Vernam Cipher

#include <stdio.h>

#include <stdlib.h>

char a[27]={'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'};

int rnd[27];

int code[10],i,j;

char in[10],temp,out[10];

int main (void)

{ //----------------------------input-------------------------

printf("Please enter the input");

for(i=0;i<10;i++)

{

scanf("%c",&temp);

if (temp == 32 || temp =='\0')

{

continue;

}

else

{

in[i]=temp;

}

}

//--------------------random genration & preserving it -------------------

for (i=0;i<10;i++)

{

for(j=0;j<27;j++)

{

if (in[i]==a[j])

{

rnd[j]= rand()%100;

}

}

}

//-------------------------Vernam operation---------------

for (i=0;i<10;i++)

{

for(j=0;j<27;j++)

{

if (rnd[j]!=0)

{

code[i]= (in[i] + rnd[j])%26;

}

}

}

//----------------Actual Text conversion from code --------------

for (i=0;i<10;i++)

{

for(j=0;j<27;j++)

{

if (code[i]==j)

{

out[i]=a[j];

}

}

}

//--------------OUT------------------------

for(i=0;i<10;i++)

{

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

}

return 0;

}

OUTPUT: ./a.out

Please enter the input:hello world

encryption: iifmmpixps

DES   
ALGORITHM

#include<stdio.h>  
int main()  
{  
int i, cnt=0, p8[8]={6,7,8,9,1,2,3,4};  
int p10[10]={6,7,8,9,10,1,2,3,4,5};  
  
char input[11], k1[10], k2[10], temp[11];  
char LS1[5], LS2[5];  
//k1, k2 are for storing interim keys  
//p8 and p10 are for storing permutation key  
  
//Read 10 bits from user...  
printf("Enter 10 bits input:");  
scanf("%s",input);   
input[10]='\0';  
  
//Applying p10...  
for(i=0; i<10; i++)  
{  
cnt = p10[i];  
temp[i] = input[cnt-1];  
}  
temp[i]='\0';  
printf("\nYour p10 key is    :");  
for(i=0; i<10; i++)  
{ printf("%d,",p10[i]); }  
  
printf("\nBits after p10     :");  
puts(temp);  
//Performing LS-1 on first half of temp  
for(i=0; i<5; i++)  
{  
if(i==4)  
temp[i]=temp[0];  
else  
temp[i]=temp[i+1];   
}  
//Performing LS-1 on second half of temp  
for(i=5; i<10; i++)  
{  
if(i==9)  
temp[i]=temp[5];  
else  
temp[i]=temp[i+1];   
}  
printf("Output after LS-1  :");  
puts(temp);  
  
printf("\nYour p8 key is     :");  
for(i=0; i<8; i++)  
{ printf("%d,",p8[i]); }  
  
//Applying p8...  
for(i=0; i<8; i++)  
{  
cnt = p8[i];  
k1[i] = temp[cnt-1];  
}  
printf("\nYour key k1 is     :");  
puts(k1);   
//This program can be extended to generate k2 as per DES algorithm.  
}

Output:./a.out

Enter 10 bits input:1 0 1 0 1 0 1 0 1 0

Your p10 key is :6,7,8,9,10,1,2,3,4,5,

Bits after p10 :

Your p8 key is :6,7,8,9,1,2,3,4,

Your key k1 is :

Rail Fence Algorithm

#include<stdio.h>

#include<string.h>

void encryptMsg(char msg[], int key){

int msgLen = strlen(msg), i, j, k = -1, row = 0, col = 0;

char railMatrix[key][msgLen];

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

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

railMatrix[i][j] = '\n';

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

railMatrix[row][col++] = msg[i];

if(row == 0 || row == key-1)

k= k \* (-1);

row = row + k;

}

printf("\nEncrypted Message: ");

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

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

if(railMatrix[i][j] != '\n')

