def get\_diff\_index\_list(value\_bin1, value\_bin2):

"""

Получить список индексов различающихся битов

"""

diff\_index\_list = []

for index, char\_bin\_items in enumerate(zip(list(value\_bin1), list(value\_bin2)), 1):

if char\_bin\_items[0] != char\_bin\_items[1]:

diff\_index\_list.append(index)

return diff\_index\_list

def encode(source):

"""

Кодирование данных

"""

text\_bin = chars\_to\_bin(source)

result = ''

for chunk\_bin in chunk\_iterator(text\_bin):

chunk\_bin = set\_check\_bits(chunk\_bin)

result += chunk\_bin

return result

def decode(encoded, fix\_errors=True):

"""

Декодирование данных

"""

decoded\_value = ''

fixed\_encoded\_list = []

for encoded\_chunk in chunk\_iterator(encoded, CHUNK\_LENGTH + len(CHECK\_BITS)):

if fix\_errors:

encoded\_chunk = check\_and\_fix\_error(encoded\_chunk)

fixed\_encoded\_list.append(encoded\_chunk)

clean\_chunk\_list = []

for encoded\_chunk in fixed\_encoded\_list:

encoded\_chunk = exclude\_check\_bits(encoded\_chunk)

clean\_chunk\_list.append(encoded\_chunk)

for clean\_chunk in clean\_chunk\_list:

for clean\_char in [clean\_chunk[i:i + 8] for i in range(len(clean\_chunk)) if not i % 8]:

decoded\_value += chr(int(clean\_char, 2))

return decoded\_value

if name == 'main':

source = input('Укажите текст для кодирования/декодирования:')

print('Длина блока кодирования: {0}'.format(CHUNK\_LENGTH))

print('Контрольные биты: {0}'.format(CHECK\_BITS))

encoded = encode(source)

print('Закодированные данные: {0}'.format(encoded))

decoded = decode(encoded)

print('Результат декодирования: {0}'.format(decoded))

encoded\_with\_error = set\_errors(encoded)

print('Допускаем ошибки в закодированных данных: {0}'.format(encoded\_with\_error))

diff\_index\_list = get\_diff\_index\_list(encoded, encoded\_with\_error)

print('Допущены ошибки в битах: {0}'.format(diff\_index\_list))

decoded = decode(encoded\_with\_error, fix\_errors=False)

print('Результат декодирования ошибочных данных без исправления ошибок: {0}'.format(decoded))

decoded = decode(encoded\_with\_error)

print('Результат декодирования ошибочных данных с исправлением ошибок: {0}'.format(decoded))