#include<iostream>

#include<bits/stdc++.h>

using namespace std;

int w0[8]; int w1[8]; int w[8]; int arr[10]; int w3[8];

int w2[8]; int w4[8]; int w5[8];

int binary[4]; int temp1[8]; int temp2[8];

int\* bin1; int\* bin2;

string pl\_txt,key;

int key1[16],key2[16];

int res[16];

int s00[4],s01[4],s10[4],s11[4]; int s\_arr1[16];

int\* t1; int\* t2; int\* t3; int\* t4 ;

int\* p1; int\* p2; int\* p3; int\* p4 ;

int t11[4],t12[4],t13[4],t14[4];

int p11[4],p12[4],p13[4],p14[4];

int sbox[] = {9,4,10,11,13,1,8,5,6,2,0,3,12,14,15,7};

int dec\_sbox[] = {10,5,9,11,1,7,8,15,6,0,2,3,12,4,13,14};

int s1[] = {1,0,0,0,0,0,0,0};

int s2[] = {0,0,1,1,0,0,0,0};

int GF\_mul\_1[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};

int GF\_mul\_4[] = {0,4,8,12,3,7,11,15,6,2,14,10,5,1,13,9};

int GF\_mul\_2[] = {0,2,4,6,8,10,12,14,3,1,7,5,11,9,15,13};

int GF\_mul\_9[] = {0,9,1,8,2,11,3,10,4,13,5,12,6,15,7,14};

int mat1[] = {1,4,4,1};

int convert\_to\_integer(int nibble[]){

int n;

n = 1\*nibble[3] + 2\*nibble[2] + 4\*nibble[1] + 8\*nibble[0];

return n;

}

int\* convert\_to\_binary(int num){

int p=0;

int rem;

for(int i=0;i<4;i++)

arr[i] = 0;

// cout<<"Num is: "<<num;

while(num>0){

rem = num%2;

num = num/2;

arr[p] = rem;

p++;

}

int j=0;

for(int i=3; i>=0; i--){

binary[j] = arr[i];

j++;

}

/\*cout<<"In convert\_to\_binary is: ";

for(int i=0;i<4;i++)

cout<<binary[i];

\*/

return binary;

}

int\* rotNib(int w[]){

for(int i=0;i<8;i++){

if(i<4)

temp1[i+4] = w[i];

else

temp1[i-4] = w[i];

}

return temp1;

}

int\* subNib(int rw[],int sbx[]){

int first\_nibble[4]; int second\_nibble[4];

int n1,n2;

for(int i=0;i<8;i++){

if(i<4)

first\_nibble[i] = rw[i];

else

second\_nibble[i-4] = rw[i];

}

n1 = convert\_to\_integer(first\_nibble);

n2 = convert\_to\_integer(second\_nibble);

//cout<<endl<<"n1: "<<n1<<" n2: "<<n2;

bin1 = convert\_to\_binary(sbx[n1]);

//cout<<endl<<"bin1 is: ";

for(int i=0;i<4;i++){

//cout<<bin1[i];

temp2[i] = bin1[i];

}

bin2 = convert\_to\_binary(sbx[n2]);

//cout<<endl<<"bin2 is: ";

for(int i=0;i<4;i++){

//cout<<bin2[i];

temp2[i+4] = bin2[i];

}

cout<<endl<<"IN subNib is(temp2): ";

for(int i=0;i<8;i++)

cout<<temp2[i];

return temp2;

}

int\* calculate\_w2(int w0[],int w1[],int s[]){

int wi[8]; int wj[8]; int\* rotate; int\* sub;

rotate = rotNib(w1);

for(int i=0;i<8;i++)

wi[i] = rotate[i];

cout<<endl<<"IN rotNib is(temp1): ";

for(int i=0;i<8;i++)

cout<<wi[i];

sub = subNib(wi,sbox);

for(int i=0;i<8;i++)

wj[i] = sub[i];

cout<<endl<<"IN SubNib is(temp2): ";

for(int i=0;i<8;i++)

cout<<wj[i];

for(int i=0;i<8;i++){

wi[i] = w0[i] ^ s[i];

w[i] = wi[i] ^ wj[i];

