**CSA5179- Cryptography and Network Security**

**Manual Practical.**

Ex.No : 1 **To implement Caesar Cipher**

Date : 01-08-2023

1. **To implement Caesar Cipher in C program :-**

**Program :**

#include<stdio.h>

#include<string.h>

#include<ctype.h>

int main(){

int k=3;

char input[1000],cipher[1000];

printf("Enter input string:");

scanf("%s",&input);

printf("Encrytption is : ");

for(int i=0;i<strlen(input);i++){

if(islower(input[i])){

cipher[i]=((input[i]-'a'+k)%26)+'a';

}

else{

cipher[i]=((input[i]-'A'+k)%26)+'A';

}

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

}

printf("\nDecrytption is : ");

for(int i=0;i<strlen(cipher);i++){

if(islower(input[i])){

input[i]=((cipher[i]-'a'-k)%26)+'a';

}

else{

input[i]=((cipher[i]-'A'-k)%26)+'A';

}

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

}

}

**Output:**

Enter the String : apple

Encryption is : dssoh

Decryption is : apple

Ex.No : 2 **To Implement Play fair cipher**

Date : 01-08-2023

1. **To implement play fair cipher in C programming**

**Program :**

#include<stdio.h>

int check(char table[5][5], char k) {

int i, j;

for (i = 0; i < 5; ++i)

for (j = 0; j < 5; ++j) {

if (table[i][j] == k)

return 0;

}

return 1;

}

void main() {

int i, j, key\_len;

char table[5][5];

for (i = 0; i < 5; ++i)

for (j = 0; j < 5; ++j)

table[i][j] = '0';

printf("\*\*\*\*\*\*\*\*\*\*Playfair Cipher\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("Enter the length of the Key. ");

scanf("%d", &key\_len);

char key[key\_len];

printf("Enter the Key. ");

for (i = -1; i < key\_len; ++i) {

scanf("%c", &key[i]);

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

key[i] = 'i';

}

int flag;

int count = 0;

// inserting the key into the table

for (i = 0; i < 5; ++i) {

for (j = 0; j < 5; ++j) {

flag = 0;

while (flag != 1) {

if (count > key\_len)

goto l1;

flag = check(table, key[count]);

++count;

}// end of while

table[i][j] = key[(count - 1)];

}// end of inner for

}// end of outer for

l1: printf("\n");

int val = 97;

//inserting other alphabets

for (i = 0; i < 5; ++i) {

for (j = 0; j < 5; ++j) {

if (table[i][j] >= 97 && table[i][j] <= 123) {

} else {

flag = 0;

while (flag != 1) {

if ('j' == (char) val)

++val;

flag = check(table, (char) val);

++val;

}// end of while

table[i][j] = (char) (val - 1);

}//end of else

}// end of inner for

}// end of outer for

printf("The table is as follows:\n");

for (i = 0; i < 5; ++i) {

for (j = 0; j < 5; ++j) {

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

}

printf("\n");

}

int l = 0;

printf("\nEnter the length length of plain text.(without spaces) ");

scanf("%d", &l);

printf("\nEnter the Plain text. ");

char p[l];

for (i = -1; i < l; ++i) {

scanf("%c", &p[i]);

}

for (i = -1; i < l; ++i) {

if (p[i] == 'j')

p[i] = 'i';

}

printf("\nThe replaced text(j with i)");

for (i = -1; i < l; ++i)

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

count = 0;

for (i = -1; i < l; ++i) {

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

count = count + 1;

}

printf("\nThe cipher has to enter %d bogus char.It is either 'x' or 'z'\n",

count);

int length = 0;

if ((l + count) % 2 != 0)

length = (l + count + 1);

else

length = (l + count);

printf("\nValue of length is %d.\n", length);

char p1[length];

//inserting bogus characters.

char temp1;

int count1 = 0;

for (i = -1; i < l; ++i) {

p1[count1] = p[i];

if (p[i] == p[i + 1]) {

count1 = count1 + 1;

if (p[i] == 'x')

p1[count1] = 'z';

else

p1[count1] = 'x';

}

count1 = count1 + 1;

}

//checking for length

char bogus;

if ((l + count) % 2 != 0) {

if (p1[length - 1] == 'x')

p1[length] = 'z';

else

p1[length] = 'x';

}

printf("The final text is:");

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

printf("%c ", p1[i]);

char cipher\_text[length];

int r1, r2, c1, c2;

int k1;

for (k1 = 1; k1 <= length; ++k1) {

for (i = 0; i < 5; ++i) {

for (j = 0; j < 5; ++j) {

if (table[i][j] == p1[k1]) {

r1 = i;

c1 = j;

} else if (table[i][j] == p1[k1 + 1]) {

r2 = i;

c2 = j;

}

}

}

if (r1 == r2) {

cipher\_text[k1] = table[r1][(c1 + 1) % 5];

cipher\_text[k1 + 1] = table[r1][(c2 + 1) % 5];

}

else if (c1 == c2) {

cipher\_text[k1] = table[(r1 + 1) % 5][c1];

cipher\_text[k1 + 1] = table[(r2 + 1) % 5][c1];

} else {

cipher\_text[k1] = table[r1][c2];

cipher\_text[k1 + 1] = table[r2][c1];

}

k1 = k1 + 1;

