já desenvolvi alguns trabalhos e sendo uma linguagem com que estou mais a vontade para o fazer.

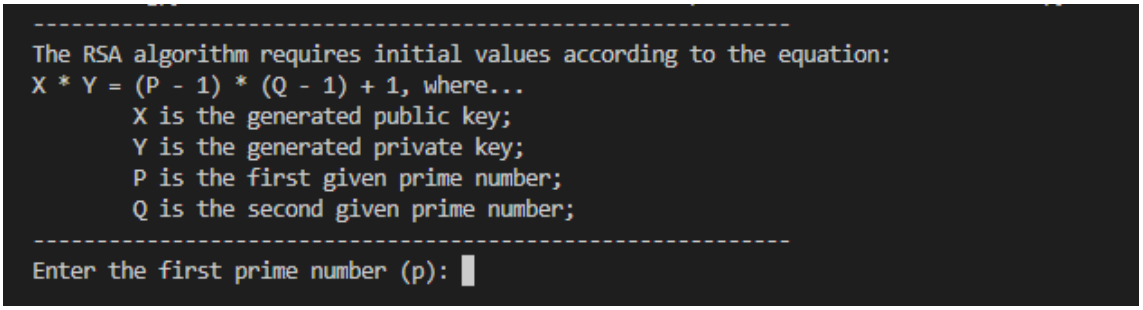
Decidi não usar as bibliotecas criptográficas disponíveis, de modo que, o código executa sem ter as bibliotecas, conforme vou mostrar posteriormente.

A paste que enviei, tem 4 ficheiros, um ficheiro hey.txt onde tem frases e números para ambos os programas testarem os seus algoritmos. Um ficheiro .pdf sendo o relatório, e dois ficheiros .py, sendo estes os algoritmos de encriptação.

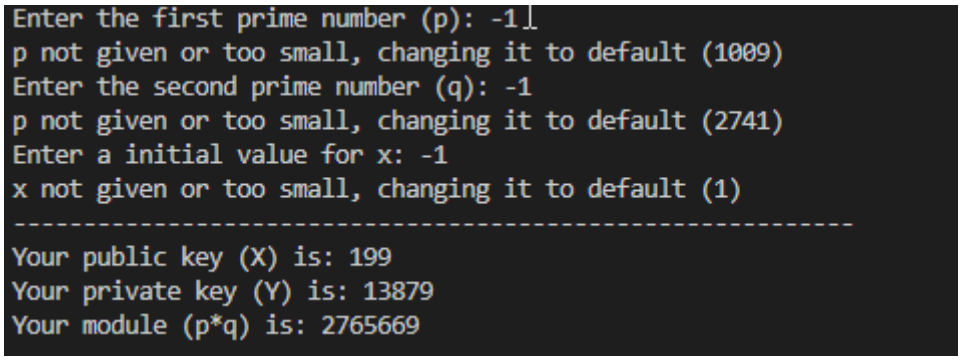
No ficheiro assimetric.py, encontra-se o algoritmo de encriptação assimétrico. Este código começa por ter uma função que analisa se dois números são números Primos ou não, uma vez que decidi trabalhar com o modelo de encriptação RSA.

Esta função que analisa se dois números são primos é chamada na função setup, onde o User tem a hipótese de escolha dos dois números primos, e do valor de x. Caso o User não escolha números primos, será dado um erro.

Caso o user escolha uns números não permitidos “-1” o programa irá dar uns números por default tanto para o P como para o Q como para o X.

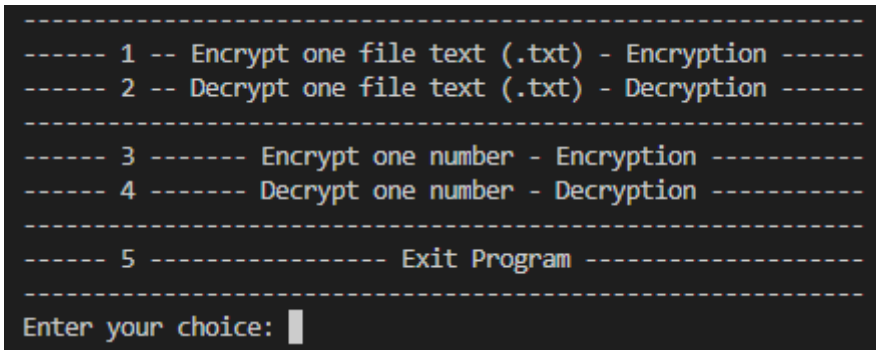


Img 1- Inicialização do código.



