**EXPERIMENT NO.1**

**CEASER CIPHER**

**SOURCE CODE:**

#include <stdio.h>

#include <string.h>

#include<conio.h>

#include <ctype.h>

void main()

{

char plain[10], cipher[10];

int key,i,length;

int result;

//clrscr();

printf("\n Enter the plain text:");

scanf("%s", plain);

printf("\n Enter the key value:");

scanf("%d", &key);

printf("\n \n \t PLAIN TEXt: %s",plain);

printf("\n \n \t ENCRYPTED TEXT: ");

for(i = 0, length = strlen(plain); i < length; i++)

{

cipher[i]=plain[i] + key;

if (isupper(plain[i]) && (cipher[i] > 'Z'))

cipher[i] = cipher[i] - 26;

if (islower(plain[i]) && (cipher[i] > 'z'))

cipher[i] = cipher[i] - 26;

printf("%c", cipher[i]);

}

printf("\n \n \t AFTER DECRYPTION : ");

for(i=0;i<length;i++)

{

plain[i]=cipher[i]-key;

if(isupper(cipher[i])&&(plain[i]<'A'))

plain[i]=plain[i]+26;

if(islower(cipher[i])&&(plain[i]<'a'))

plain[i]=plain[i]+26;

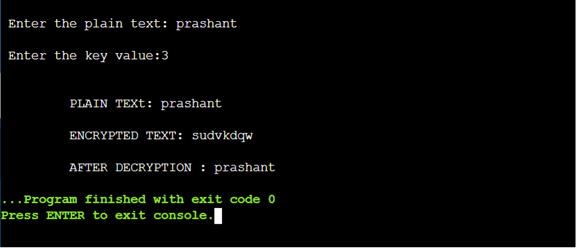
printf("%c",plain[i]);

}

getch();

}

**OUTPUT:**



**EXPERIMENT NO.2**

**PLAYFAIR CIPHER**

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<ctype.h>

#define MX 5

void playfair(char ch1,char ch2, char key[MX][MX])

{

int i,j,w,x,y,z;

FILE \*out;

if((out=fopen("cipher.txt","a+"))==NULL)

{

printf("File Currupted.");

}

for(i=0;i<MX;i++)

{

for(j=0;j<MX;j++)

{

if(ch1==key[i][j])

{

w=i;

x=j;

}

else if(ch2==key[i][j])

{

y=i;

z=j;

}}}

//printf("%d%d %d%d",w,x,y,z);

if(w==y)

{

x=(x+1)%5;z=(z+1)%5;

printf("%c%c",key[w][x],key[y][z]);

fprintf(out, "%c%c",key[w][x],key[y][z]);

}

else if(x==z)

{

w=(w+1)%5;y=(y+1)%5;

printf("%c%c",key[w][x],key[y][z]);

fprintf(out, "%c%c",key[w][x],key[y][z]);

}

else

{

printf("%c%c",key[w][z],key[y][x]);

fprintf(out, "%c%c",key[w][z],key[y][x]);

}

fclose(out);

}

void main()

{

int i,j,k=0,l,m=0,n;

char key[MX][MX],keyminus[25],keystr[10],str[25]={0};

char

alpa[26]={'A','B','C','D','E','F','G','H','I','J','K','L'

,'M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'}

;

clrscr();

printf("\nEnter key:");

gets(keystr);

printf("\nEnter the plain text:");

gets(str);

n=strlen(keystr);

//convert the characters to uppertext

for (i=0; i<n; i++)

{

if(keystr[i]=='j')keystr[i]='i';

else if(keystr[i]=='J')keystr[i]='I';

keystr[i] = toupper(keystr[i]);

}

//convert all the characters of plaintext to uppertext

for (i=0; i<strlen(str); i++)

{

if(str[i]=='j')str[i]='i';

else if(str[i]=='J')str[i]='I';

str[i] = toupper(str[i]);

