Programming Assignment 1

Author: Gabriel Hofer

CSC-325

Instructor: Dr. Karlsson

Due: September 25, 2020

Computer Science and Engineering
South Dakota School of Mines and Technology

Hamming(7,4)

First, we need a message, p to work with.

Listing 1: makeMessage

```
def makeMessage(n):
  return np.random.randint(2, size=(n, 1))
```

Listing 2: encode74

```
def encode(p):
  G74=np.array([ \
     [1,1,0,1], \
     [1,0,1,1], \
     [1,0,0,0], \
     [0,1,1,1], \
     [0,1,0,0], \
     [0,0,1,0], \
     [0,0,0,1]])
  x = np.matmul(G74,p)
  x = x & 1
  return x
```

Listing 3: encode74

```
def makeError(p):
  rdm=random.randint(0,p.shape[0]-1)
  p[rdm,0]=p[rdm,0]^1;
  return p
```

Listing 4: encode74

```
def parityCheck(r):
  H74 = np.array([ \
     [1,0,1,0,1,0,1], \
     [0,1,1,0,0,1,1], \
     [0,0,0,1,1,1,1]])
  z = np.matmul(H74,r)
  z = z & 1
  return z
```

Listing 5: encode74

```
def decodeMessage(r):
  R74 = np.array([ \
    [0,0,1,0,0,0,0], \
    [0,0,0,0,1,0,0], \
    [0,0,0,0,0,1,0], \
    [0,0,0,0,0,0,1]])
  pr=np.matmul(R74,r)
  return pr
```

Listing 6: encode74

```
def correctError(z,r):
  loc=0
  for i in range(0,z.shape[0]):
    loc+=z[i,0]*pow(2,i)
  print("loc: "+str(loc))
  r[loc-1,0]=r[loc-1,0]^1;
  return r
```