}

return w;

}

int\* addRoundKey(){

cout<<" addRoundKey res: ";

for(int i=0;i<16;i++){

res[i] = pl\_txt[i] ^ key[i];

cout<<res[i];

}

return res;

}

int\* MixColumn(int arr[], int GF\_mul[],int GF\_mul1[]){

int arr\_nib1[4],arr\_nib2[4],arr\_nib3[4],arr\_nib4[4];

int a,b,c,d;

for(int i=0;i<16;i++){

if(i<4)

arr\_nib1[i] = arr[i];

else if(i>3 && i<8)

arr\_nib2[i-4] = arr[i];

else if(i>7 && i<12)

arr\_nib3[i-8] = arr[i];

else

arr\_nib4[i-12] = arr[i];

}

a = convert\_to\_integer(arr\_nib1);

b = convert\_to\_integer(arr\_nib2);

c = convert\_to\_integer(arr\_nib3);

d = convert\_to\_integer(arr\_nib4);

t1 = convert\_to\_binary(GF\_mul[c]);

for(int i=0;i<4;i++)

t11[i] = t1[i];

p4 = convert\_to\_binary(GF\_mul1[d]);

for(int i=0;i<4;i++)

p14[i] = p4[i];

for(int i=0;i<4;i++){

s11[i] = t11[i] ^ p14[i];

//cout<<t1[i];

}

t2 = convert\_to\_binary(GF\_mul[b]);

for(int i=0;i<4;i++)

t12[i] = t2[i];

p2 = convert\_to\_binary(GF\_mul1[a]);

for(int i=0;i<4;i++)

p12[i] = p2[i];

for(int i=0;i<4;i++){

s00[i] = p12[i] ^ t12[i];

//cout<<t2[i];

}

t3 = convert\_to\_binary(GF\_mul[a]);

for(int i=0;i<4;i++)

t13[i] = t3[i];

p3 = convert\_to\_binary(GF\_mul1[b]);

for(int i=0;i<4;i++)

p13[i] = p3[i];

for(int i=0;i<4;i++){

s10[i] = p13[i] ^ t13[i];

//cout<<t3[i];

}

t4 = convert\_to\_binary(GF\_mul[d]);

for(int i=0;i<4;i++)

t14[i] = t4[i];

p1 = convert\_to\_binary(GF\_mul1[c]);

for(int i=0;i<4;i++)

p11[i] = p1[i];

for(int i=0;i<4;i++){

s01[i] = p11[i] ^ t14[i];

}

cout<<endl<<"S00 is: ";

for(int i=0;i<4;i++)

cout<<s00[i];

cout<<" s10 is: ";

for(int i=0;i<4;i++)

cout<<s10[i];

cout<<" s01 is: ";

for(int i=0;i<4;i++)

cout<<s01[i];

cout<<" s11 is: ";

for(int i=0;i<4;i++)

cout<<s11[i];

for(int i=0;i<16;i++){

if(i<4)

s\_arr1[i] = s00[i];

else if(i>3 && i<8)

s\_arr1[i] = s10[i-4];

else if(i>7 && i<12)

s\_arr1[i] = s01[i-8];

else

s\_arr1[i] = s11[i-12];

}

cout<<endl<<"S\_arr1 is: ";

for(int i=0;i<16;i++)

cout<<s\_arr1[i];

return s\_arr1;

}

int main(){

//Accept inputs

cout<<"Enter the Plain text(16-bits): ";

cin>>pl\_txt;

cout<<"Enter the key(16-bits): ";

cin>>key;

//divide key into w0 and w1

for(int i=0;i<16;i++){

if(i<8)

w0[i] = key[i]-48;

else

w1[i-8] = key[i]-48;

