Q1.

Plaintext: And I shall remain satisfied, and proud to have been the first who has ever enjoyed the fruit  of his writings as fully as he could desire; for my desire has been no other than to deliver  over to the detestation of mankind the false and foolish tales of the books of chivalry, which,  thanks to that of my true Don Quixote, are even now tottering, and doubtless doomed to fall  for ever. Farewell.

Ciphertext Using key ‘-3’: Xka F pexii objxfk pxqfpcfba, xka molra ql exsb ybbk qeb cfopq tel exp bsbo bkglvba qeb corfq  lc efp tofqfkdp xp criiv xp eb zlria abpfob; clo jv abpfob exp ybbk kl lqebo qexk ql abifsbo  lsbo ql qeb abqbpqxqflk lc jxkhfka qeb cxipb xka cllifpe qxibp lc qeb yllhp lc zefsxiov, tefze,  qexkhp ql qexq lc jv qorb Alk Nrfulqb, xob bsbk klt qlqqbofkd, xka alryqibpp alljba ql cxii  clo bsbo. Cxobtbii.

Q2.

Ciphertext: Vg jbhyq frrz gung, nf ur rknzvarq gur frireny cbffvovyvgvrf, n fhfcvpvba pebffrq uvf zvaq: gur zrzbel bs ubj ur uvzfrys unq orunirq va rneyvre qnlf znqr uvz nfx jurgure fbzrbar zvtug or  uvqvat ure sebz gur jbeyq

I created a method which can brute-force the key for an encrypted Caesar cipher by running the decrypt process with every possible key shift i.e. the key shifts between 1 and 26. The method would check the potentially decrypted message with each key shift to see if all the substrings in the potentially decrypted text was valid English, using the pyEnchant plugin to achieve this. It would then return the decrypted text where this was the case. The key shift was found to be 13.

Plaintext decrypted using key ‘13’: It would seem that, as he examined the several possibilities, a suspicion crossed his mind: the memory of how he himself had behaved in earlier days made him ask whether someone might be hiding her from the world

Q3.

Plaintext: I shall (from now on) select and take the ingots individually in my own yard, and I shall  exercise against you my right of rejection because you have treated me with contempt.

Cipher-text with password ‘PASSWORD’:

X szshz (wudm fgs ce) vtlwup oeg iacw pvv lcgglo wegxvavqocon if eu cnq najv, wbu L hhsdh sohgcaka oxdxnkl ucl pn raydh fi gebwyhzrc bwuwijh nom zwjv wgeslar dh lilz ycewtmhl.

Q4.

Knowing that Thursday with a capital ‘T’ was a piece of plaintext I could conclude that the text would decrypt to English and that there was only 1 capitalized word in the entire message. In addition to this I found that “Yhwvtroi”, the only capitalized substring in the cipher-text, has the same number of character as “Thursday”, and so could deduce that ‘Yhwvtroi’ was the encrypted version of Thursday. Following on from this, I concluded that logically a month would be the most likely word to fit in between the two-digit number which most likely represented the date, “28” and year “2016”, and judging from the fact that this word was only 4 letters, I deduced that it would either be June or July. When cross referencing these two substrings of the cipher-text against a Vigenere cipher graph, I found that this would spell out ‘FacebookPass’. However, when I tried to use this as the key to decrypt, it did not fully decrypt, so I deduced that the remainder of the key was ‘word’ (FacebookPassword). The message then fully decrypted to:

“thursday, 28 july 2016 ­ ten years of reign having been completed, king piodasses made known the doctrine of piety to men; and from this moment he has made men more pious, and everything thrives throughout the whole world. and the king abstains from killing living beings, and other men and those who are huntsmen and fishermen of the king have desisted from hunting. and if some were intemperate, they have ceased from their intemperance as was in their power; and obedient to their father and mother and to the elders, in opposition to the past also in the future, by so acting on every occasion, they will live better and more happily”.