}//end of for with k1

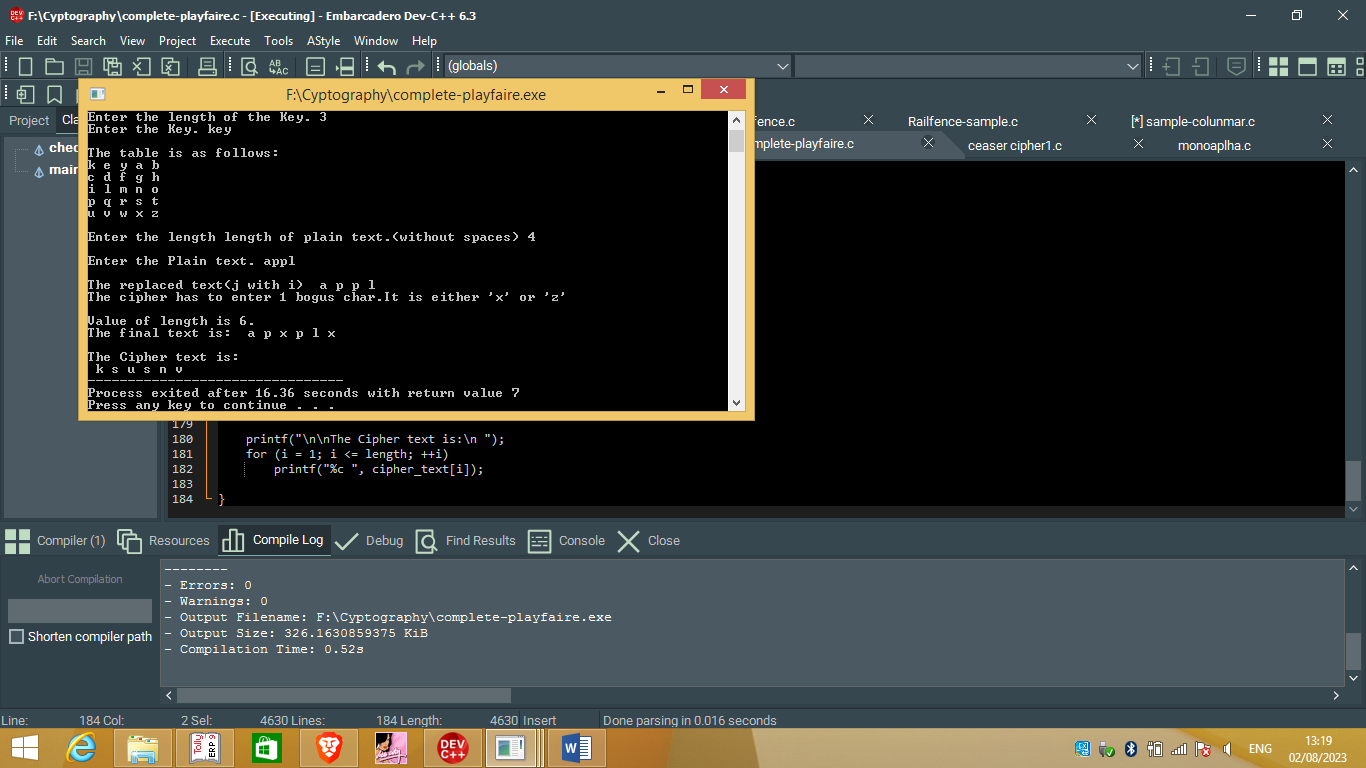
printf("\n\nThe Cipher text is:\n ");

for (i = 1; i <= length; ++i)

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

}

**Output:**



Ex.no : 3 **To implement mono-alphabetic cipher**

Date : 02-08-2023

1. **To implement monoalphabetic cipher in C programming**

**Program:**

#include<stdio.h>

int main(){

char alpha[100]="abcdefghijklmnopqrstuvwxyz",key[100]="zyxwvutsrqponmlkjihgfedcba",plain[100],cipher[100];

int m=0,index[100];

printf("Enter plain text :");

scanf("%s",&plain);

for(int i=0;i<strlen(plain);i++){

for(int j=0;j<strlen(alpha);j++){

if(plain[i]==alpha[j]){

index[m]=j;

m++;

}

}

}

printf("Cipher text: ");

for(int i=0;i<strlen(plain);i++){

cipher[i]=key[index[i]];

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

}

printf("\n Plain text : ");