}

j=0;

for(i=0;i<26;i++)

{

for(k=0;k<n;k++)

{

if(keystr[k]==alpa[i])

break;

else if(alpa[i]=='J')

break;

}

if(k==n)

{

keyminus[j]=alpa[i];j++;

}

}

//construct key keymatrix

k=0;

for(i=0;i<MX;i++)

{

for(j=0;j<MX;j++)

{

if(k<n)

{

key[i][j]=keystr[k];

k++;}

else

{

key[i][j]=keyminus[m];m++;

}

printf("%c ",key[i][j]);

}

printf("\n");

}

printf("\n\nEntered text :%s\nCipher Text :",str);

for(i=0;i<strlen(str);i++)

{

if(str[i]=='J')str[i]='I';

if(str[i+1]=='\0')

playfair(str[i],'X',key);

else

{

if(str[i+1]=='J')str[i+1]='I';

if(str[i]==str[i+1])

playfair(str[i],'X',key);

else

{

playfair(str[i],str[i+1],key);i++;

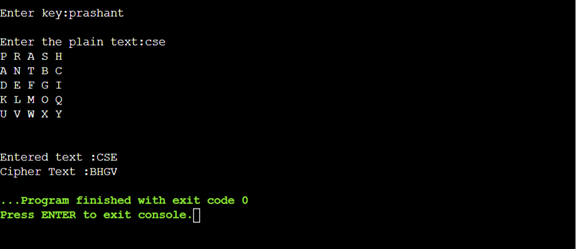
}}

}

getch();

}

**OUTPUT:**





**EXPERIMENT NO.3**

**HILL CIPHER**

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

int main(){

unsigned int a[3][3]={{6,24,1},{13,16,10},{20,17,15}};

unsigned int b[3][3]={{8,5,10},{21,8,21},{21,12,8}};

int i,j, t=0;

unsigned int c[20],d[20];

char msg[20];

//clrscr();

printf("Enter plain text: ");

scanf("%s",msg);

for(i=0;i<strlen(msg);i++)

{ c[i]=msg[i]-65;

printf("%d ",c[i]);

}

for(i=0;i<3;i++)

{ t=0;

for(j=0;j<3;j++)

{

t=t+(a[i][j]\*c[j]);

}

d[i]=t%26;

}

printf("\nEncrypted Cipher Text :");

for(i=0;i<3;i++)

printf(" %c",d[i]+65);

for(i=0;i<3;i++)

{

t=0;

for(j=0;j<3;j++)

{

t=t+(b[i][j]\*d[j]);

}

c[i]=t%26;

}

printf("\nDecrypted Cipher Text :");

for(i=0;i<3;i++)

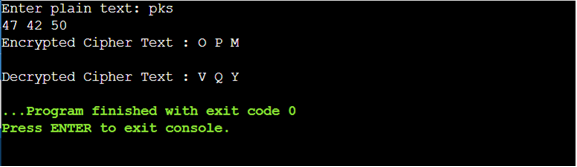
printf(" %c",c[i]+65);

getch();

return 0;

}

**OUTPUT:**



**EXPERIMENT NO.4**

**VIGENERE CIPHER**

**SOURCE CODE:**

#include <stdio.h>

#include<conio.h>

#include <ctype.h>

#include <string.h>

void encipher();

void decipher();

void main()

{

int choice;

clrscr();

while(1)

{

printf("\n1. Encrypt Text");

printf("\t2. Decrypt Text");

printf("\t3. Exit");

printf("\n\nEnter Your Choice : ");

scanf("%d",&choice);

if(choice == 3)

exit(0);

else if(choice == 1)

encipher();

else if(choice == 2)

decipher();

else

printf("Please Enter Valid Option.");

}

}

void encipher()

{

unsigned int i,j;

char input[50],key[10];

printf("\n\nEnter Plain Text: ");

scanf("%s",input);

printf("\nEnter Key Value: ");

scanf("%s",key);

printf("\nResultant Cipher Text: ");

for(i=0,j=0;i<strlen(input);i++,j++)

