# Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology, Pune-37

*(An autonomous Institute of Savitribai Phule Pune University)*

**

# Department of Computer Engineering

|  |  |
| --- | --- |
| **Name** | Thakare Prathamesh Prabhakar |
| **Batch** | Batch - 2 |
| **PRN** | 12220016 |
| **Roll No** | 58 |
| **Division** | CS-D |
| **Subject** | Cyber Security |

**Problem Statement:** Implement Simplified DES Algorithm.

**Code:**

import java.util.\*;

public class DESAlgorithm {

public static void main(String[] args) {

int[] PT = {0,0,1,1,0,1,1,0},

K = {0,0,1,0,0,1,0,1,1,1},

P10 = {3,5,2,7,4,10,1,9,8,6},

P8 = {6,3,7,4,8,5,10,9},

P4 = {2,4,3,1},

EP = {4,1,2,3,2,3,4,1},

IP = {2,6,3,1,4,8,5,7},

IPinverse = {4,1,3,5,7,2,8,6}; //IP inverse

int[][] S0 = {{1,0,3,2},{3,2,1,0},{0,2,1,3},{3,1,3,2}},

S1 = {{0,1,2,3},{2,0,1,3},{3,0,1,0},{2,1,0,3}};

System.out.println(Arrays.toString(PT) + " = Plain text");

int[] p10=convert(P10, K);

System.out.println(Arrays.toString(p10) + " = P10");

//LS-1

LS1\_5(p10);

LS6\_10(p10);

System.out.println(Arrays.toString(p10) + " = P10");

int[] K1 = convert(P8, p10); //KEY-1

System.out.println(Arrays.toString(K1) + " = Key1");

//LS-2

LS1\_5(p10); LS1\_5(p10);

LS6\_10(p10); LS6\_10(p10);

System.out.println(Arrays.toString(p10) + " = P10");

int[] K2 = convert(P8, p10); //KEY-2

System.out.println(Arrays.toString(K2) + " = Key2");

int[] ip = convert(IP,PT);

System.out.println(Arrays.toString(ip) + " = IP"); //IP

int[] ep = convert(EP,Arrays.copyOfRange(ip, 4, 9));

System.out.println(Arrays.toString(ep) + " = EP"); //EP

int[] xor = XOR(ep,K1);

System.out.println(Arrays.toString(xor) + " = XOR");

int[] s0s1 = S0S1(xor,S0,S1);

System.out.println(Arrays.toString(s0s1) + " = S0S1");

int[] p4 = convert(P4, s0s1);

System.out.println(Arrays.toString(p4) + " = P4");

int[] xor1 = XOR(p4, Arrays.copyOf(ip, 4));

System.out.println(Arrays.toString(xor1) + " = XOR(P4xIP:0-4)");

Swap(xor1, ip); //SWAP

System.out.println(Arrays.toString(ip) + " = NewIP");

ep = convert(EP,Arrays.copyOfRange(ip, 4, 9));

System.out.println(Arrays.toString(ep) + " = NewEP"); //EP

xor = XOR(ep,K2);

System.out.println(Arrays.toString(xor) + " = XOR");

s0s1 = S0S1(xor,S0,S1);

System.out.println(Arrays.toString(s0s1) + " = S0S1");

p4 = convert(P4, s0s1);

System.out.println(Arrays.toString(p4) + " = NewP4");

xor1 = XOR(p4, Arrays.copyOf(ip, 4));

System.out.println(Arrays.toString(xor1) + " = XOR(P4xIP:0-4)");

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

ip[i]=xor1[i];

}

int[] C =convert(IPinverse, ip);

System.out.print("\n\nCipher Text: ");

for(int i=0;i<C.length;i++){System.out.print(C[i]);}

//decryption

int[] ip\_dec = convert(IP,C);

System.out.println("\n"+Arrays.toString(ip\_dec) + " = IP"); //IP

int[] ep\_dec = convert(EP,Arrays.copyOfRange(ip\_dec, 4, 9));