for(int i=0;i<strlen(plain);i++){

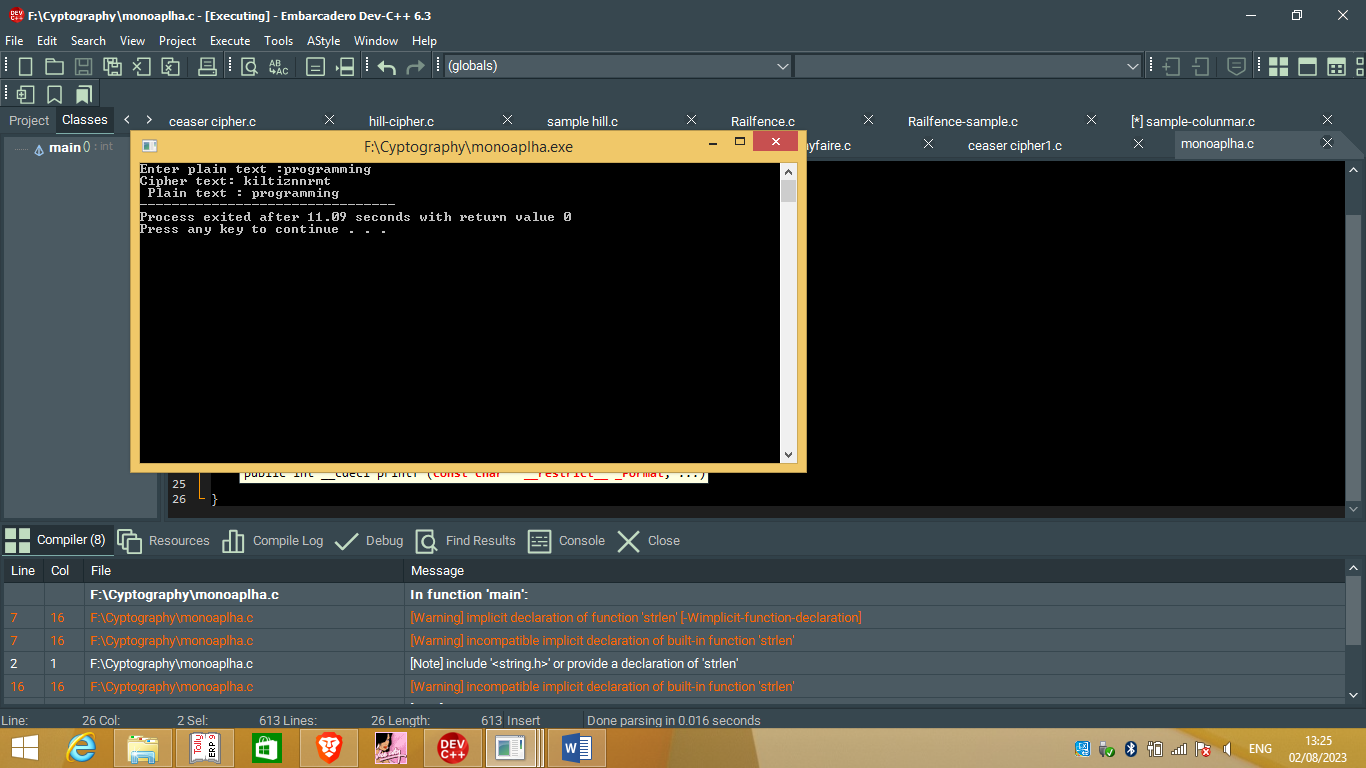
plain[i]=alpha[index[i]];

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

}

}

**Output:**



Ex.no : 4 **To implement Hill cipher**

Date : 02-08-2023

1. **To implement hill cipher in C programming.**

**Program:**

#include<stdio.h>

#include<string.h>

int en[100][100],m[100][100],msg[100];

char ms[100];

void getkeymatrix(){

printf("Enter message: ");

scanf("%s",&ms);

for(int i=0;i<strlen(ms);i++){

msg[i]=ms[i]-65;

}

for(int i=0;i<strlen(ms);i++){

for(int j=0;j<strlen(ms);j++){

scanf("%d",&m[i][j]);

}

}

}

void encryption(){

int i, j, k,n,o;

for(i = 0,n=0; i < strlen(ms); i++,n++)

for(j = 0; j < strlen(ms); j++)

for(k = 0,o=0; k < strlen(ms); k++,o++)

en[i][j] = en[i][j] + m[n][k] \* msg[k];

for(i = 0; i < strlen(ms); i++){

printf("%c ",(en[i][0]%26)+65);