}

cout<<endl<<"W0 is: ";

for(int i=0;i<8;i++)

cout<<w0[i];

cout<<endl<<"W1 is: ";

for(int i=0;i<8;i++)

cout<<w1[i];

int\* temp\_w2;

temp\_w2 = calculate\_w2(w0,w1,s1);

cout<<endl<<"W2 is: ";

for(int i=0;i<8;i++){

w2[i] = temp\_w2[i];

cout<<w2[i];

}

//Calculate w3

cout<<endl<<"W3 is: ";

for(int i=0;i<8;i++){

w3[i] = w2[i] ^ w1[i];

cout<<w3[i];

}

//Calculation for w4

int\* temp\_w4;

temp\_w4 = calculate\_w2(w2,w3,s2);

cout<<endl<<"W4 is: ";

for(int i=0;i<8;i++){

w4[i] = temp\_w4[i];

cout<<w4[i];

}

//Calculate w5

cout<<endl<<"W5 is: ";

for(int i=0;i<8;i++){

w5[i] = w4[i] ^ w3[i];

cout<<w5[i];

}

for(int i=0;i<16;i++){

if(i<8){

key1[i] = w2[i];

key2[i] = w4[i];

}

else{

key1[i] = w3[i-8];

key2[i] = w5[i-8];

}

}

cout<<endl<<"Key1 is: ";

for(int j=0;j<16;j++)

cout<<key1[j];

cout<<endl<<"Key2 is: ";

for(int j=0;j<16;j++)

cout<<key2[j];

//Encryption Procedure Started

int\* res1;

res1 = addRoundKey();

int div1[8]; int div2[8];

//S-box apply

for(int i=0;i<16;i++){

if(i<8)

div1[i] = res1[i];

else

div2[i-8] = res1[i];

}

int\* div1\_res; int\* div2\_res;

int div\_res\_combine[16];

div1\_res = subNib(div1,sbox);

cout<<" div1\_res is: ";

for(int i=0;i<8;i++){

cout<<div1\_res[i];

div\_res\_combine[i] = div1\_res[i];

}

div2\_res = subNib(div2,sbox);

for(int i=0;i<8;i++){

//cout<<div2\_res[i];

div\_res\_combine[i+8] = div2\_res[i];

}

//swap 2 and 4 nibble

int div\_res\_combine\_swap[16];

for(int i=0;i<16;i++){

if(i<4)

div\_res\_combine\_swap[i] = div\_res\_combine[i];

else if(i>3 && i<8)

div\_res\_combine\_swap[i] = div\_res\_combine[i+8];

else if(i>7 && i<12)

div\_res\_combine\_swap[i] = div\_res\_combine[i];

else

div\_res\_combine\_swap[i] = div\_res\_combine[i-8];

}

cout<<endl<<"div\_res\_combine\_swap: ";

for(int i=0;i<16;i++)

cout<<div\_res\_combine\_swap[i];

int\* s\_arr1\_res; //Mix column

s\_arr1\_res = MixColumn(div\_res\_combine\_swap,GF\_mul\_4,GF\_mul\_1);

int xor\_res[16]; //Add round key1

for(int j=0;j<16;j++)

xor\_res[j] = s\_arr1\_res[j] ^ key1[j];

//Final round (Encryption)

int xor\_res\_fhalf[8]; int xor\_res\_shalf[8];

for(int i=0;i<16;i++){

if(i<8)

xor\_res\_fhalf[i] = xor\_res[i];

else

xor\_res\_shalf[i-8] = xor\_res[i];

}

int\* xor\_res1; int\* xor\_res2;

int xor\_res\_combine[16];

xor\_res1 = subNib(xor\_res\_fhalf,sbox);

cout<<" xor\_res1 is: ";

for(int i=0;i<8;i++){

cout<<xor\_res1[i];

xor\_res\_combine[i] = xor\_res1[i];

}

xor\_res2 = subNib(xor\_res\_shalf,sbox);

for(int i=0;i<8;i++){

cout<<xor\_res2[i];

xor\_res\_combine[i+8] = xor\_res2[i];