{

if(j>=strlen(key))

{ j=0;

}

printf("%c",65+(((toupper(input[i])-65)+(toupper(key[j])-

65))%26));

}}

void decipher()

{

unsigned int i,j;

char input[50],key[10];

int value;

printf("\n\nEnter Cipher Text: ");

scanf("%s",input);

printf("\n\nEnter the key value: ");

scanf("%s",key);

for(i=0,j=0;i<strlen(input);i++,j++)

{

if(j>=strlen(key))

{ j=0; }

value = (toupper(input[i])-64)-(toupper(key[j])-64);

if( value < 0)

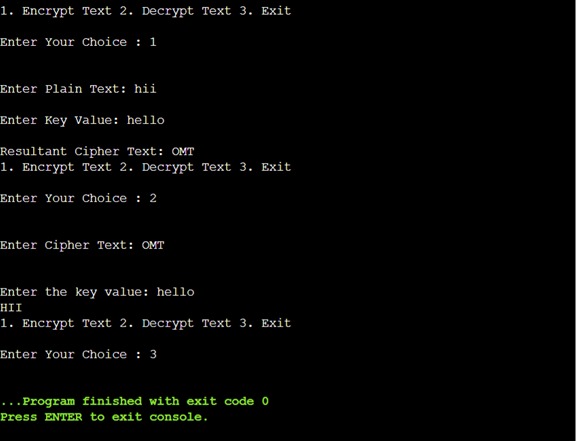
{ value = value \* -1;

}

printf("%c",65 + (value % 26));

}}

**OUTPUT:**



**EXPERIMENT NO.5**

**RAIL FENCE TECHNIQUE**

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

int i,j,k,l;

char a[20],c[20],d[20];

clrscr();

printf("\n\t\t RAIL FENCE TECHNIQUE");

printf("\n\nEnter the input string : ");

gets(a);

l=strlen(a);

/\*Ciphering\*/

for(i=0,j=0;i<l;i++)

{

if(i%2==0)

c[j++]=a[i];

}

for(i=0;i<l;i++)

{

if(i%2==1)

c[j++]=a[i];

}

c[j]='\0';

printf("\nCipher text after applying rail fence :");

printf("\n%s",c);

/\*Deciphering\*/

if(l%2==0)

k=l/2;

else

k=(l/2)+1;

for(i=0,j=0;i<k;i++)

{

d[j]=c[i];

j=j+2;

}

for(i=k,j=1;i<l;i++)

{

d[j]=c[i];

j=j+2;

}

d[l]='\0';

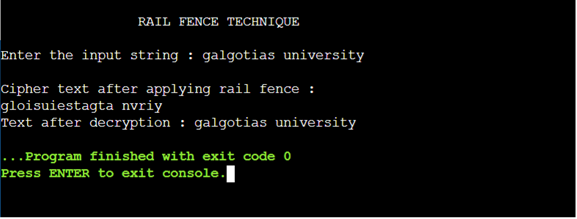
printf("\nText after decryption : ");

printf("%s",d);

getch();

}

**OUTPUT:**



**EXPERIMENT NO.6**

**IMPLEMENTATION OF DES**

**AIM:** To write a java program to implement Data Encryption Standard (DES) using java.

**SOURCE CODE:**

import javax.swing.\*;

import java.security.SecureRandom;

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import java.util.Random ;

class DES {

byte[] skey = new byte[1000];

String skeyString;

static byte[] raw;

String inputMessage,encryptedData,decryptedMessage;

public DES()

{

try

{

generateSymmetricKey();

inputMessage=JOptionPane.showInputDialog(null,"Enter message to encrypt");

byte[] ibyte = inputMessage.getBytes();

byte[] ebyte=encrypt(raw, ibyte);