}

}

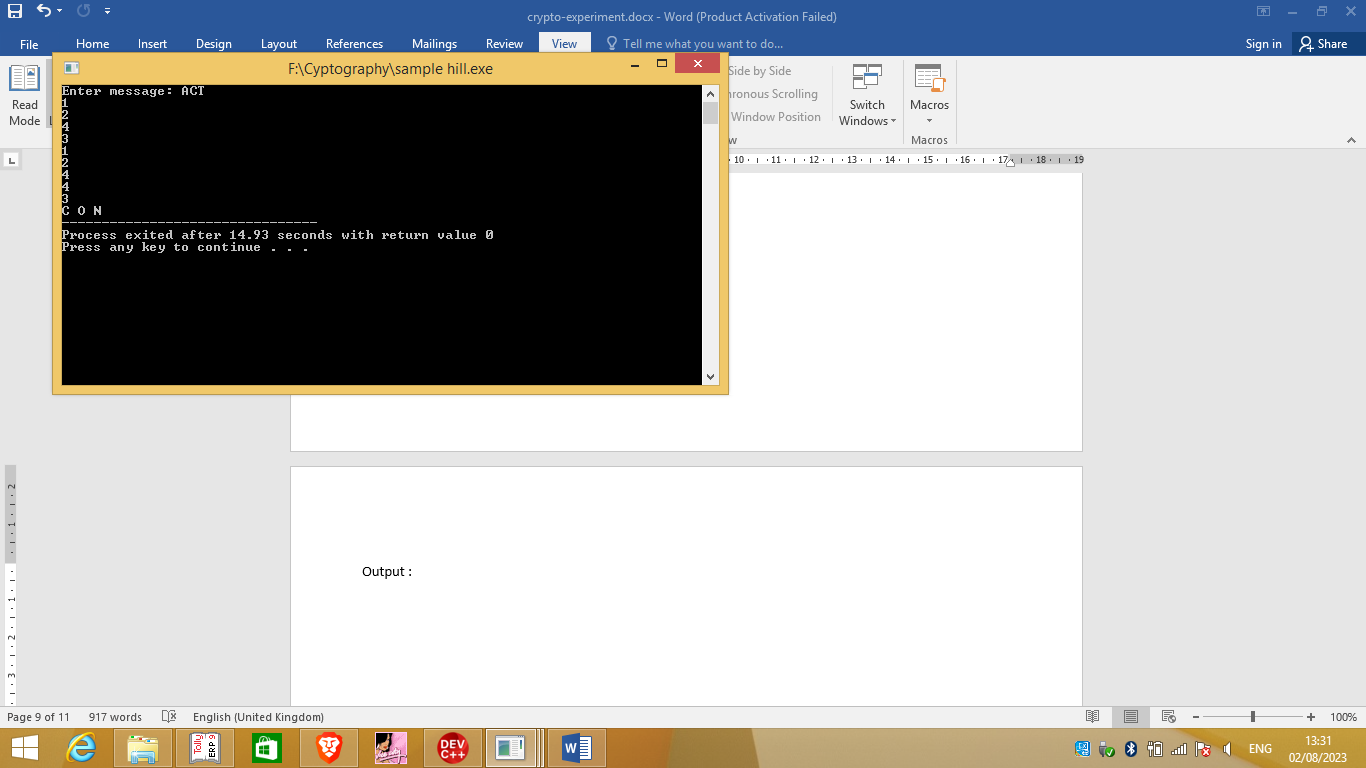
int main(){

getkeymatrix();

encryption();

}

**Output :**



Ex. No:5 **To Implement RailFence Algorithm**

Date: 02-08-2023

1. **To Implement Railfence cipher in C programming .**

**Program :**

#include<stdio.h>

int main(){

int count=0,len=0,rail=0,j=0;

char str[100],code[100][100];

scanf("%d",&rail);

printf("Enter string : ");

scanf("%s",&str);

len=strlen(str);

while(j<len){

if(count%2==0){

for(int i=0;i<rail;i++){

code[i][j]=str[j];

j++;

}

}

else{

for(int i=rail-2;i>0;i--){

code[i][j]=str[j];

j++;

}

}

count++;

}

for(int i=0;i<rail;i++){

for(int j=0;j<len;j++){

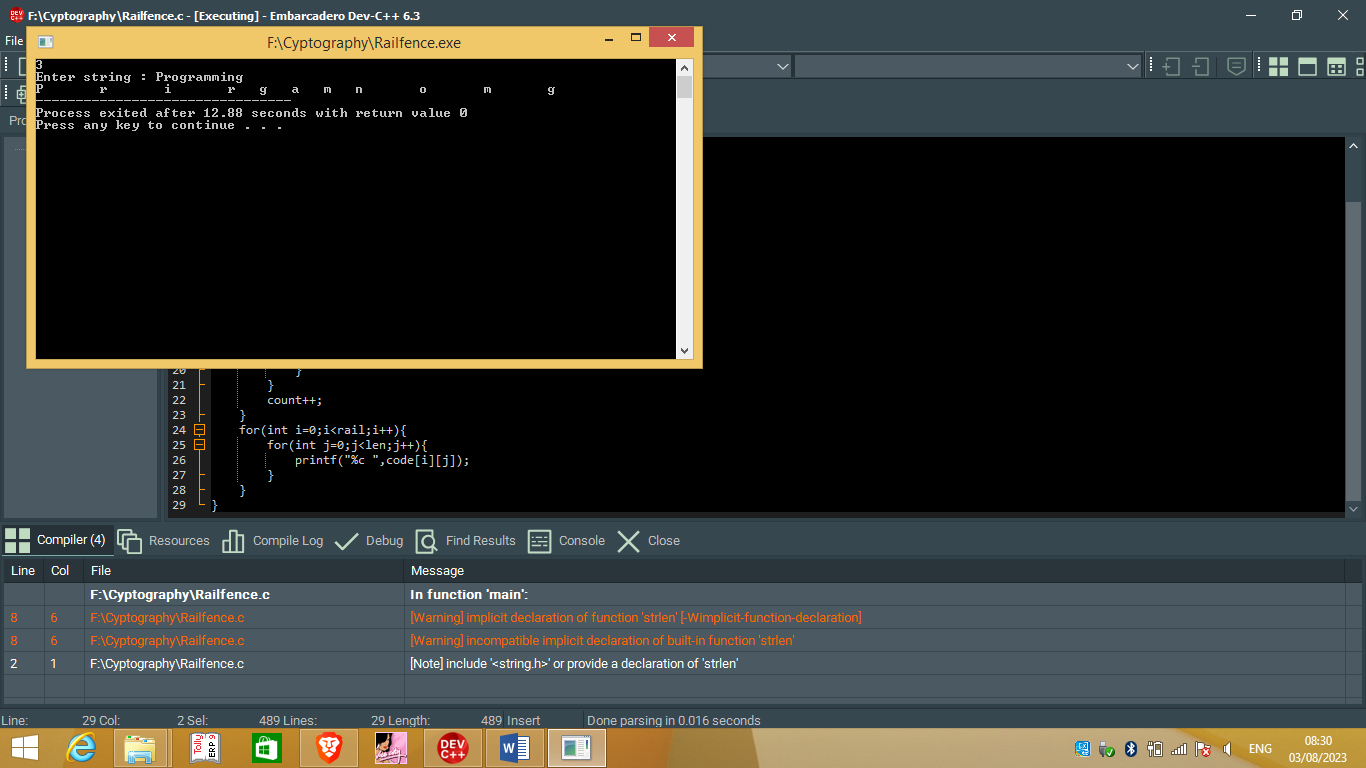
printf("%c ",code[i][j]);

}

}

}

**Output:**



Ex. No: 6 **To implement Columnar Cipher**

Date: 02-08-3-2023

1. **To implement Columnar cipher in C programming**

**Program:**

#include<stdio.h>

#include<string.h>

void encrypt(char message[],int key){

int len=strlen(message),row=(len+key-1)/key,m=0;

char encry[100][100];

int index=0;

for(int i=0;i<row;i++){

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

if(m<len){

encry[i][j]=message[m];

m++;

