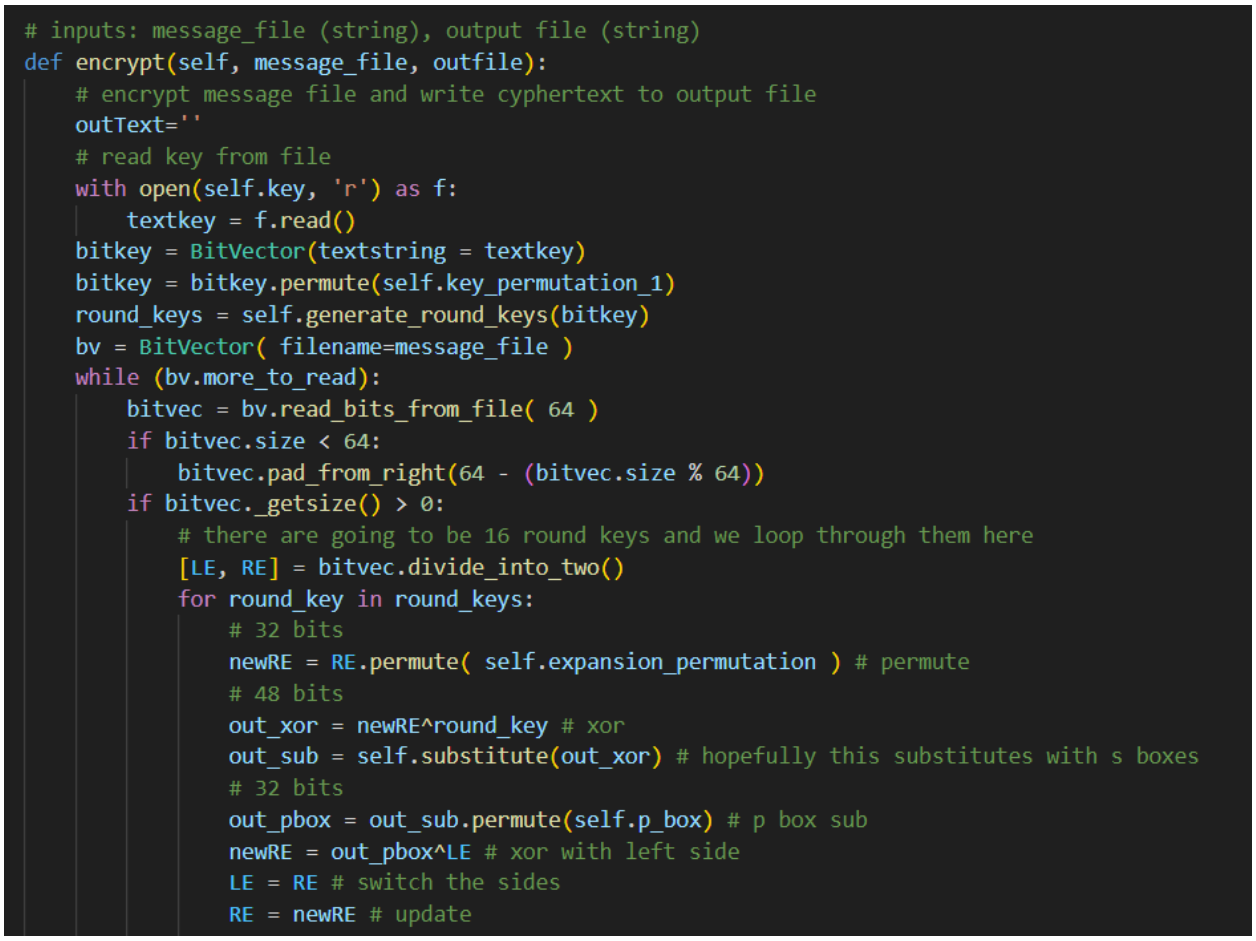
Aubrey Gatewood

Init: I put the P-box, key, permutation 1 and 2, expansion permutation, and s-boxes as attributes of the DES class. These I pulled from the lecture 3 starter code. I also based my solution for encrypt, decrypt, and image encrypt heavily on the sample code from the lecture notes.