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

}

void decryptMsg(char enMsg[], int key){

int msgLen = strlen(enMsg), i, j, k = -1, row = 0, col = 0, m = 0;

char railMatrix[key][msgLen];

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

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

railMatrix[i][j] = '\n';

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

railMatrix[row][col++] = '\*';

if(row == 0 || row == key-1)

k= k \* (-1);

row = row + k;

}

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

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

if(railMatrix[i][j] == '\*')

railMatrix[i][j] = enMsg[m++];

row = col = 0;

k = -1;

printf("\nDecrypted Message: ");

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

printf("%c", railMatrix[row][col++]);

if(row == 0 || row == key-1)

k= k \* (-1);

row = row + k;

}

}

int main(){

char msg[] = "Hello World";

char enMsg[] = "Horel ollWd";

int key = 3;

printf("Original Message: %s", msg);

encryptMsg(msg, key);

decryptMsg(enMsg, key);

return 0;

}

Output : ./a.out

Original Message: Hello World

Encrypted Message: Horel ollWd

CAESAR CIPHER

#include <stdio.h>

#include <ctype.h>

#define MAXSIZE 1024

void encrypt(char\*);

void decrypt(char\*);

int menu();

int

main(void)

{

char c,

choice[2],

s[MAXSIZE];

while(1)

{

menu();

gets(choice);

if((choice[0]=='e')||(choice[0]=='E'))

{

puts("Input text to encrypt->");

gets(s);

encrypt(s);

}

else if((choice[0]=='d')||(choice[0]=='D'))

{

puts("Input text to decrypt->");

gets(s);

decrypt(s);

}

else

break;

}

return 0;

}

void encrypt(char\*str)

{

int n=0;

char \*p=str,

q[MAXSIZE];

while(\*p)

{

if(islower(\*p))

{

if((\*p>='a')&&(\*p<'x'))

q[n]=toupper(\*p + (char)3);

else if(\*p=='x')

q[n]='A';

else if(\*p=='y')

q[n]='B';

else

q[n]='C';

}

else

{

q[n]=\*p;

}

n++; p++;

}

q[n++]='\0';

puts(q);

}

void decrypt(char\*str)

{

int n=0;

char \*p=str,

q[MAXSIZE];

while(\*p)

{

if(isupper(\*p))

{

if((\*p>='D')&&(\*p<='Z'))

q[n]=tolower(\*p - (char)3);

else if(\*p=='A')

q[n]='x';

else if(\*p=='B')

q[n]='y';

else

q[n]='z';

}

else

{

q[n]=\*p;

}

n++; p++;

}

q[n++]='\0';

puts(q);

}

int menu()

{

puts("To encrypt, input e or E\n");

puts("To decrypt, input d or D\n");

puts("To exit, input any other letter\n");

puts("Your choice:->\n");

return 0;

}

OUTPUT: ./a.out

To encrypt, input e or E

To decrypt, input d or D

To exit, input any other letter

Your choice:->

e

Input text to encrypt->

hello

KHOOR

To encrypt, input e or E

To decrypt, input d or D

To exit, input any other letter

Your choice:->

d

Input text to decrypt->

KHOOR

hello

To encrypt, input e or E

To decrypt, input d or D

To exit, input any other letter

MODIFIED CIPHER

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void decrypt(char arr[])

{

int i;

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

{

arr[i] = arr[i] + 10;

}

}

void encrypt(char arr[])

{

int i;

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

{

arr[i] = arr[i] - 10;

}

}

int main()

{

char password[40];

int ch;

printf("Enter a Password:\t");

scanf("%s", password);

printf("\nPassword:\t%s\n",password);

encrypt(password);

printf("\nEncrypted Password:\t%s\n", password);

decrypt(password);

printf("\nDecrypted Password:\t%s\n", password);

return 0;

}

OUTPUT :./a.out

Enter a Password: kishor

Password: kishor

Encrypted Password: `juhs.e

Decrypted Password: kishor