}

else{

encry[i][j]='X';

}

}

}

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

for(int i=0;i<row;i++){

if(encry[i][j]!='X')

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

}

}

}

int main(){

char message[100];

int key;

scanf("%s",&message);

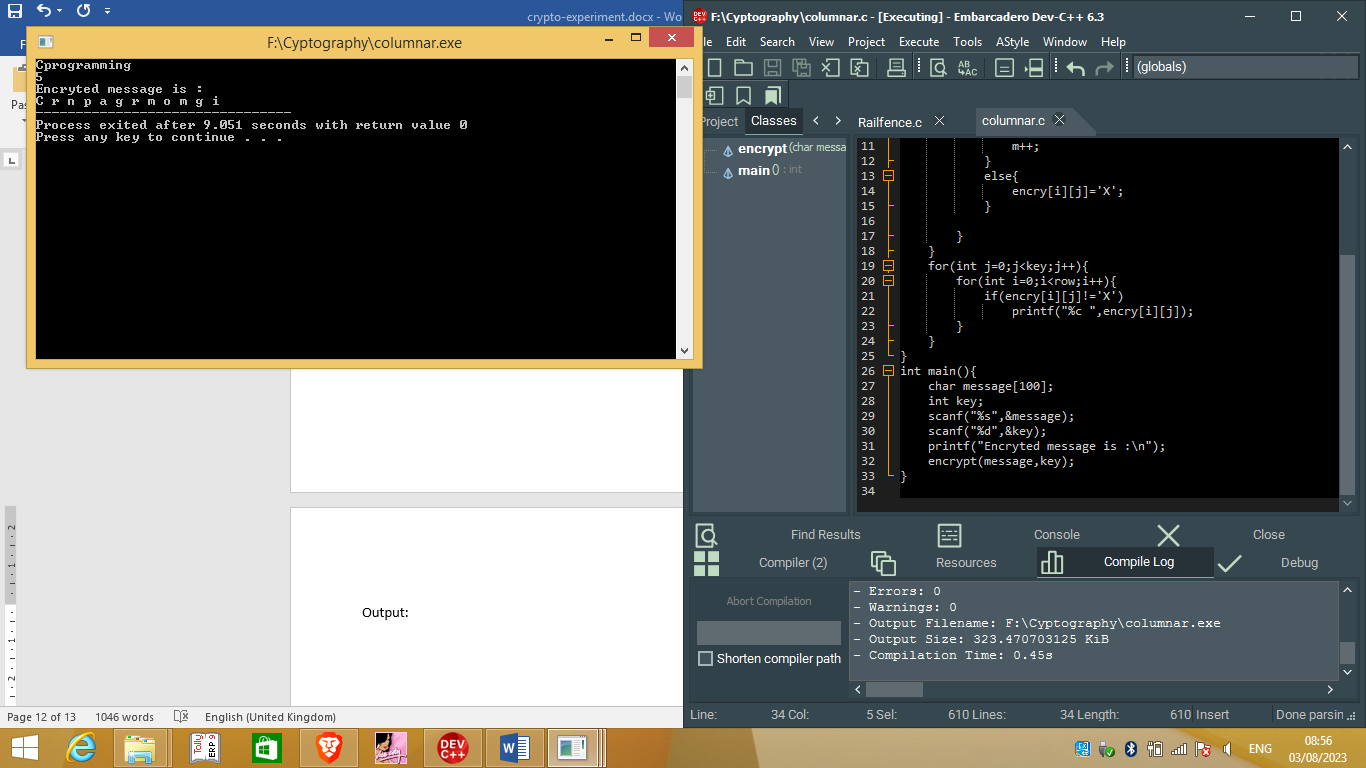
scanf("%d",&key);

printf("Encryted message is :\n");

encrypt(message,key);

}

**Output:**



Ex.no:7 **To implement RSA Algorithm**

Date:

1. **To implement RSA algorithm in c program.**

**Program:**

#include<stdio.h>

#include<conio.h>

int main(){

int c,p,q,n,n1,i,j,m=5,result=0,d[1000],result2=0,temp;

printf("Enter a value : ");

scanf("%d",&p);

printf("Enter another value : ");

scanf("%d",&q);

n=p\*q;

printf("Value of n = %d\n",n);

n1=(p-1)\*(q-1);

printf("Value of n1 = %d\n",n1);

int e[10]={3,5,7,11,13,17};

for(i=0;i<e[i];i++){

if(n1%n1==0&&n1%e[i]==0){

result=e[i];

break;

}

}

printf("The value of e is %d\n",result);

for(i=0;i<e[i] && result2!=1;i++){

for(j=1;j<1000;j++){

result2=(j\*e[i])%n1;

if(result2==1){

break;

}

}

printf("The value of d is %d\n",j);

