



LABWORK 3

Write MATLAB functions with the following descriptions:

Decode2D: The matrix y is a 7×7 binary message matrix. By adding parity bits to 8th row and column, the 8×8 matrix x is obtained.

The function `Decode2D` takes the input x . If there are no errors, it gives y and prints a message. If there is one error, it corrects the error and gives y . It also prints the coordinates of the error. If there are two or more errors, it prints "REQUEST RETRANSMISSION".

HammingEncode: Given a message in the form of an $n \times 2$ binary matrix, the function `HammingEncode` outputs an $n \times 5$ binary matrix according to the following translation rules:

00 → 00000
01 → 01011
10 → 10101
11 → 11110

Decode2D

```
function y = Decode2D(x)

if (size(x,1) ~= 8 || size(x,2) ~= 8)
    error('The input x must be an 8 x 8 matrix');
end

RowErr = 0;
for i = 1:8
    sum = 0;
    for j = 1:8
        if (x(i,j) ~= 0 && x(i,j) ~= 1)
            error('The input x must be a binary matrix');
        end
        sum = sum + x(i,j);
    end
    if(mod(sum,2) ~= 0)
        RowErr = RowErr + 1;
        row = i;
    end
end

ColErr = 0;
for i = 1:8
    sum = 0;
    for j = 1:8
        sum = sum + x(j,i);
    end
    if(mod(sum,2) ~= 0)
        ColErr = ColErr + 1;
        col = i;
    end
end

if (RowErr == 0 && ColErr == 0)
    y = x;    y(8,:)=[]; y(:,8)=[];
    fprintf('There are no errors\n\n');
elseif (RowErr == 1 && ColErr == 1)
    x(row,col) = 1 - x(row,col);
    y = x;    y(8,:)=[]; y(:,8)=[];
    fprintf('There is one error at (%d,%d). It is corrected\n\n',row,col);
else
    y = zeros(7);
    fprintf('There are two or more errors. REQUEST RETRANSMSSION.\n\n');
end

end
```

HammingEncode

```
function B = HammingEncode( A )

if (size(A,2) ~= 2)
    error('The input A must be an n x 2 matrix');
end

n = size(A,1);
B=zeros(n,5);
Codewords = [ 0 0 0 0 0; 0 1 0 1 1;...
             1 0 1 0 1; 1 1 1 1 0];

for i = 1:n
    k = 1 + A(i,2) + A(i,1) * 2;
    B(i,:)=Codewords(k,:);
end

end
```