**LZW CODES**

**Compression:**

clc;

clear all;

%counters

x=1;

dictCounter=256;

d=1;

%file input

fileID = fopen('../data.txt','r');

a = fscanf(fileID,'%s');

fclose(fileID);

fileLength = length(a)\*8;

%output array

output=[];

%init

p=a(x);

c=a(x+1);

string =[p,c];

output=[output,dec2bin(double(p),12)];

dict{d,1} = dictCounter;

dict{d,2} = string;

dictCounter= dictCounter+1;

x=x+1;

d=d+1;

p=c;

while x<length(a)

c=a(x+1);

disp(['current: ',p,' Next: ',c]);

string=[p,c];

[bool,index] = inDictionary(dict,string);

if bool

disp(['MatchFound for: ',string, 'at: ', num2str(index)]);

p=string;

else

disp(['adding: ',p]);

if length(p)==1

output=[output,dec2bin(double(p),12)];

else

[bool,index] = inDictionary(dict,p);

output=[output,dec2bin(double(index),12)];

end

if dictCounter ~= 4096

dict{d,1} = dictCounter;

dict{d,2} = string;

dictCounter = dictCounter+1;

d=d+1;

end

p=c;

end

x=x+1;

end

if length(p)==1

disp(['p: ', p]);

output=[output,dec2bin(double(p),12)];

else

[bool,index] = inDictionary(dict,p);

output=[output,dec2bin(double(index),12)];

end

for i=1:12:length(output)

ce{floor(i/12)+1} = ['Decimal: ',num2str(bin2dec(output(i:i+11)))];

disp(ce{floor(i/12)+1});

disp(['Binary: ',output(i:i+11),' Decimal: ',num2str(bin2dec(output(i:i+11)))]);

end

fileID = fopen('output.txt','w');

for i=1:12:length(output)

fprintf(fileID,'%s',output(i:i+11));

fprintf(fileID,'\n');

end

fclose(fileID);

compressedLength = length(output);

Compression = 1-(compressedLength/fileLength);

disp(['fileLength: ',num2str(fileLength)]);

disp(['CompressedLength: ',num2str(compressedLength)]);

disp(['Compression: ',num2str(Compression),'%']);

**Output:**

fileLength: 22968

CompressedLength: 14832

Compression: 35.4232%

**Decompression:**

clc ;

clear all;

fileID = fopen('output.txt','r');

a=bin2dec(num2str(fscanf(fileID,'%12lu')));

output=[];

x=1;

d=1;

dictCounter=256;

p=char(a(x));

c=char(a(x+1));

string=[p,c];

dict{d,1} = dictCounter;

dict{d,2} = string;

output=[output,p];

p=c;

d=d+1;

dictCounter=dictCounter+1;

x=x+1;

while x<length(a)

if a(x+1)<255

c= char(a(x+1));

else

c=dict{a(x+1)-255,2};

end

disp(num2str(dictCounter));

disp(c);

string = [p,c(1)];

[bool,index] = inDictionary(dict,string);

if bool

p=string;

else

output = [output,p];

dict{d,1} = dictCounter;

dict{d,2} = string;

dictCounter = dictCounter+1;

d=d+1;

p=c;

end

x=x+1;

end

output=[output,p];

disp(output);