}

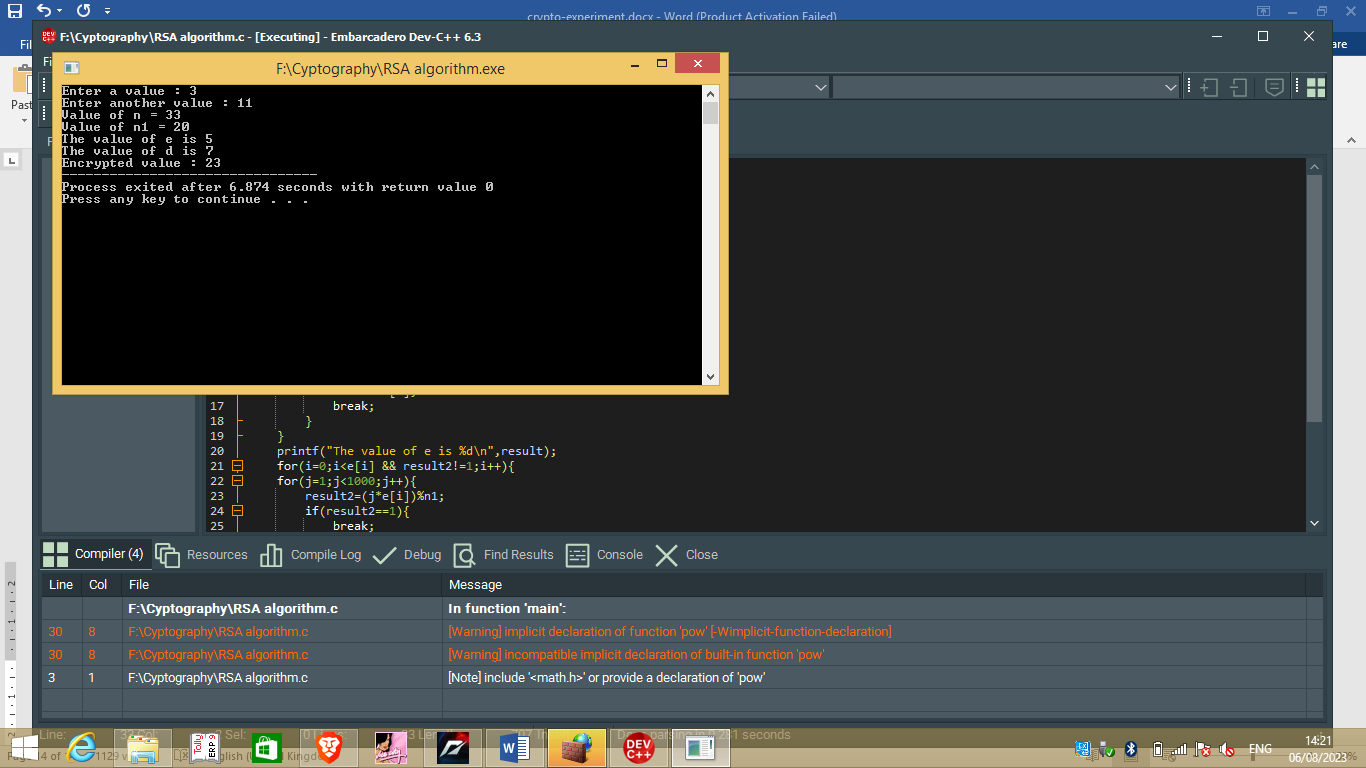
temp=(pow(m,result));

c=temp%n;

printf("Encrypted value : %d",c);

}

**Output:**



Ex.no:8 **To implement Diffe-hellman algorithm**

Date: 04-08-2023

1. **To implement Diffe-hellman algorithm in c program.**

**Program**

#include<stdio.h>

#include<conio.h>

#include<math.h>

int main(){

int q,b,Xa,Xb,Ya,Yb,K1,K2,temp1,temp2,temp3,temp4;

printf("Enter the value of q : ");

scanf("%d",&q);

printf("Enter the value of alpha : ");

scanf("%d",&b);

printf("Enter the value of Xa : ");

scanf("%d",&Xa);

printf("Enter the value of Xb : ");

scanf("%d",&Xb);

temp1=(pow(b,Xa));

Ya=temp1%q;

printf("Ya = %d\n",Ya);

temp2=(pow(b,Xb));

Yb=temp2%q;

printf("Yb = %d\n",Yb);

temp3=(pow(Yb,Xa));

K1=temp3%q;

temp4=(pow(Ya,Xb));

K2=temp4%q;