String encryptedData = new String(ebyte);

System.out.println("Encrypted message "+encryptedData);

JOptionPane.showMessageDialog(null,"Encrypted Data "+"\n"+encryptedData);

byte[] dbyte= decrypt(raw,ebyte);

String decryptedMessage = new String(dbyte);

System.out.println("Decrypted message "+decryptedMessage);

JOptionPane.showMessageDialog(null,"Decrypted Data "+"\n"+decryptedMessage);

}

catch(Exception e)

{

System.out.println(e);

}

}

void generateSymmetricKey()

{

try

{

Random r = new Random();

int num = r.nextInt(10000);

String knum = String.valueOf(num);

byte[] knumb = knum.getBytes();

skey=getRawKey(knumb);

skeyString = new String(skey);

System.out.println("DES Symmetric key = "+skeyString);

}

catch(Exception e)

{

System.out.println(e);

}

}

private static byte[] getRawKey(byte[] seed) throws Exception

{

KeyGenerator kgen = KeyGenerator.getInstance("DES");

SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");

sr.setSeed(seed);

kgen.init(56, sr);

SecretKey skey = kgen.generateKey();

raw = skey.getEncoded();

return raw;

}

private static byte[] encrypt(byte[] raw, byte[] clear) throws

Exception {

SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");

Cipher cipher = Cipher.getInstance("DES");

cipher.init(Cipher.ENCRYPT\_MODE, skeySpec);

byte[] encrypted = cipher.doFinal(clear);

return encrypted;

}

private static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception

{

SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");

Cipher cipher = Cipher.getInstance("DES");

cipher.init(Cipher.DECRYPT\_MODE, skeySpec);

byte[] decrypted = cipher.doFinal(encrypted);

return decrypted;

}

public static void main(String args[])

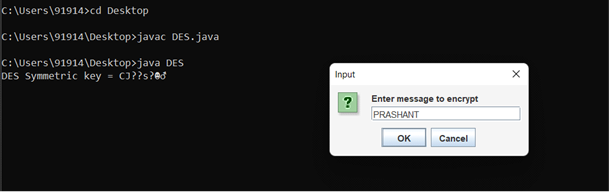
{

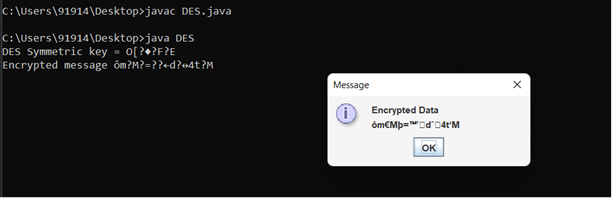
DES des = new DES();

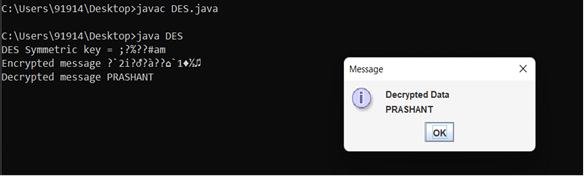
}

}

**OUTPUT:**







**EXPERIMENT NO.7**

**IMPLEMENTATION OF RSA**

**AIM:** To write a C program to implement the RSA encryption algorithm.

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<math.h>

#include<string.h>

long int

p,q,n,t,flag,e[100],d[100],temp[100],j,m[100],en[100],i;

char msg[100];

int prime(long int);

void ce();

long int cd(long int);

void encrypt();

void decrypt();

void main()

{

//clrscr();

printf("\nENTER FIRST PRIME NUMBER\n");

scanf("%d",&p);

flag=prime(p);

if(flag==0)

{

printf("\nWRONG INPUT\n");

getch();

}

printf("\nENTER ANOTHER PRIME NUMBER\n");

scanf("%d",&q);

flag=prime(q);

if(flag==0||p==q)

{

printf("\nWRONG INPUT\n");

getch();