Img 2 – Escolha dos números primos para gerar as key.

De seguida o programa executa um menu, onde poderá escolher a opção que deseja fazer.

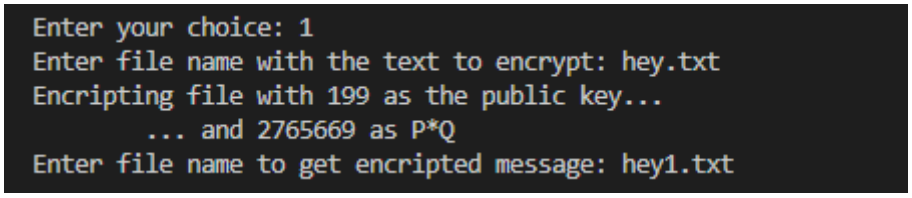


Img 3 – Menu encriptação Assimétrica

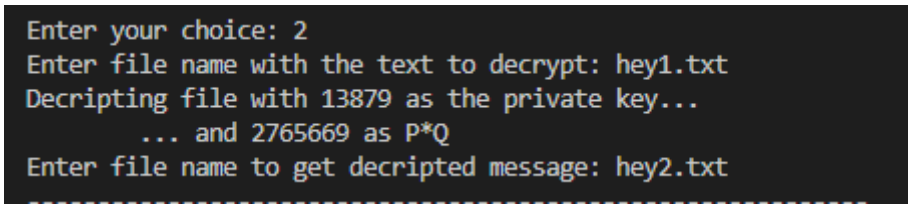
Ao escolher a opção, irá direcionar para as funções de encriptação de ficheiros de texto ou de números, ou para as funções de desencriptação de um ficheiro ou número.

O próprio programa guarda os valores da public key, private key e do module para a execução de todas as tarefas.

Na função de encriptar, o programa analisa cada char do ficheiro de texto, transformando essa char num número e separando cada char por um hifen “-“, de modo ao fazer a desencriptação o programa consegue ver quais as letras usando a mesma forma, só alterando a chave, visto que a encriptação é feita com a public key e a desencriptação é feita com a private key.



Img 4 – Processo de encriptação.

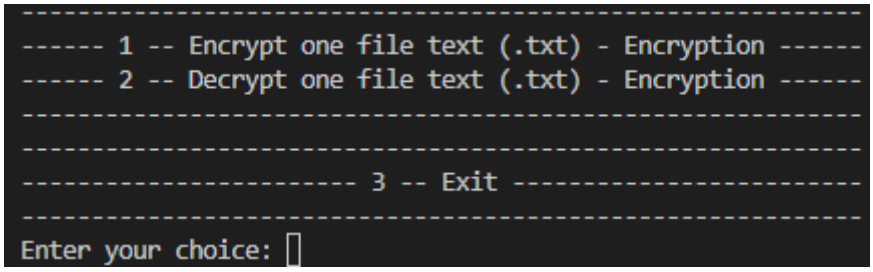


Img 5 – Processo de desencriptação.

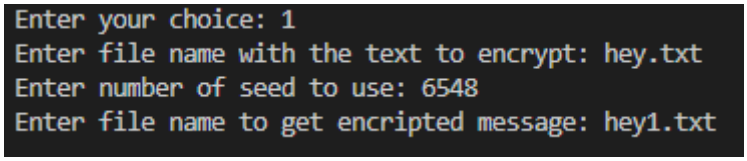
No ficheiro simetric.py, encontra-se o algoritmo de encriptação simétrico.

Este algoritmo apesar de parecer idêntico ao algoritmo assimétrico, é diferente no modo de encriptação e desencriptação, visto que apenas usa uma chave para a encriptação e a mesma chave para a desencriptação. Neste projeto o nome dessa chave é Seed, que é dada pelo User quando assim é pedido pelo programa.

Funciona num sistema de menu também, onde pode ser escolhido o que fazer com o programa, sendo encriptação de ficheiros de texto e desencriptação de ficheiros de texto.



Img 6 – Menu encriptação Simétrica



Img 7 – Escolha da Hipótese, escolha de Seed e os nomes dos ficheiros

Neste algoritmo, caso a desencriptação não use o mesmo Seed da Encriptação, o programa não conseguirá desencriptar corretamente, não obtendo assim a mensagem original, sendo na mensagem original ter letras e números de modo a encriptar tudo o que o ficheiro de texto tem.