System.out.println(Arrays.toString(ep\_dec) + " = EP"); //EP

int[] xor\_dec = XOR(ep\_dec,K2);

System.out.println(Arrays.toString(xor\_dec) + " = XOR");

int[] s0s1\_dec= S0S1(xor\_dec,S0,S1);

System.out.println(Arrays.toString(s0s1\_dec) + " = S0S1");

int[] p4\_dec = convert(P4, s0s1\_dec);

System.out.println(Arrays.toString(p4\_dec) + " = P4");

int[] xor1\_dec = XOR(p4, Arrays.copyOf(ip\_dec, 4));

System.out.println(Arrays.toString(xor1\_dec) + " = XOR(P4xIP:0-4)");

Swap(xor1\_dec, ip\_dec); //SWAP

System.out.println(Arrays.toString(ip\_dec) + " = NewIP");

ep\_dec = convert(EP,Arrays.copyOfRange(ip\_dec, 4, 9));

System.out.println(Arrays.toString(ep\_dec) + " = NewEP"); //EP

xor\_dec = XOR(ep\_dec,K1);

System.out.println(Arrays.toString(xor\_dec) + " = XOR");

s0s1\_dec = S0S1(xor\_dec,S0,S1);

System.out.println(Arrays.toString(s0s1\_dec) + " = S0S1");

p4\_dec = convert(P4, s0s1\_dec);

System.out.println(Arrays.toString(p4\_dec) + " = NewP4");

xor1\_dec = XOR(p4\_dec, Arrays.copyOf(ip\_dec, 4));

System.out.println(Arrays.toString(xor1\_dec) + " = XOR(P4xIP:0-4)");

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

ip\_dec[i]=xor1\_dec[i];

}

int[] P =convert(IPinverse, ip\_dec);

System.out.print("\n\nPlain Text: ");

for(int i=0;i<P.length;i++){System.out.print(P[i]);}

}

static void Swap(int[] xor, int[] ip) {

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

ip[i]=ip[i+4];

ip[i+4]=xor[i];

}

}

static int[] S0S1(int[] xor, int[][] s0, int[][] s1) {

int[] ans = new int[4];

String sr0 = xor[0]+""+xor[3],

sc0 = xor[1]+""+xor[2],

sr1 = xor[4]+""+xor[7],

sc1 = xor[5]+""+xor[6];

int x = s0[Integer.parseInt(sr0,2)][Integer.parseInt(sc0,2)],

y = s1[Integer.parseInt(sr1,2)][Integer.parseInt(sc1,2)];

if(x==0) {

ans[0]=0; ans[1]=0;

}

else if (x==1) {

ans[0]=0; ans[1]=1;

}

else if (x==2) {

ans[0]=1; ans[1]=0;

}

else{

ans[0]=1; ans[1]=1;

}

if(y==0) {

ans[2]=0; ans[3]=0;

}

else if (y==1) {

ans[2]=0; ans[3]=1;

}

else if (y==2) {

ans[2]=1; ans[3]=0;

}

else if (y==3) {

ans[2]=1; ans[3]=1;

}

return ans;

}

static int[] XOR(int[] ep, int[] k) {

int[] ans = new int[ep.length];

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

ans[i]= (ep[i]==k[i]) ? 0:1;

}

return ans;

}

static void LS6\_10(int[] p10) {

int temp=p10[5];

for(int i=5; i<9; i++) {

p10[i]=p10[i+1];

}

p10[9]=temp;

}

static void LS1\_5(int[] p10) {

int temp=p10[0];

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

p10[i]=p10[i+1];

}

p10[4]=temp;

}

static int[] convert(int[] p, int[] k)

{

int[] ans = new int[p.length];

int i=0;

for (int x : p) {

ans[i++] = k[x-1];

}

return ans;

}

}

**Output:**