}

//swap 2 and 4 nibble

int xor\_res\_combine\_swap[16];

for(int i=0;i<16;i++){

if(i<4)

xor\_res\_combine\_swap[i] = xor\_res\_combine[i];

else if(i>3 && i<8)

xor\_res\_combine\_swap[i] = xor\_res\_combine[i+8];

else if(i>7 && i<12)

xor\_res\_combine\_swap[i] = xor\_res\_combine[i];

else

xor\_res\_combine\_swap[i] = xor\_res\_combine[i-8];

}

cout<<endl<<"xor\_res\_combine\_swap: ";

for(int i=0;i<16;i++)

cout<<xor\_res\_combine\_swap[i];

int cipher[16]; //Add round key2

for(int j=0;j<16;j++)

cipher[j] = xor\_res\_combine\_swap[j] ^ key2[j];

cout<<endl<<"Cipher Text is: ";

for(int k=0;k<16;k++)

cout<<cipher[k];

//Decryption procedure

int dec\_xor\_res[16]; //Add round key1

for(int j=0;j<16;j++)

dec\_xor\_res[j] = cipher[j] ^ key2[j];

cout<<"Xor-res is: ";

for(int i=0;i<16;i++)

cout<<dec\_xor\_res[i];

//Final round (Encryption)

int dec\_xor\_res\_fhalf[8]; int dec\_xor\_res\_shalf[8];

for(int i=0;i<16;i++){

if(i<8)

dec\_xor\_res\_fhalf[i] = dec\_xor\_res[i];

else

dec\_xor\_res\_shalf[i-8] = dec\_xor\_res[i];

}

int\* dec\_xor\_res1; int\* dec\_xor\_res2;

int dec\_xor\_res\_combine[16];

dec\_xor\_res1 = subNib(dec\_xor\_res\_fhalf,dec\_sbox);

cout<<" dec\_xor\_res1 is: ";

for(int i=0;i<8;i++){

cout<<dec\_xor\_res1[i];

dec\_xor\_res\_combine[i] = dec\_xor\_res1[i];

}

dec\_xor\_res2 = subNib(dec\_xor\_res\_shalf,dec\_sbox);

for(int i=0;i<8;i++){

cout<<dec\_xor\_res2[i];

dec\_xor\_res\_combine[i+8] = dec\_xor\_res2[i];

}

//swap 2 and 4 nibble

int dec\_xor\_res\_combine\_swap[16];

for(int i=0;i<16;i++){

if(i<4)

dec\_xor\_res\_combine\_swap[i] = dec\_xor\_res\_combine[i];

else if(i>3 && i<8)

dec\_xor\_res\_combine\_swap[i] = dec\_xor\_res\_combine[i+8];

else if(i>7 && i<12)

dec\_xor\_res\_combine\_swap[i] = dec\_xor\_res\_combine[i];

else

dec\_xor\_res\_combine\_swap[i] = dec\_xor\_res\_combine[i-8];

}

cout<<endl<<"dec\_xor\_res\_combine\_swap: ";

for(int i=0;i<16;i++)

cout<<dec\_xor\_res\_combine\_swap[i];

int dec\_round1[16]; //Add round key1

for(int j=0;j<16;j++)

dec\_round1[j] = dec\_xor\_res\_combine\_swap[j] ^ key1[j];

cout<<endl<<" Decryption after round1 is: ";

for(int k=0;k<16;k++)

cout<<dec\_round1[k];

int\* dec\_mat\_res;

dec\_mat\_res = MixColumn(dec\_round1,GF\_mul\_2,GF\_mul\_9);

cout<<endl<<"Result of matrix multiplication: ";

for(int i=0;i<16;i++)

cout<<dec\_mat\_res[i];

