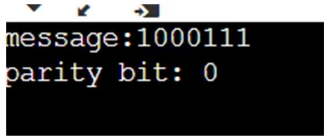
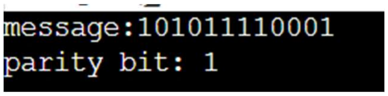
Parity bit:

```
n=input("message:")

c=n.count('1')

print("parity bit:",c%2)
```

Output:



```
message:1000111
parity bit: 0
```

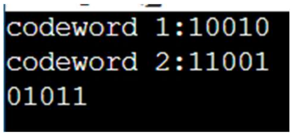


```
message:101011110001
parity bit: 1
```

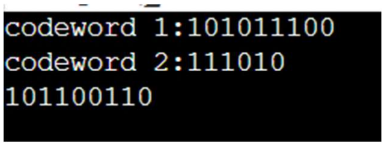
XOR:

```
q1=input("codeword 1:")

q2=input("codeword 2:")

n1=int(q1,2)

n2=int(q2,2)

x=n1^n2

x=bin(x).replace("0b","")

x=(max(len(q1),len(q2))-len(x))*"0"+x

print(x)
```

Output:



```
codeword 1:10010
codeword 2:11001
01011
```

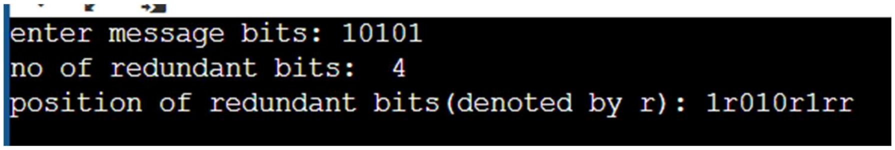


```
codeword 1:101011100
codeword 2:111010
101100110
```

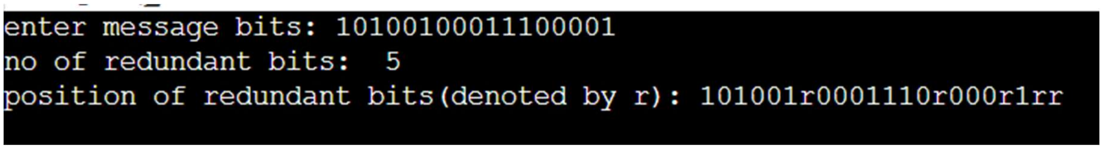
Hamming code:

```
n=input("enter message bits: ")

r=0

q=list(n)

q=list(reversed(q))

while(1):

   if(2**r>=(len(n)+1+r)):

      break

   else:

      r+=1

print("no of redundant bits: ",r)

for i in range(r):

   q.insert((2**i)-1,"r")

print("position of redundant bits(denoted by r): "+"".join(list(reversed(q))))
```

Output:



```
enter message bits: 10101
no of redundant bits:  4
position of redundant bits(denoted by r): 1r010r1rr
```



```
enter message bits: 10100100011100001
no of redundant bits:  5
position of redundant bits(denoted by r): 101001r0001110r000r1rr
```