



LABWORK 2

Write MATLAB functions with the following descriptions:

NoiseFlip: User gives a binary message (a vector) of arbitrary length and a probability p . The function modifies each bit (transforms ones into zeros and vice versa) with probability p .

AddFlag: The user gives a binary message (a vector) whose length is a multiple of 8. The function adds (inserts) the flag 1 1 1 1 between 8 bit parts. (Also to the beginning and end)

NoiseDel: User gives a binary message (a vector) of arbitrary length. The function chooses one bit randomly and deletes it. Output length = input length -1 .

Parity7: The user gives a binary message (a vector) whose length is a multiple of 7. The function adds a parity bit for every group of 7. The output will be a $k \times 8$ matrix.

NoiseFlip

```
function y = NoiseFlip(x,p)

%Usage Example: x = [1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1]
%NoiseFlip(x,0.25)

n = size(x,2);
for i = 1:n
    if rand() < p
        y(i) = 1 - x(i);
    else
        y(i) = x(i);
    end
end

end
```

AddFlag

```
function y = AddFlag(x)
%Usage Example: x = [1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 1]
%AddFlag(x)

n = size(x,2)/8;
y = [1 1 1 1];
for i = 1:n
    for j = 1:8
        y = [y x((i-1)*8+j)];
    end
    y = [y 1 1 1 1];
end

end
```

NoiseDel

```
function y = NoiseDel(x)

%Usage Example: x = [0 0 0 1 0 1 0 1 0 0 0 1 1 1 0 0]
%NoiseDel(x)

n = size(x,2);
j = ceil(n * rand());
for i = 1:j-1
    y(i) = x(i);
end

for i = j+1:n
    y(i-1) = x(i);
end

%A better idea is: y=[x(1:j-1) x(j+1:n)]

end
```

Parity7

```
function y = Parity7(x)

%Usage Example: x = [0 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1]
%Parity7(x)

n = size(x,2)/7;
y=[ ];
for i = 1:n
    sum = 0;
    for j = 1:7
        v(j) = x((i-1) * 7 + j);
        sum = sum + v(j);
    end
    v(8) = mod(sum,2);
    y = [y;v];
end

end
```