Name: Sohel Momin

Roll No:1513126/a4

CSS EXPERIMENT 3

**Program:**

%% AFFINE CIPHER

% This program takes a user message string

% Asks user for VALID multiplicative and additive cipher keys [k1,k2]

% Encrypts the message according to the key C=(P\*k1+k2)mod26

% Decrypts and displays the message D=((C-k2)\*k1\_inv))mod26

clc;clear variables;close all;

%% Getting Valid keys

[k1,k2] = deal(-1,-1);

while (k1==-1) || (k2==-1)

if k1==-1

temp = input('Enter multiplicative cipher key in Z-26\* -> ');

[mi,gcd] = mulinv(temp,26);

if gcd == 1

[k1,k1\_inv] = deal(temp,mi);

end

elseif k2==-1

temp = input('Enter additive cipher key in Z-26 -> ');

if (temp>0) && (temp<26)

k2=temp;

end

end

end

%% ENCRYPTION

msg = double(input('Enter message to be encrypted -> ','s'));

msg = msg - 65\*ismember(msg,65:90) - 97\*ismember(msg,97:122); % so that upper/lower case doesnt matter

enc\_num = mod(msg\*k1 + k2, 26);

enc\_msg = char(enc\_num + 97);

fprintf('The encrypted message is -> "%s"\n',enc\_msg);

%% DECRYPTION

dec\_num = mod((double(enc\_msg)-97-k2)\*k1\_inv,26);

dec\_msg = char(dec\_num + 97);

fprintf('The decrypted message is -> "%s"',dec\_msg);

%% TRANSPOSITION ENCRYPTION

% This program takes a user message string

% Asks user to define the null character and the key

% Encrypts the message according to the key

% Decrypts and displays the message

clc;clear variables;close all;

%% Get message, null character and key

msg = input('Enter message string -> ','s');

null = input('Enter null character -> ','s');

enc\_key = input('Enter encryption key in square brackets -> ');

keysize = length(enc\_key);

rows = ceil(length(msg)/keysize);

%% Adding extra null chr

extra = (keysize\*rows)-length(msg);

for i = 1:extra

msg = strcat(msg,null);

end

msg\_matrix = reshape(msg,[keysize,rows])';

%% Encoding

for i = 1:keysize

enc\_matrix(:,i) = msg\_matrix(:,enc\_key(i));

end

enc\_msg = reshape(enc\_matrix,[1,rows\*keysize]);

fprintf('The encrypted message is -> "%s"\n',enc\_msg);

%% Decoding without Decryption Key

for i = 1:keysize

dec\_matrix(:,enc\_key(i)) = enc\_matrix(:,i);

end

dec\_msg = reshape(dec\_matrix',[1,rows\*keysize]);

fprintf('The decrypted message is -> "%s"',dec\_msg);

%% Decryption Key generation

% for i=1:keysize

% dec\_key(i)=find(enc\_key==i);

% end

% fprintf('The decryption key is');

% disp(dec\_key);

**OUTPUT (AFFINE CIPHER)**

Enter multiplicative cipher key in Z-26\* -> 7

Enter additive cipher key in Z-26 -> 15

Enter message to be encrypted -> WindsofWinter

The encrypted message is -> "ntckljyntcsre"

The decrypted message is -> "windsofwinter"

**OUTPUT (TRANSPOSITION CIPHER)**

Enter message string -> red alert tomorrow

Enter null character -> #

Enter encryption key in square brackets -> [3 2 1 5 4]

The encrypted message is -> "drmweeoorltra r# to#"

The decrypted message is -> "red alert tomorrow##"