

Алгоритм программы.

1. Определим функцию «noOfParityBits», которая определяет исходя из условия $2^m \geq k + m + 1$ 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(
            k) - 1] = '1'

```

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

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	
X		X		X		X		X		X		X		X		X		X		X	1
	X	X			X	X			X	X			X	X			X	X			2
			X	X	X	X					X	X	X	X					X	X	4
							X	X	X	X	X	X	X	X							8
															X	X	X	X	X	X	16

- Теперь, проверим решение и приведем скриншоты выполнения программы:

E:\PythonYandex\work\venv\Scripts\python.exe E:/P

input: 1101

$K1+1+1+1 \rightarrow K1=1.$

$K2+1+0+1 \rightarrow K2=0.$

$K3+1+0+1 \rightarrow K3=0.$

позиция символов -----кодировое слово-----		

_____1_____	_____K_1_____	_____1_____
_____2_____	_____K_2_____	_____0_____
_____3_____	_____1_____	_____1_____
_____4_____	_____K_3_____	_____0_____
_____5_____	_____1_____	_____1_____
_____6_____	_____0_____	_____0_____
_____7_____	_____1_____	_____1_____

CODE:1010101

E:\PythonYandex\work\venv\Scripts\python.exe E:/P

input: 0101

$K1+0+1+1 \rightarrow K1=0.$

$K2+0+0+1 \rightarrow K2=1.$

$K3+1+0+1 \rightarrow K3=0.$

позиция символов -----кодировое слово-----		

_____1_____	_____K_1_____	_____0_____
_____2_____	_____K_2_____	_____1_____
_____3_____	_____0_____	_____0_____
_____4_____	_____K_3_____	_____0_____
_____5_____	_____1_____	_____1_____
_____6_____	_____0_____	_____0_____
_____7_____	_____1_____	_____1_____

CODE:0100101