The encryption used in the encrypt_classical.py is chain block chaining.
The algorithm needs an initialization vector and a key for encryption. Initialization vector can be made public after it's used for encryption, but it shouldn't be predictable.
Blocksize of 64bits is considered here.
The algorithm:

-> The passphrase is used a initialization vector. It's initialized with zero values and it's size is equal to the block size.
- If the passphrase is smaller than block size then it would be padded with zero. This isn't as secure as having passphrase greater than or equal to the Blocksize as for the rest of the bits the c1 block would just have xor of key and plaintext.
- If the passphrase is larger than block size then it would be converted to 64bits by repeatedly xoring the excess bits with the initial bits of passphrase.
- Security issue : If passphrase of size 64bits is repeated even number of times, ultimately the initialization vector would be a block of 64 0's.
    e.g. abcdefghabcdefgh This would result in IV being
        abcdefgh xor abcdefgh = 64 0's
      If the passphrase is palindrome of 16bit blocks then the IV would be all 1's
    e.g. abcdefghhgfedcba This would result in IV being
        abcdefgh xor hgfedcba = 64 1's

-> The key is also converted to 64bit block using the same method as above
- Faces the same security issues.
-If weak key is chosen like above, then the blocks would result into
    c1 = iv xor b1, c2 = c1 xor b2 and so on if key results in being 0.

-> The encryption is carried out as follows,
- c1 = IV xor b1 xor key
- c2 = c1 xor b2 xor key
- c3 = c2 xor b3 xor key

The final cipher would be c1c2c3....

-> If last block of plain text is less than 64bits. Part of key can be discovered using padding attack.

For the crack_classical file.
The cipher is converted into MTP problem by performing following operations,
c1 xor c2, c1 xor IV, c2 xor c3 and so on.
This results in,
b2 xor key, b1 xor key, b3 xor key respectively.

Since, the problem is now converted into Multi time pad problem, it's solved using the same technique used in problem 2.2 of the assignment.