}

printf("\nENTER MESSAGE\n");

fflush(stdin);

scanf("%s",msg);

for(i=0;msg[i]!=NULL;i++)

m[i]=msg[i];

n=p\*q;

t=(p-1)\*(q-1);

ce();

printf("\nPOSSIBLE VALUES OF e AND d ARE\n");

for(i=0;i<j-1;i++)

printf("\n%ld\t%ld",e[i],d[i]);

encrypt();

decrypt();

getch();

}

int prime(long int pr)

{

int i;

j=sqrt(pr);

for(i=2;i<=j;i++)

{

if(pr%i==0)

return 0;

}

return 1;

}

void ce()

{

int k;

k=0;

for(i=2;i<t;i++)

{

if(t%i==0)

continue;

flag=prime(i);

if(flag==1&&i!=p&&i!=q)

{

e[k]=i;

flag=cd(e[k]);

if(flag>0)

{

d[k]=flag;

k++;

}

if(k==99)

break;

} } }

long int cd(long int x)

{

long int k=1;

while(1)

{

k=k+t;

if(k%x==0)

return(k/x);

} }

void encrypt() {

long int pt,ct,key=e[0],k,len;

i=0;

len=strlen(msg);

while(i!=len) {

pt=m[i];

pt=pt-96;

k=1;

for(j=0;j<key;j++)

{ k=k\*pt;

k=k%n;

}

temp[i]=k;

ct=k+96;

en[i]=ct;

i++;

}

en[i]=-1;

printf("\nTHE ENCRYPTED MESSAGE IS\n");

for(i=0;en[i]!=-1;i++)

printf("%c",en[i]);

}

void decrypt()

{

long int pt,ct,key=d[0],k;

i=0;

while(en[i]!=-1)

{

ct=temp[i];

k=1;

for(j=0;j<key;j++)

{

k=k\*ct;

k=k%n;

}

pt=k+96;

m[i]=pt;

i++;

}

m[i]=-1;

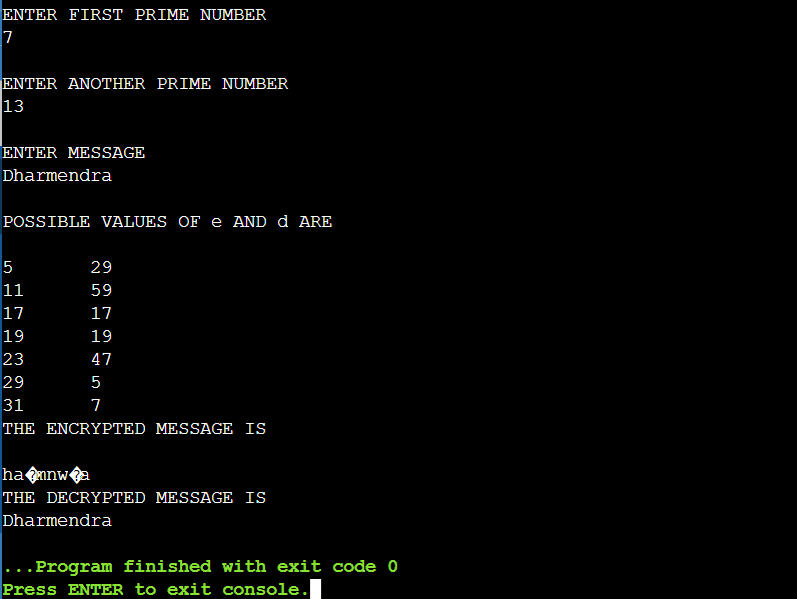
printf("\nTHE DECRYPTED MESSAGE IS\n");

for(i=0;m[i]!=-1;i++)

printf("%c",m[i]);

}

**OUTPUT:**



**EXPERIMENT NO.9**

**IMPLEMENTATION OF DIFFIE HELLMAN KEY EXCHANGE**  **ALGORITHM**

**AIM:** To implement the Diffie-Hellman Key Exchange algorithm using C language.

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

long long int power(int a, int b, int mod)