**OUTPUT:**

Lorem\_ipsum\_dolor\_sit\_amet,\_consectetur\_adipiscing\_elit.\_Nunc\_at\_pellentesque\_ex.\_Maecenas\_et\_magna\_varius,\_fringilla\_massa\_vel,\_pharetra\_tellus.\_Ut\_id\_lectus\_in\_turpis\_mollis\_ullamcorper\_nec\_non\_felis.\_Nam\_rutrum\_fringilla\_neque,\_vel\_eleifend\_tortor\_gravida\_sed.\_Phasellus\_id\_lacus\_ac\_purus\_placerat\_malesuada\_ut\_et\_elit.\_Phasellus\_dignissim\_ante\_vitae\_tortor\_efficitur\_vestibulum.\_Donec\_interdum,\_augue\_quis\_malesuada\_faucibus,\_augue\_nisi\_congue\_elit,\_id\_egestas\_ligula\_justo\_sit\_amet\_arcu.\_Nullam\_in\_malesuada\_nisl,\_vel\_dapibus\_felis.\_Suspendisse\_sed\_dapibus\_lacus,\_vitae\_molestie\_nunc.\_Integer\_arcu\_massa,\_tristique\_id\_consequat\_at,\_rhoncus\_eu\_urna.\_In\_hac\_habitasse\_platea\_dictumst.\_Donec\_est\_risus,\_scelerisque\_in\_odio\_nec,\_luctus\_euismod\_nisi.\_Donec\_laoreet\_vitae\_lacus\_vel\_faucibus.\_Nullam\_pretium\_mauris\_diam,\_non\_hendrerit\_eros\_elementum\_quis.Donec\_a\_imperdiet\_magna.\_Suspendisse\_euismod,\_tellus\_ac\_luctus\_consequat,\_nulla\_sem\_dapibus\_nisl,\_quis\_congue\_sapien\_erat\_a\_nibh.\_Pellentesque\_habitant\_morbi\_tristique\_senectus\_et\_netus\_et\_malesuada\_fames\_ac\_turpis\_egestas.\_Ut\_nec\_venenatis\_nisi.\_Nulla\_sit\_amet\_lectus\_molestie,\_malesuada\_magna\_quis,\_tincidunt\_nulla.\_Nam\_ornare\_felis\_imperdiet\_tristique\_ultricies.\_Duis\_congue\_placerat\_ligula.Fusce\_at\_placerat\_lorem.\_Mauris\_vestibulum\_mauris\_nec\_quam\_eleifend,\_a\_tristique\_sapien\_condimentum.\_Quisque\_nisi\_purus,\_cursus\_sed\_mauris\_id,\_gravida\_porta\_nisi.\_Proin\_sit\_amet\_dapibus\_lorem.\_Nulla\_vitae\_convallis\_lacus.\_Nam\_vel\_luctus\_mauris.\_Cras\_rutrum\_enim\_est,\_eget\_pretium\_eros\_porttitor\_pellentesque.\_Phasellus\_vel\_tempor\_eros.Fusce\_malesuada,\_arcu\_eu\_molestie\_condimentum,\_lorem\_enim\_laoreet\_urna,\_et\_efficitur\_purus\_lacus\_id\_est.\_Nunc\_aliquam\_elementum\_pharetra.\_Nunc\_odio\_justo,\_bibendum\_in\_tempor\_eget,\_aliquam\_sed\_nisl.\_Aenean\_vel\_eros\_eget\_magna\_tincidunt\_tempus.\_Pellentesque\_neque\_elit,\_dictum\_vel\_volutpat\_id,\_lacinia\_condimentum\_leo.\_Phasellus\_ultricies\_molestie\_erat\_ut\_aliquam.\_Cras\_euismod\_at\_ipsum\_eu\_tristique.\_Cras\_suscipit\_neque\_ac\_nunc\_pulvinar,\_ac\_ornare\_ligula\_pharetra.\_Aenean\_tempus\_auctor\_lorem\_convallis\_tempus.\_Aliquam\_erat\_volutpat.\_Orci\_varius\_natoque\_penatibus\_et\_magnis\_dis\_parturient\_montes,\_nascetur\_ridiculus\_mus.\_In\_volutpat\_nulla\_ac\_eros\_accumsan\_ullamcorper.\_Maecenas\_ultricies\_massa\_nec\_tortor\_tristique\_hendrerit.\_Praesent\_vehicula\_metus\_mauris,\_ut\_rutrum\_elit\_fringilla\_quis.Morbi\_consectetur\_sit\_amet\_justo\_in\_cursus.\_Orci\_varius\_natoque\_penatibus\_et\_magnis\_dis\_parturient\_montes,\_nascetur\_ridiculus\_mus.\_Pellentesque\_luctus\_nunc\_finibus,\_facilisis\_eros\_et,\_volutpat\_lectus.\_Nam\_consequat,\_ante\_quis\_vulputate\_consectetur,\_nisl\_ante\_pretium\_mi,\_in\_malesuada\_sapien\_libero\_id\_quam.\_Donec\_dictum\_ligula\_nec\_elit\_luctus\_semper.\_Aenean\_consequat\_felis\_nec\_ante\_consequat\_porta.\_Praesent\_odio\_purus,\_porta\_vitae\_suscipit\_dapibus,\_tincidunt\_ac\_erat.\_Sed\_faucibus\_sollicitudin\_cursus.