Problem 1 encrypt:



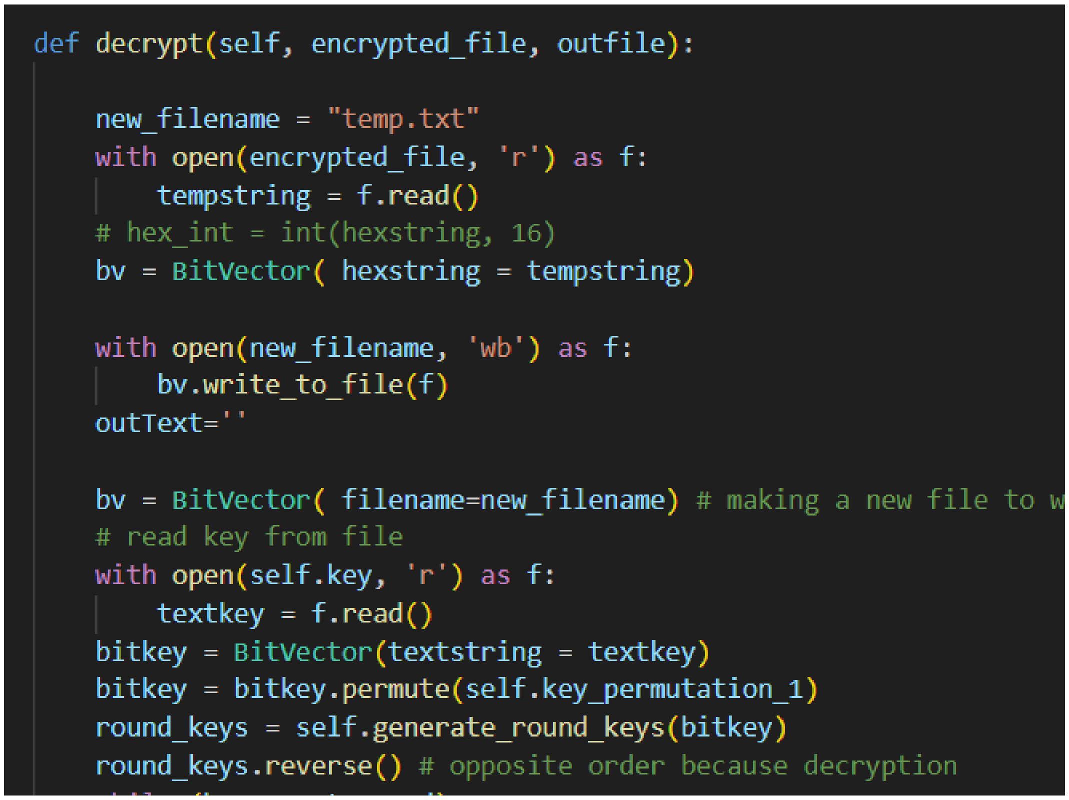
```python
# inputs: message_file (string), output file (string)
def encrypt(self, message_file, outfile):
    # encrypt message file and write cyphertext to output file
    outText=''
    # read key from file
    with open(self.key, 'r') as f:
        textkey = f.read()
    bitkey = BitVector(textstring = textkey)
    bitkey = bitkey.permute(self.key_permutation_1)
    round_keys = self.generate_round_keys(bitkey)
    bv = BitVector( filename=message_file )
    while (bv.more_to_read):
        bitvec = bv.read_bits_from_file( 64 )
        if bitvec.size < 64:
            bitvec.pad_from_right(64 - (bitvec.size % 64))
        if bitvec._getsize() > 0:
            # there are going to be 16 round keys and we loop through them here
            [LE, RE] = bitvec.divide_into_two()
            for round_key in round_keys:
                # 32 bits
                newRE = RE.permute( self.expansion_permutation ) # permute
                # 48 bits
                out_xor = newRE^round_key # xor
                out_sub = self.substitute(out_xor) # hopefully this substitutes with s boxes
                # 32 bits
                out_pbox = out_sub.permute(self.p_box) # p box sub
                newRE = out_pbox^LE # xor with left side
                LE = RE # switch the sides
                RE = newRE # update
```

