```
%%Limpel_Ziv Simulation :
```

Encoding :

```
N=input('Enter the number of Bits : ');
input_data=randi([0 1],1,N); %Genrate N bits randomly

%enter K " check if its enough
k=input('Enter the number of Locations : ');
dictionary=(0:k);  %create a dctionary
Truth_table=[de2bi(dictionary,'left-msb')]; %Create a truth table

sequence=cell(1,k+1);
s='';


code_letter=cell(1,k+1);
base=cell(1,k+1);
check='';

fprintf('\nInput data : \n ');
fprintf('%d',input_data);

fprintf('\n sequence\t\t\t base \t\t\tcode letter \n');
%for loop to take letters of the binary text
for i=1:length(input_data)
         L=num2str(input_data(i));
         if or((any(strcmp(sequence,'1')==1)),
(any(strcmp(sequence,'0')==1)))


          check=strcat(check,L);


                 if(any(strcmp(sequence,check)==1))

                    continue

                    else
                     sequence{i}=check;
                 end

             check='';
             else

             sequence{i}=L;


          end
 %final output for Sequence
emptie=find(cellfun(@isempty,sequence));
sequence(emptie)=[];
```

```
%Base and Location and Code :
for i=1:length(sequence)
    base{i}=sequence{i}(end);
       if or((any(strcmp('0',sequence{i})==1)),
 (any(strcmp('1',sequence{i})==1)))

          code_letter{i}=[num2str(Truth_table(1,:)) sequence{i}];

              else

                   s=sequence{i}(1:end-1);
                    for index=1:length(sequence)
                   m=find(strcmp(s,sequence));
                    end

                   code_letter{i}=strcat(num2str(Truth_table(m
+1,:)),sequence{i}(end));
      end

end

%Show output in command window :
      fprintf('%s\t\t\t\t%s\t\t\t%s\t
\n',sequence{i},base{i},code_letter{i})  ;

end

empties=find(cellfun(@isempty,code_letter));
code_letter(empties)=[];
```

```
Error using input
Cannot call INPUT from EVALC.

Error in untitled0 (line 3)
N=input('Enter the number of Bits : ');
```

Decoding :

```
DecodedData='';
DeSequence=cell(1,length(code_letter));
Test=cell(1,length(code_letter));

location=cell(1,length(code_letter));
fprintf('Refrence location\t\t\tDeSequence\t\t\tLocation\n');
for i=1:length(code_letter)
    DeSequence{i}=code_letter{i}(end);

location{i}=num2str(Truth_table(i+1,:));
Test{i}=code_letter{i}(1:end-1);
    if(strcmp(Test{i},num2str(Truth_table(1,:)))==1)
    DecodedData=strcat(DecodedData,DeSequence{i});
    else
    m=find(strcmp(location,Test{i}));
    DeSequence{i}=strcat(DeSequence{m},DeSequence{i});
    DecodedData=strcat(DecodedData,DeSequence{i});

    end
fprintf('%s\t\t\t%s\t\t\t%s\n',Test{i},DeSequence{i},location{i});
end


disp('Decoding Data :');
disp(DecodedData);
```

Published with MATLAB® R2020a