///////////////////////////////

//swap 2 and 4 nibble

int dec\_mat\_res\_swap[16];

for(int i=0;i<16;i++){

if(i<4)

dec\_mat\_res\_swap[i] = dec\_mat\_res[i];

else if(i>3 && i<8)

dec\_mat\_res\_swap[i] = dec\_mat\_res[i+8];

else if(i>7 && i<12)

dec\_mat\_res\_swap[i] = dec\_mat\_res[i];

else

dec\_mat\_res\_swap[i] = dec\_mat\_res[i-8];

}

cout<<endl<<"dec\_mat\_res\_swap: ";

for(int i=0;i<16;i++)

cout<<dec\_mat\_res\_swap[i];

//Inverse Nibble sub

int dec\_mat\_res\_fhalf[8]; int dec\_mat\_res\_shalf[8];

for(int i=0;i<16;i++){

if(i<8)

dec\_mat\_res\_fhalf[i] = dec\_mat\_res\_swap[i];

else

dec\_mat\_res\_shalf[i-8] = dec\_mat\_res\_swap[i];

}

int\* dec\_mat\_res1; int\* dec\_mat\_res2;

int dec\_mat\_res\_combine[16];

dec\_mat\_res1 = subNib(dec\_mat\_res\_fhalf,dec\_sbox);

cout<<" dec\_mat\_res1 is: ";

for(int i=0;i<8;i++){

cout<<dec\_mat\_res1[i];

dec\_mat\_res\_combine[i] = dec\_mat\_res1[i];

}

dec\_mat\_res2 = subNib(dec\_mat\_res\_shalf,dec\_sbox);

cout<<" dec\_mat\_res1 is: ";

for(int i=0;i<8;i++){

cout<<dec\_mat\_res2[i];

dec\_mat\_res\_combine[i+8] = dec\_mat\_res2[i];

}

int decrpt\_txt[16]; //Add round key0

for(int j=0;j<16;j++)

decrpt\_txt[j] = (dec\_mat\_res\_combine[j] ^ key[j])-48;

cout<<endl<<"Decryption result is: ";

for(int k=0;k<16;k++)

cout<<decrpt\_txt[k];

cout<<endl;

return 0;

}

/\*Output:

akshay@akshay-pc:~$ g++ SAES.cpp

akshay@akshay-pc:~$ ./a.out

Enter the Plain text(16-bits): 1101011100101000

Enter the key(16-bits): 0100101011110101

W0 is: 01001010

W1 is: 11110101

IN rotNib is(temp1): 01011111

IN subNib is(temp2): 00010111

IN SubNib is(temp2): 00010111

W2 is: 11011101

W3 is: 00101000

IN rotNib is(temp1): 10000010

IN subNib is(temp2): 01101010

IN SubNib is(temp2): 01101010

W4 is: 10000111

W5 is: 10101111

Key1 is: 1101110100101000

Key2 is: 1000011110101111 addRoundKey res: 1001110111011101

IN subNib is(temp2): 00101110 div1\_res is: 00101110

IN subNib is(temp2): 11101110

div\_res\_combine\_swap: 0010111011101110

S00 is: 1111 s10 is: 0110 s01 is: 0011 s11 is: 0011

S\_arr1 is: 1111011000110011

IN subNib is(temp2): 10100011 xor\_res1 is: 10100011

IN subNib is(temp2): 0100001101000011

xor\_res\_combine\_swap: 1010001101000011

Cipher Text is: 0010010011101100Xor-res is: 1010001101000011

IN subNib is(temp2): 00101011 dec\_xor\_res1 is: 00101011

IN subNib is(temp2): 0001101100011011

dec\_xor\_res\_combine\_swap: 0010101100011011

Decryption after round1 is: 1111011000110011

S00 is: 0010 s10 is: 1110 s01 is: 1110 s11 is: 1110

S\_arr1 is: 0010111011101110

Result of matrix multiplication: 0010111011101110

dec\_mat\_res\_swap: 0010111011101110

IN subNib is(temp2): 10011101 dec\_mat\_res1 is: 10011101

IN subNib is(temp2): 11011101 dec\_mat\_res1 is: 11011101

Decryption result is: 1101011100101000

\*/