This is most of my encrypt function. I make the bitkey based on the text input file, then permute it with permutation 1. Then I make the round keys based off of that. I make a bitvector off of the file with the message in it, and read from that vector in chunks of 64 bits. Then I divide it into 2 outside of the round key for loop so that it doesn't split each time it loops. Then inside the for loop, I do the expansion permutation on the right side, xor with the round key, do the substitution, then do permutation with the p box. I xor the output from the p-box and the left side and make that result the new right side. Then I make the left side the old right side, and update the right side. Outside the for loop, I concatenate the left and right sides together, and add it to a bitvector in hex. Outside of the while loop (when the reading is done, I write that result to the output file.

Here is my encrypted output:

```
0c46d7cd5b7efc319691493448bb36733af8d5e4da962e15e85db329c5031857a154f62cbfb7c82d298c94
56ef29adb8e86cc51ae7f025097f513677406336598e0f3f1f0c5ecaf0b55649222b19a27da886fa8c4d2b9e
0e88a2745b99e6bbb4658cd9fd3606e05d11919eddd39723e333aa813ebd9a9ae6810271c9d634cba829
e1b7a82bd994073d054e62a79d8bbd1ebe00d2288b8c05b0f4d5ec799e3f7d5db8b04a23106d0151c6fea
8bd1826a92e611e73a1bc4949ed703d0174516196ef7faed8a411c7efc9b11b6b44fa864c7692c80a7ac2d
c6f5d467e8b6588845f5c8c1f4493c9d94f3af8d5e4da962e1580d4d42e93e281c6aab31eec856fead76a96
c9d84c4a3fce61ded79fdd9a943cb446a58d881c211b5ba21a1dc81659123283460d36ca20cba580ebd511
88824724ec416aebeff0d01d2be942433af7679b2d5d55a4b8c931151283e60d8e99e90701d26b28a139a
46c209a2a93f6250b902ff25ee8aa0f56ea075b13c3ca4dbd985da7338582b48b412c33ce01dc4bcbb7cb9
a3e905deb0caf473c5b801aa2872c62d06d015b9b7aba88a48889f7b2cd6602ec4311480ef124adff91a834
630b41c2f4d29769ca093ec31ee4779264af3a6ecd51cc098d3acfb1c5fdeff53a694ea26c872220eb2c7589
4e9e10b1beba091a61279d20154b4c46eda9c3d6b6df07eaaa1dc93f98246eefeb34d8ea72bef755805508
0ed4d73afe523bb6723e79ba8eae813579fc2f74a2a64cdf2484bc8267b7c0b0cc28ab5ba21a1dc8165912c
99d911d997a8e829853c23bcd8681544a3bc6ea2a56ae5844873d757d272114000874af4a2adff08a824e
0c1b8dbbb72a02f86fb4c95668b5bdcb5c3c3d3fc3545d14e6459f7d2b7050edc71e4c58ad593b284e6fee
59f41bf13fddf342694530d4e70c288d9a61e3515a37674fbb7bc98730a9d700b5c8d332cc75c1a41e39a2
ae33cb95d43e92b3f168a97488f8a7cfbe9993019259ed8cfdc1cddb6e60cb40803c3e931e1278d85ae808
15e10b3a7496e30b24e6b996e2400cad3f3999fdab7d3bcf897a9a376e85932b9d711e634dcf3a756b2a93
165df4a192bf0d0a271415986d5e1dbd019250095819c5e0b55b095bbb94a00a009e6c9e6a998598c2f98
075a8861a43710dbd6cb63a94d66c2d4d779ead4200ef8f58a2d2c3ab25ccd2fec9c8489ab4b8bb1c95b3b
7da5d9b5eb50e9733bdf981112601bec9feb807ef32f154f825a870d7ff1ec081545d343c085bb0bc7b2bee
895410488ad30eaec469d6170b2a502a616b4b55e49e7ab3517db4259cc90e91b70e232ec1f8a1ea85a1b
4d4c63fa94fc1b80e7005183f54ace18926dbf3330252ca26895d60dd71
```

