#include <bits/stdc++.h>

using namespace std;

unsigned long long sbox[16][16] = {

{ 0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab,

0x76 },

{ 0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72,

0xc0 },

{ 0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31,

0x15 },

{ 0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2,

0x75 },

{ 0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f,

0x84 },

{ 0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58,

0xcf },

{ 0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f,

0xa8 },

{ 0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3,

0xd2 },

{ 0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19,

0x73 },

{ 0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b,

0xdb },

{ 0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4,

0x79 },

{ 0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae,

0x08 },

{ 0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b,

0x8a },

{ 0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d,

0x9e },

{ 0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28,

0xdf },

{ 0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb,

0x16 }

};

unsigned long long Rcon[11] = {

0x00000000, 0x01000000, 0x02000000, 0x04000000,

0x08000000, 0x10000000, 0x20000000, 0x40000000,

0x80000000, 0x1b000000, 0x36000000

};string w[44];

string rotLeft(string word){

return word.substr(8) + word.substr(0,8);

}

string SBoxFun(string word){

string res = "";

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

string byte = word.substr(i\*8, 8);

int row = bitset<4>( byte.substr(0,4) ).to\_ulong();

int col = bitset<4>( byte.substr(4,4) ).to\_ulong();

res += bitset<8>(sbox[row][col]).to\_string();

}

return res;

}

string XOR(string x, string y){

string res = "";

for(int i=0; i<x.length(); i++)

{

res += (x[i] == y[i]) ? "0" : "1";

}

return res;

}

int main()

{

unsigned long long hexkey1, hexkey2;

cout << "\nEnter first 64-bit key in hexadecimal(16-digits) : " ;

cin >> hex >> hexkey1;

cout << "\nEnter next 64-bit key in hexadecimal(16-digits) : " ;

cin >> hex >> hexkey2;

string key = bitset<64>(hexkey1).to\_string() + bitset<64>(hexkey2).to\_string();

cout << "Binary key (k) \t: " << key << endl;

cout << "keyLen : " << key.length() << endl;

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

{

w[i] = key.substr(i\*32,32);

// cout << i+1 << " " << w[i] << endl;

}

for(int i=4; i<44; i++)

{

string first = w[i-4];

string second = w[i-1];

if(i % 4 == 0)

{

second = rotLeft(second);

second = SBoxFun(second);

string tmp = bitset<32>(Rcon[i/4]).to\_string();

second = XOR(second, tmp);

}

w[i] = XOR(first, second);

// cout << i+1 << " " << w[i] << endl;

}

string keys[11] = {""};

for(int i=0; i<44; i++)

{

keys[i/4] += w[i];

}

for(int i=0; i<11; i++)

{

cout << "Key " << i << ": "<< endl;

string binary\_string1,binary\_string2;

for(int j=0;j<64;j++)

{

binary\_string1+=keys[i][j];

binary\_string2+=keys[i][j+64];

}

bitset<64> set1(binary\_string1);

bitset<64> set2(binary\_string2);

cout<<hex<<set1.to\_ulong()<<" "<<set2.to\_ulong()<<endl;

}

return 0;

}

#include <bits/stdc++.h>

using namespace std;

unsigned long long sbox[16][16] = {

{ 0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe, 0xd7, 0xab, 0x76 },

{ 0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0 },

{ 0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1, 0x71, 0xd8, 0x31, 0x15 },

{ 0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75 },

{ 0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84 },

{ 0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf },

{ 0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8 },

{ 0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2 },

{ 0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73 },

{ 0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb },

{ 0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79 },

{ 0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08 },

{ 0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a },

{ 0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e },

{ 0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf },

{ 0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16 }

};

unsigned long long key[4][4] = {

{0x54,0x53,0x50,0x31},

{0x45,0x43,0x49,0x32},

{0x41,0x4f,0x41,0x33},

{0x4d,0x52,0x4e,0x34}

};

string XOR(string x, string y){

string res = "";

for(int i=0; i<x.length(); i++)

{

res += (x[i] == y[i]) ? "0" : "1";

}

return res;

}

string SBoxFun(string byte){

string res = "";

int row = bitset<4>( byte.substr(0,4) ).to\_ulong();

int col = bitset<4>( byte.substr(4,4) ).to\_ulong();

res = bitset<8>(sbox[row][col]).to\_string();

return res;

}

int main(){

string msg;

cout << "Enter message: ";

cin >>msg;

string hexMsg="";

stringstream sstream;

unsigned long long x;

for(int i=0; msg[i]!='\0';i++){

int ascii = msg[i];

sstream.str("");

sstream << hex<<ascii;

hexMsg += sstream.str();

}

string mat[4][4] , initTrans[4][4], res[4][4], res1[4][4];

int k=0;

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

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

mat[j][i] = hexMsg.substr(i\*8+j\*2,2);

}

}

cout << "\nInitial Matrix:\n";

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

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

cout << mat[i][j] <<" ";

}

cout << endl;

}

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

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

unsigned long long val = stoull(mat[i][j], nullptr, 16);

string temp1 = bitset<8>(val).to\_string();

string temp2 = bitset<8>(key[i][j]).to\_string();

//cout << temp1 << " " << temp2<<endl;

initTrans[i][j] = XOR(temp1,temp2);

}

}

cout << "\nInitial Transposition Matrix:\n";

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

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

cout << hex<< bitset<8>(initTrans[i][j]).to\_ulong() <<" ";

}

cout << endl;

}

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

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

res[i][j] = SBoxFun(initTrans[i][j]);

}

}

cout << "\nSubstituted Matrix:\n";

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

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

cout << hex<< bitset<8>(res[i][j]).to\_ulong() <<" ";

}

cout << endl;

}

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

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

res1[i][j] = res[i][(j+i)%4];

}

}

cout << "\nShiftRow Transformation:\n";

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

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

cout << hex<< bitset<8>(res1[i][j]).to\_ulong()<<" ";

}

cout << endl;

}

return 0;

}