{

long long int t;

if(b==1)

return a;

t=power(a,b/2,mod);

if(b%2==0)

return (t\*t)%mod;

else

return (((t\*t)%mod)\*a)%mod;

}

long int calculateKey(int a, int x, int n)

{

return power(a,x,n);

}

void main()

{

int n,g,x,a,y,b;

//clrscr();

printf("Enter the value of n and g : ");

scanf("%d%d",&n,&g);

printf("Enter the value of x for the first person : ");

scanf("%d",&x);

a=power(g,x,n);

printf("Enter the value of y for the second person : ");

scanf("%d",&y);

b=power(g,y,n);

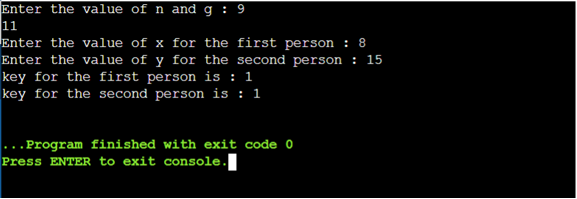
printf("key for the first person is : %lld\n",power(b,x,n));

printf("key for the second person is : %lld\n",power(a,y,n));

getch();

}

**OUTPUT:**



**EXPERIMENT NO.10**

**IMPLEMENTATION OF MD5**

**AIM:** To write a C program to implement the MD5 hashing technique.

**SOURCE CODE:**

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <math.h>

#include<conio.h>

typedef union uwb

{

unsigned w;

unsigned char b[4];

} MD5union;

typedef unsigned DigestArray[4];

unsigned func0( unsigned abcd[] ){

return ( abcd[1] & abcd[2]) | (~abcd[1] & abcd[3]);}

unsigned func1( unsigned abcd[] ){

return ( abcd[3] & abcd[1]) | (~abcd[3] & abcd[2]);}

unsigned func2( unsigned abcd[] ){

return abcd[1] ^ abcd[2] ^ abcd[3];}

unsigned func3( unsigned abcd[] ){

return abcd[2] ^ (abcd[1] |~ abcd[3]);}

typedef unsigned (\*DgstFctn)(unsigned a[]);

unsigned \*calctable( unsigned \*k)

{

double s, pwr;

int i;

pwr = pow( 2, 32);

for (i=0; i<64; i++)

{

s = fabs(sin(1+i));

k[i] = (unsigned)( s \* pwr );

}

return k;

}

unsigned rol( unsigned r, short N )

{

unsigned mask1 = (1<<N) -1;

return ((r>>(32-N)) & mask1) | ((r<<N) & ~mask1);

}

unsigned \*md5( const char \*msg, int mlen)

{

static DigestArray h0 = { 0x67452301, 0xEFCDAB89,

0x98BADCFE, 0x10325476 };

static DgstFctn ff[] = { &func0, &func1, &func2, &func3};

static short M[] = { 1, 5, 3, 7 };

static short O[] = { 0, 1, 5, 0 };

static short rot0[] = { 7,12,17,22};

static short rot1[] = { 5, 9,14,20};

static short rot2[] = { 4,11,16,23};

static short rot3[] = { 6,10,15,21};

static short \*rots[] = {rot0, rot1, rot2, rot3 };

static unsigned kspace[64];

static unsigned \*k;

static DigestArray h;

DigestArray abcd;

DgstFctn fctn;

short m, o, g;

unsigned f;

short \*rotn;

union

{

unsigned w[16];

char b[64];