if(K1==K2){

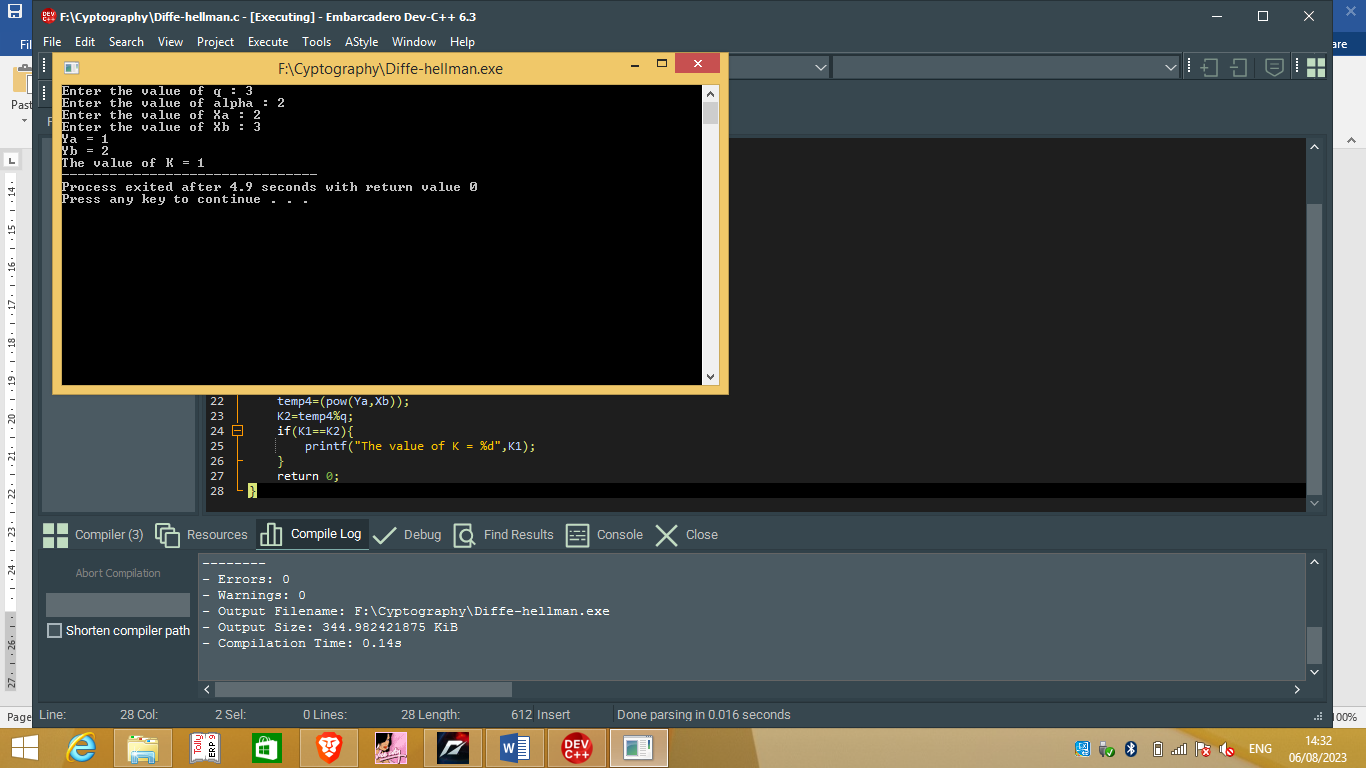
printf("The value of K = %d",K1);

}

return 0;

}

**Output:**



Ex.no:9 **To implement DES algorithm**

Date:04-08-2023

1. **To implement DES algorithm**

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>DES Encryption and Decryption</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>

</head>

<body>

<script>

// DES encryption using crypto-js library in a browser

function desEncrypt(input, key) {

const encryptedData = CryptoJS.DES.encrypt(input, key).toString();

return encryptedData;

}

// DES decryption using crypto-js library in a browser

function desDecrypt(input, key) {

const decryptedData = CryptoJS.DES.decrypt(input, key).toString(CryptoJS.enc.Utf8);

return decryptedData;

}

const plaintext = 'Hello, DES!';

const key = 'ThisIs64BitKey';

const encryptedText = desEncrypt(plaintext, key);

console.log('Encrypted:', encryptedText);

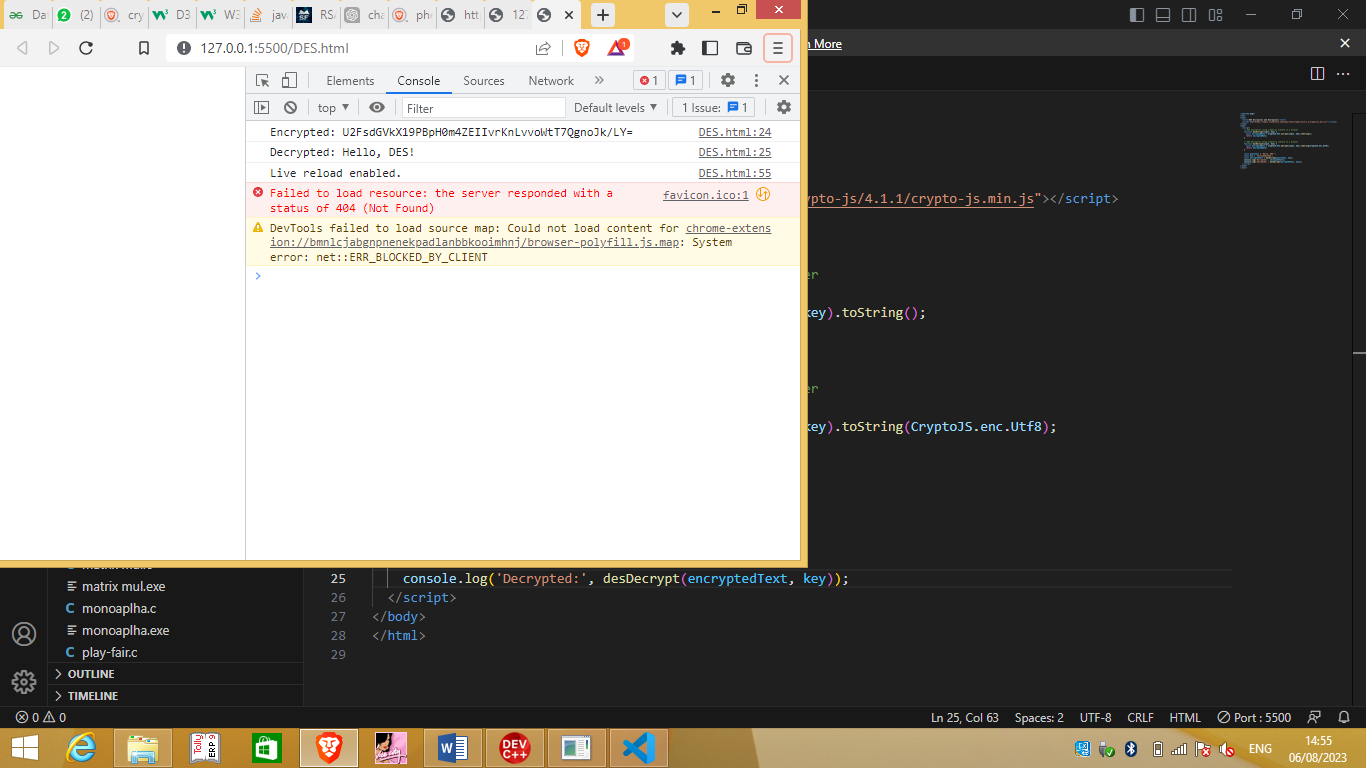
console.log('Decrypted:', desDecrypt(encryptedText, key));

