Алгоритм программы.

1. Определим функцию «noOfParityBits», которая определяет исходя из условия $2^m \ge k + m + 1\;$ и числа информационных разрядов — число избыточных битов.

```
def noOfParityBits(noOfBits):

i = 0

while 2. ** i <= noOfBits + i:

i += 1

return i
```

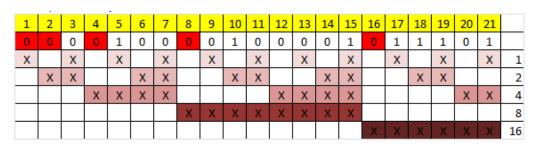
2. Функция «appendParityBits» реализует размещение контрольных символов на местах, кратных степени 2.

```
def appendParityBits(data):
   d = \{\}
   n = noOfParityBits(len(data))
   i = 0
   j = 0
   k = 0
   list1 = list()
   while i < n + len(data):
       if i == (2. ** j - 1):
          list1.insert(i, '0')
          j += 1
       else:
          list1.insert(i, data[k])
          k += 1
       i += 1
   return list1
```

3. Определим состав контрольных символов в функции «hammingCodes».

```
while i < n:
   pos = []
   k = 2. ** i
   j = 1
   total = 0
   while j * k - 1 < len(list1):
      if j * k - 1 == len(list1) - 1:
          lower_index = j * k - 1
          temp = list1[int(lower_index):len(list1)]
       elif (j + 1) * k - 1 >= len(list1):
          lower_index = j * k - 1
          temp = list1[int(lower_index):len(list1)]
       elif (j + 1) * k - 1 < len(list1) - 1:
          lower_index = (j * k) - 1
           upper_index = (j + 1) * k - 1
           temp = list1[int(lower_index):int(upper_index)]
       total = total + sum(int(e) for e in temp)
       pos.extend(temp)
       j += 2
   if total % 2 > 0:
       list1[int(
```

Алгоритм выше, в виде кода, можно представить в виде таблицы, на примере блока, длиной в 16 бит, где знаком «Х» обозначены контролируемые биты.



4. Теперь, проверим решение и приведем скриншоты выполнения программы:

```
E:\PythonYandex\work\venv\Scripts\python.exe E:/P'
input: 1101
K1+1+1+1->K1=1.
K2+1+0+1->K2=0.
K3+1+0+1->K3=0.
|-----|
|позиция символов|-----кодовое слово-----|
|-----|
|____1__K_1___|
|____2___|__K_2___|__0___|
1_____1___1____1
|_____4___|__K_3___|__0___|
1_____5_____1____1____1____1
|_____6____|___0____|
|_____7____|___1____1____1
|-----|
CODE:1010101
E:\PythonYandex\work\venv\Scripts\python.exe E:/P
input: 0101
K1+0+1+1->K1=0.
K2+0+0+1->K2=1.
K3+1+0+1->K3=0.
|-----|
|позиция символов|-----кодовое слово-----| |
|---|---|---|
|____1___K_1____|___0____|
|____2___|__K_2___|__1___|
|_____3____|___0____|
|_____4___|__K_3___|__0___|
|_____5___|___1____|
|_____6____|___0____|
|_____7____|____1____1
|-----|
CODE:0100101
```