}mm;

int os = 0;

int grp, grps, q, p;

unsigned char \*msg2;

if (k==NULL) k= calctable(kspace);

for (q=0; q<4; q++) h[q] = h0[q]; // initialize

{

grps = 1 + (mlen+8)/64;

msg2 = malloc( 64\*grps);

memcpy( msg2, msg, mlen);

msg2[mlen] = (unsigned char)0x80;

q = mlen + 1;

while (q < 64\*grps){ msg2[q] = 0; q++ ; }

{

MD5union u;

u.w = 8\*mlen;

q -= 8;

memcpy(msg2+q, &u.w, 4 );

}

}

for (grp=0; grp<grps; grp++)

{

memcpy( mm.b, msg2+os, 64);

for(q=0;q<4;q++) abcd[q] = h[q];

for (p = 0; p<4; p++)

{

fctn = ff[p];

rotn = rots[p];

m = M[p]; o= O[p];

for (q=0; q<16; q++)

{

g = (m\*q + o) % 16;

f = abcd[1] + rol( abcd[0]+ fctn(abcd)+k[q+16\*p]

+ mm.w[g], rotn[q%4]);

abcd[0] = abcd[3];

abcd[3] = abcd[2];

abcd[2] = abcd[1];

abcd[1] = f;

}}

for (p=0; p<4; p++)

h[p] += abcd[p];

os += 64;

}

return h;}

void main()

{

int j,k;

const char \*msg = "The quick brown fox jumps over the lazy dog";

unsigned \*d = md5(msg, strlen(msg));

MD5union u;

//clrscr();

printf("\t MD5 ENCRYPTION ALGORITHM IN C \n\n");

printf("Input String to be Encrypted using MD5 : \n\t%s",msg);

printf("\n\nThe MD5 code for input string is: \n");

printf("\t= 0x");

for (j=0;j<4; j++){

u.w = d[j];

for (k=0;k<4;k++) printf("%02x",u.b[k]);

}

printf("\n");

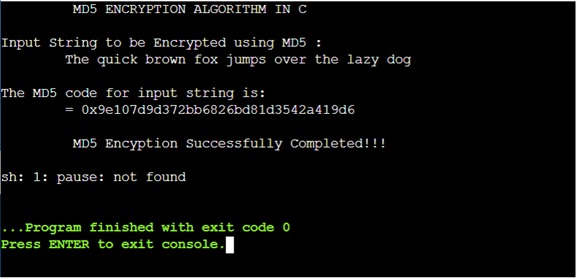
printf("\n\t MD5 Encyption Successfully Completed!!!\n\n");

getch();

system("pause");

getch();}

**OUTPUT:**



**IMPLEMENTATION OF SHA-I**

**AIM:** To implement the SHA-I hashing technique using JAVA.

**SOURCE CODE:**

import java.security.\*;

public class SHA1 {

public static void main(String[] a) {

try {

MessageDigest md = MessageDigest.getInstance("SHA1");

System.out.println("Message digest object info: ");

System.out.println(" Algorithm = " +md.getAlgorithm());

System.out.println(" Provider = " +md.getProvider());

System.out.println(" ToString = " +md.toString());

String input = "";

md.update(input.getBytes());

byte[] output = md.digest();

System.out.println();

System.out.println("SHA1(\""+input+"\") = " +bytesToHex(output));

input = "abc";

md.update(input.getBytes());

output = md.digest();

System.out.println();

System.out.println("SHA1(\""+input+"\") = " +bytesToHex(output));

input = "abcdefghijklmnopqrstuvwxyz";

md.update(input.getBytes());

output = md.digest();

System.out.println();

System.out.println("SHA1(\"" +input+"\") = " +bytesToHex(output));

System.out.println(""); }

catch (Exception e) {

System.out.println("Exception: " +e);

}

}

public static String bytesToHex(byte[] b)

{

char hexDigit[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

StringBuffer buf = new StringBuffer();

for (int j=0; j<b.length; j++) {

buf.append(hexDigit[(b[j] >> 4) & 0x0f]);

buf.append(hexDigit[b[j] & 0x0f]); }

return buf.toString(); }

}

**OUTPUT:**