</script>

</body>

</html>

**Output:**



Ex.no: 10 **To implement SHA algorithm**

Date:05-04-2023

1. **To implement SHA algorithm**

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>SHA-256 Hashing</title>

</head>

<body>

<script>

async function sha256(input) {

const encoder = new TextEncoder();

const data = encoder.encode(input);

const hashBuffer = await crypto.subtle.digest('SHA-256', data);

const hashArray = Array.from(new Uint8Array(hashBuffer));

const hashHex = hashArray.map(byte => byte.toString(16).padStart(2, '0')).join('');

return hashHex;

}

const data = 'Hello, SHA-256!';

sha256(data).then(hashValue => {

console.log('SHA-256:', hashValue);

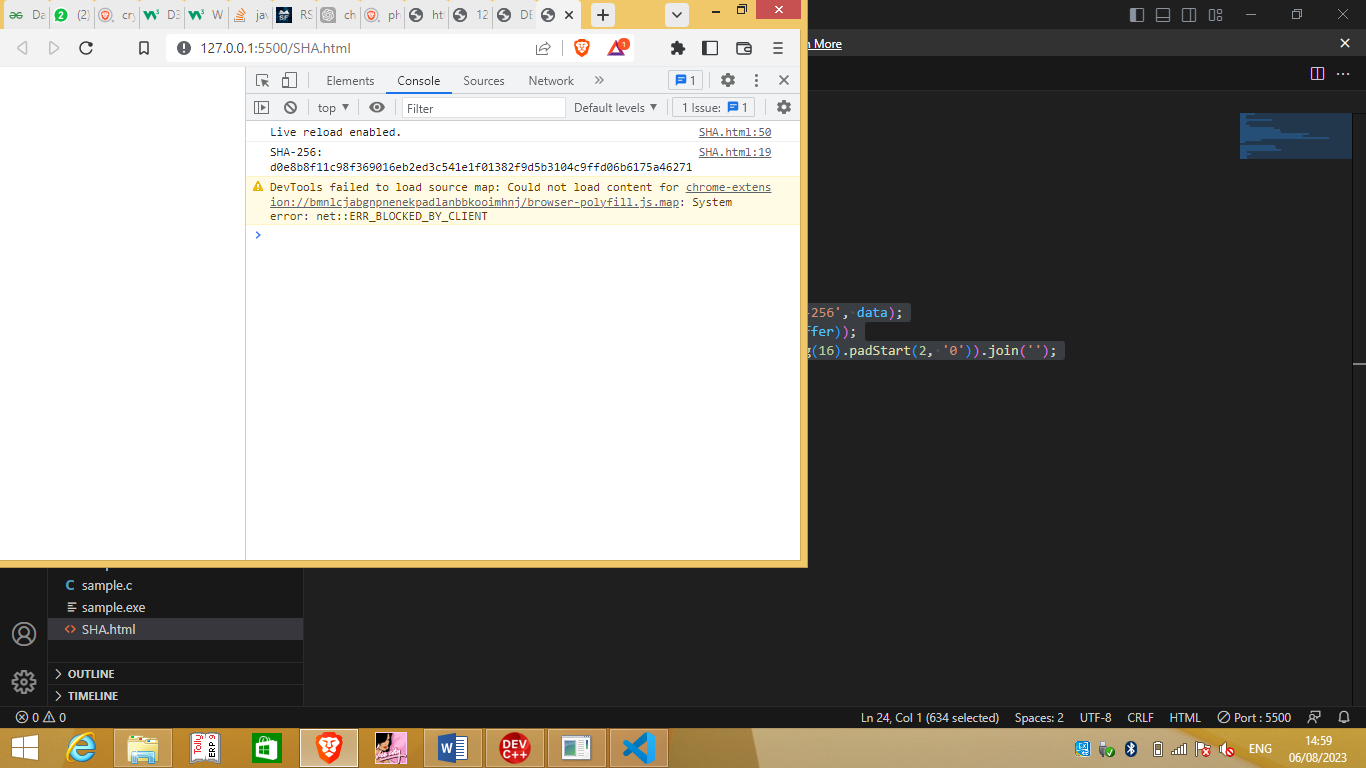
});

</script>

</body>

</html>

**Output:**



Ex.no: 11 **To implement MD5 algorithm**

Date:05-04-2023

1. **To implement MD5 algorithm**

**Program:**

<!DOCTYPE html>

<html>

<head>

<title>SHA-256 Hashing</title>

</head>

<body>

<script>

async function MD5(input) {

const encoder = new TextEncoder();

const data = encoder.encode(input);

const hashBuffer = await crypto.subtle.digest('SHA-256', data);

const hashArray = Array.from(new Uint8Array(hashBuffer));

const hashHex = hashArray.map(byte => byte.toString(16).padStart(2, '0')).join('');

return hashHex;

}

const data = 'Hello, SHA-256!';

MD5(data).then(hashValue => {

console.log('SHA-256:', hashValue);

});

</script>

</body>

</html>

**Output:**

