#include <iostream>

#include <stdio.h>

using namespace std;

unsigned short packes[16] = { 0x45, 0x27, 0x16, 0x0,

0x62, 0x53, 0x4e, 0x31,

0x2c, 0x1d, 0x74, 0x58,

0x3a, 0x69, 0x7f, 0x00 };

unsigned short vec\_err[7] = {0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01};

unsigned short polinom = 0x58;

unsigned short s1[4]={0x40,0x20,0x10,0x04};

unsigned short s2[4]={0x20,0x10,0x08,0x02};

unsigned short s3[4]={0x40,0x20,0x08,0x01};

unsigned short res\_s[3]={0,0,0};

unsigned short i, k, l, m, n;

unsigned short packet\_err,result;

void computing\_S1()

{ for (m=0; m<4; m++)

{result=packet\_err&s1[m];

if (result!=0)

{res\_s[l]=res\_s[l]^1;

if (m==3) cout<<"1=";

else

cout<<"1+";

} else

if (m==3) cout<<"0=";

else

cout<<"0+";

}

cout<<res\_s[l];

return;

}

void computing\_S2()

{ for (m=0; m<4; m++)

{result=packet\_err&s2[m];

if (result!=0)

{res\_s[l]=res\_s[l]^1;

if (m==3) cout<<"1=";

else

cout<<"1+";

} else

if (m==3) cout<<"0=";

else

cout<<"0+";

}

cout<<res\_s[l];

return;

}

void computing\_S3()

{ for (m=0; m<4; m++)

{result=packet\_err&s3[m];

if (result!=0)

{res\_s[l]=res\_s[l]^1;

if (m==3) cout<<"1=";

else

cout<<"1+";

} else

if (m==3) cout<<"0=";

else

cout<<"0+";

}

cout<<res\_s[l];

return;

}

void computing\_sindrom()

{ for(l = 0; l<3; l++)

switch (l)

{case

0:cout<<"S1=";computing\_S1();cout<<"\n";break;

case

1:cout<<"S2=";computing\_S2();cout<<"\n";break;

case

2:cout<<"S3=";computing\_S3();cout<<"\n";break;

}

return;

}

extern void output\_screen(unsigned short);

int main(int argc, char \*\*argv)

{

cout <<"Input position number of error by code, please...\n";

cin >> i;

cout<<"\n";

cout << "Input packet number, please...\n";

cin >> k;

cout<<"\n";

packet\_err= packets[k] ^ vec\_err[i];

cout << "Packet\_err(1,0)=";

output\_screen (packet\_err);

cout<<"\n";cout<<"\n";

// Computing sindrom

computing\_sindrom();

cout<<"\n";

cout<<"Sindrom has computed! \n";

cout<<"\n";

cout<<"S1S2S3=";

for (i=0;i<3;i++) cout<<res\_s[i];

cout<<"\n";

return(0);

}

void output\_screen(unsiged short var)

{ unsigned short var\_1;

var\_1=var;

for (n = 0; n<7; n++)

{

var = var&vec\_err[n];

if (var !=0) cout << "1";

else cout <<"0";

var=var\_1;

}

return;

}