

- 1) O programa possui uma função `hex_to_64`. A entrada dela é uma string em hexadecimal. Nele, possui uma lista chamada `convert_list` que possui todos os caracteres de base 64. A `hex_list` é uma lista que irá guardar cada valor da string de entrada de forma que cada elemento já esteja no seu respectivo valor decimal. O `try` serve para verificar caso o valor é uma letra ou não. Caso seja, ele vai converter o valor para o hexadecimal correspondente. Caso não seja, só irá converter de string para int. Depois, criei uma lista que converte cada algarismo para o seu respectivo binário. Por fim, existem 3 casos: caso termine com "=", "== ou sem ". Isso foi feito através de um cálculo feito através do módulo. Caso seja divisível por 6, então não terá ". Caso o resto da divisão por 6 seja 4, terá um ". Caso o resto seja 2, terá "=".
- 2) Primeiro, a função verifica se é possível fazer o XOR entre ambas as entradas. Caso não seja, irá retornar -1. Caso seja, ele cria uma lista que irá salvar o resultado do XOR de cada algarismo. Primeiro, é feita a conversão de cada hexa para o seu correspondente na base 2. Depois disso, é feito o XOR e esse resultado é salvo numa lista. Por fim, converte cada byte para o seu correspondente algarismo.
- 3) Essa questão foi separada em algumas funções para facilitar o trabalho. A `single_xor` recebe uma entrada, o caractere e o tamanho de 1 byte (para caso o byte não seja de 8 bits, algo que tinha sido utilizado na versão inicial quando eu ainda não tinha entendido o problema). O `join_hex` serve para juntar dois algarismos hexa para poder formar um byte. Nela, é chamada a `hex_to_byte` que serve para converter cada hexa para base 2 e retorna os dois juntos. Por fim, a `join_hex` retorna a entrada inteira. A `get_byte` serve para retornar um hexa_decimal na sua forma binária. `Byte_to_ascii` serve para converter um número na base 2 para o seu correspondente na tabela ASCII. A função `decifrar` tem dois parâmetros que são: a entrada e o caractere que está sendo testado. Esse teste é baseado na frequência dos caracteres no inglês. Ou seja, os que são mais utilizados serão testados primeiro. Em seguida, é criado um dicionário que salvará a frequência de cada caractere na frase. Aquele que mais se repetir tem a maior chance de ser o carácter mais frequente do inglês. Depois, é feito um XOR entre esse caractere e o seu correspondente no inglês para descobrir qual foi o caractere utilizado para fazer o XOR na entrada. Por fim, é feito o XOR novamente com esse novo caractere e o retorna o resultado da entrada descriptografada.