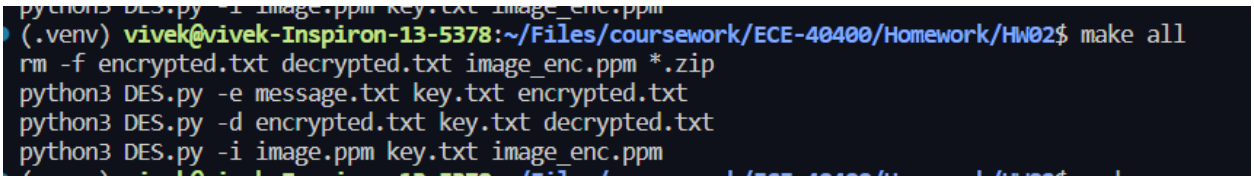
Homework 2:

rpanchag

Vivek Panchagnula

0033584584

Commands used to generate output of Homework:



Problem 1:

Hex dump of Encrypted File:

“0c46d7cd5b7efc319691493448bb36733af8d5e4da962e15e85db329c5031857a154f62cbfb7c82d298c9
456ef29adb8e86cc51ae7f025097f513677406336598e0f3f1f0c5ecaf0b55649222b19a27da886fa8c4d2b9
e0e88a2745b99e6bbb4658cd9fd3606e05d11919eddd39723e333aa813ebd9a9ae6810271c9d634cba82
9e1b7a82bd994073d054e62a79d8bbd1ebe00d2288b8c05b0f4d5ec799e3f7d5db8b04a23106d0151c6fe
a8bd1826a92e611e73a1bc4949ed703d0174516196ef7faed8a411c7efc9b11b6b44fa864c7692c80a7ac2
dc6f5d467e8b6588845f5c8c1f4493c9d94f3af8d5e4da962e1580d4d42e93e281c6aab31eec856fead76a9
6c9d84c4a3fce61ded79fdd9a943cb446a58d881c211b5ba21a1dc81659123283460d36ca20cba580ebd5
1188824724ec416aebeff0d01d2be942433af7679b2d5d55a4b8c931151283e60d8e99e90701d26b28a13
9a46c209a2a93f6250b902ff25ee8aa0f56ea075b13c3ca4dbd985da7338582b48b412c33ce01dc4bcbb7c
b9a3e905deb0caf473c5b801aa2872c62d06d015b9b7aba88a48889f7b2cd6602ec4311480ef124adff91a
834630b41c2f4d29769ca093ec31ee4779264af3a6ecd51cc098d3acfb1c5fdeff53a694ea26c872220eb2c7
5894e9e10b1beba091a61279d20154b4c46eda9c3d6b6df07eaaa1dc93f98246eefeb34d8ea72bef755805
5080ed4d73afe523bb6723e79ba8eae813579fc2f74a2a64cdf2484bc8267b7c0b0cc28ab5ba21a1dc8165
912c99d911d997a8e829853c23bcd8681544a3bc6ea2a56ae5844873d757d272114000874af4a2adff08a8
24e0c1b8dbbb72a02f86fb4c95668b5bdcb5c3c3d3fc3545d14e6459f7d2b7050edc71e4c58ad593b284e6
fee59f41bf13fddf342694530d4e70c288d9a61e3515a37674fbb7bc98730a9d700b5c8d332cc75c1a41e39
a2ae33cb95d43e92b3f168a97488f8a7cfbe9993019259ed8cfdc1cddb6e60cb40803c3e931e1278d85ae8
0815e10b3a7496e30b24e6b996e2400cad3f3999fdab7d3bcf897a9a376e85932b9d711e634dcf3a756b2a
93165df4a192bf0d0a271415986d5e1dbd019250095819c5e0b55b095bbb94a00a009e6c9e6a998598c2f
98075a8861a43710dbd6cb63a94d66c2d4d779ead4200ef8f58a2d2c3ab25ccd2fec9c8489ab4b8bb1c95b
3b7da5d9b5eb50e9733bdf981112601bec9feb807ef32f154f825a870d7ff1ec081545d343c085bb0bc7b2b
ee895410488ad30eaec469d6170b2a502a616b4b55e49e7ab3517db4259cc90e91b70e232ec1f8a1ea85a
1b4d4c63fa94fc1b80e7005183f54ace18926dbf3330252ca26895d60dd71”

Decrypted File:

“Scuderia Ferrari is the racing division of luxury Italian auto manufacturer Ferrari and the racing team that competes in Formula One racing. The team is also known by the nickname "The Prancing Horse", in reference to their logo. It is the oldest surviving and most successful Formula One team, having competed in every world championship since the 1950 Formula One season. The team was founded by Enzo Ferrari, initially to race cars produced by Alfa Romeo. By 1947 Ferrari had begun building its own cars. Among its important achievements outside Formula One are winning the World Sportscar Championship, 24 Hours of Le Mans, 24 Hours of Spa, 24 Hours of Daytona, 12 Hours of Sebring, Bathurst 12 Hour, races for Grand tourer cars and racing on road courses of the Targa Florio, the Mille Miglia and the Carrera Panamericana. The team is also known for its passionate support base, known as the tifosi. The Italian Grand Prix at Monza is regarded as the team's home race.”

Explanation:

I will explain the relevant parts of the DES object in my code

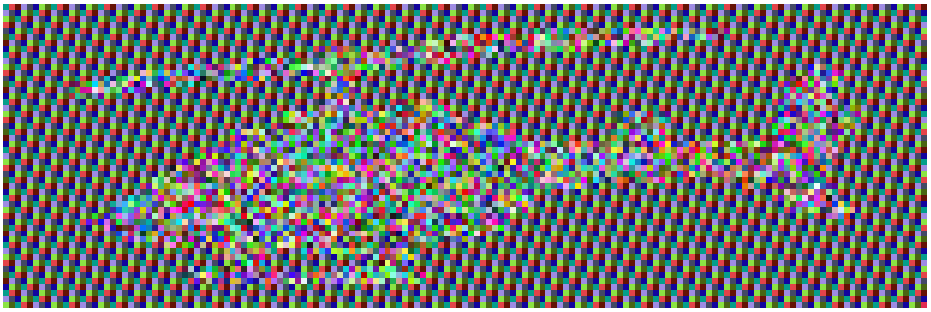
__init__ : This initializes the object and key, it uses the get_encryption_key function that is adapted from lecture notes to set the key variable

Encrypt: This function initializes the key from self.key, which is what we initialized in the __init__ earlier. The function then generates the round keys using the lecture code and initializes the BV and loads the output file. The program pads any chunks less than 64 bits to be 64 bits before proceeding. Next, while taking 64 size chunks or until the file comes to an end, the program performs the permutation, and substitution steps for the right-hand side before switching them for the next loop. This is the Feistel Algorithm being done on each of the 32 bits. This uses the substitute function which was adapted from the lecture code. The encrypted code is written as a direct binary file, and I have read the hex dump of this file to compare against the first round given output for the sake of validation and it all matches up.

Decrypt: This is the same as the encrypt function except the round keys were reversed. This function reads the binary and performs the Feistel Algorithm the same way as encrypt and outputs a coherent string with padded nulls at the end which was added as part of the padding process in Encrypt.

The remaining functions are adaptations from lecture code.

Problem 2:



Image_encrypt: The code in the image_encrypt function is largely the same as the encrypt function in problem 1, except for a few small modifications. The function passes the first three lines of the PPM file directly to the out file since the header should be transported as is. It then removes the first 112 bits

from the BV that was read, which represent these first three lines take up the first 112 bits. It then encrypts the image before writing it to the new file.