Uses Python 2.7

Diffie-Hellman Key Arrangement Usage:

1. Run “Diffie-Hellman Key Arrangement.py”
2. Enter a prime modulus suitable for the arrangement. E.g. ‘23’
3. Enter a suitable base. E.g. ‘5’
4. Enter your secret integer. E.g ‘6’
5. Value of A is then calculated. E.g ‘8’
6. Meanwhile, the person on the other end of the arrangement will have chosen their secret integer and used the same prime modulus and base to compute their value, B
7. Once the value for B has been found and communicated to user 1 they can then enter ‘y’ for the question “Do you have a value for B”
8. Enter the value for B. E.g. ‘19
9. Secret value can then be calculated. E.g. ‘2’

RSA Usage:

1. Run RSA.py
2. First, a public-private key pair must be generated. Enter ‘g’ to generate a random public-private key using relatively small primes. E.g. “Private key (d,n) is: (90737, 123559)
3. Public key (e,n) is: (65537, 123559)”
4. Now that a public-private key pair has been generated, user can enter ‘e’ to begin encryption process. But take not of the generated public-private key pair in the previous step.
5. Enter the message you wish to encrypt. E.g. “Hello world!”
6. Enter the value for e, followed by the value for n, the public key paring. E.g. ‘65537’ followed by ‘123559’
7. The encryption process will then split the message into a block of characters, the converting each of the characters into their ordinal representation, then encrypting this value, then padding each of these encrypted values into 8 bit blocks and appending each block to the cipher text array. E.g. ‘751820033471300317031003170310034351600310302102970790034351600385114003170310039863100361588003’
8. The program will then return to the main menu and the user can enter ‘d’ to begin decryption process.
9. The program will then prompt you to enter the cipher text. E.g. 751820033471300317031003170310034351600310302102970790034351600385114003170310039863100361588003
10. It will then prompt you to enter the value for d, which was generated in step 2. E.g. 90737
11. It will then prompt you to enter the value for n, which was generated in step 2. E.g. 123559
12. The decryption process will then decrypt the cipher text, first splitting it into 8 bit blocks, then un-padding the each of the blocks and running the decryption process on each block and finally converting the decrypted value back into its character representation. E.g. Hello world!