Decrypt function:



```
def decrypt(self, encrypted_file, outfile):

    new_filename = "temp.txt"
    with open(encrypted_file, 'r') as f:
        tempstring = f.read()
    # hex_int = int(hexstring, 16)
    bv = BitVector( hexstring = tempstring)

    with open(new_filename, 'wb') as f:
        bv.write_to_file(f)
    outText=''

    bv = BitVector( filename=new_filename) # making a new file to w
    # read key from file
    with open(self.key, 'r') as f:
        textkey = f.read()
    bitkey = BitVector(textstring = textkey)
    bitkey = bitkey.permute(self.key_permutation_1)
    round_keys = self.generate_round_keys(bitkey)
    round_keys.reverse() # opposite order because decryption
```

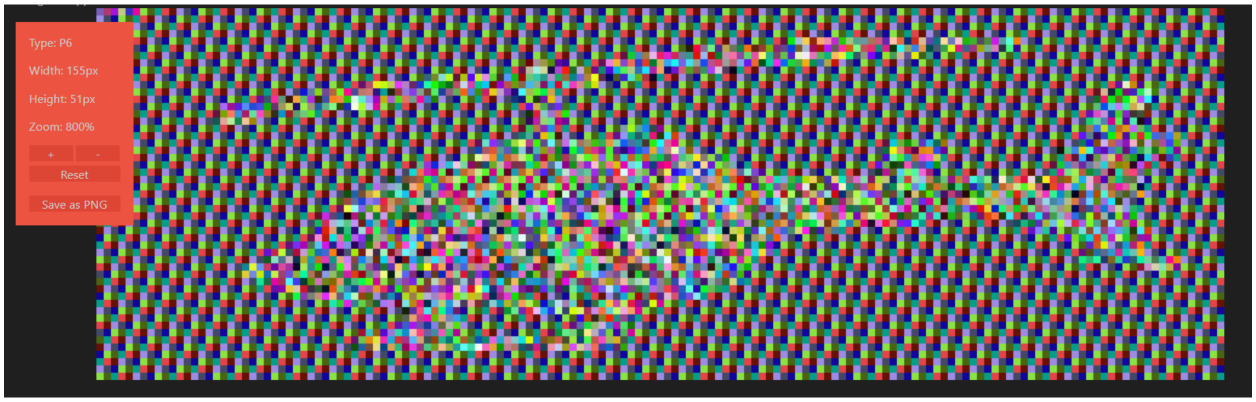
This is the beginning of my decrypt function. Little is different from encrypt. I make a new file because that is the only way I could figure out to manipulate the bitvector so that it could be fed exactly into the encryption code with only the round key order reversed. I read from the file, make a bitvector with the hex string input, and write that to a temporary file. Then I'm able to read directly from the temp file for the desired bitvector. The only other change is that I reverse the order of the round keys.

Output: 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.

Problem 2:

For the image encoding I took the text encoding function and copied it over almost verbatim. The only difference was that at the beginning I wrote the first 3 lines (the header) of the image file to the output file, then proceed with encryption.

Output image:


Type: P6
Width: 155px
Height: 51px
Zoom: 800%
+
-
Reset
Save as PNG