REVERSIBLE AND QUANTUM COMPUTING ASSIGNMENT

-ANUSH RAJAGOPALAN, 205002012

1. Consider 2bit reversible circuit. We can have 24 permutations (P1 to P24). Any two such permutations are taken and multiplied. Write a program to identify the result among P1 to P24.

CODE:

#include<bits/stdc++.h>

using namespace std;

struct datr{

int inp[2][4];

};

int main()

{

datr p[25];

int j=0,t=0;

int m=-1;

int a[] = { 1, 2, 3, 4 },b[4];

int n = sizeof(a) / sizeof(a[0]);

for(int i=1;i<=24;i++)

{ if(i<10){

p[i].inp[0][0]=0;

p[i].inp[0][1]=i;}

else{

j=i%10;

t=i/10;

p[i].inp[0][0]=t;

p[i].inp[0][1]=j;

}

if(i==1)

{

sort(a, a + n);

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

{ p[i].inp[1][k]=a[k];}

}

else{

if(next\_permutation(a, a + n))

{

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

{ p[i].inp[1][k]=a[k];}

}

}

}

cout<<"24 different combinations"<<endl;

for(int i=1;i<=24;i++)

{

cout<<p[i].inp[0][0]<<p[i].inp[0][1];

cout<<"-";

for(int t=0;t<4;t++){ cout<<p[i].inp[1][t]<<" ";}

cout<<endl;

}

int ch,num1,num2,temp,l;

cout<<"\n Enter numbers:";

cout<<"\n Number 1:";

cin>>num1;

cout<<"\n Number 2:";

cin>>num2;

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

temp=p[num1].inp[1][l];

b[l]=p[num2].inp[1][temp-1];}

int x=((b[0]-1)\*6) +1;

for(int t=x;t<=b[0]\*6;t++)

{ //cout<<"inside";

if(b[1]==p[t].inp[1][1])

{

if(b[2]==p[t].inp[1][2] && b[3]==p[t].inp[1][3])

{ cout<<"Resultant: p"<<t<<":"; break;}

}

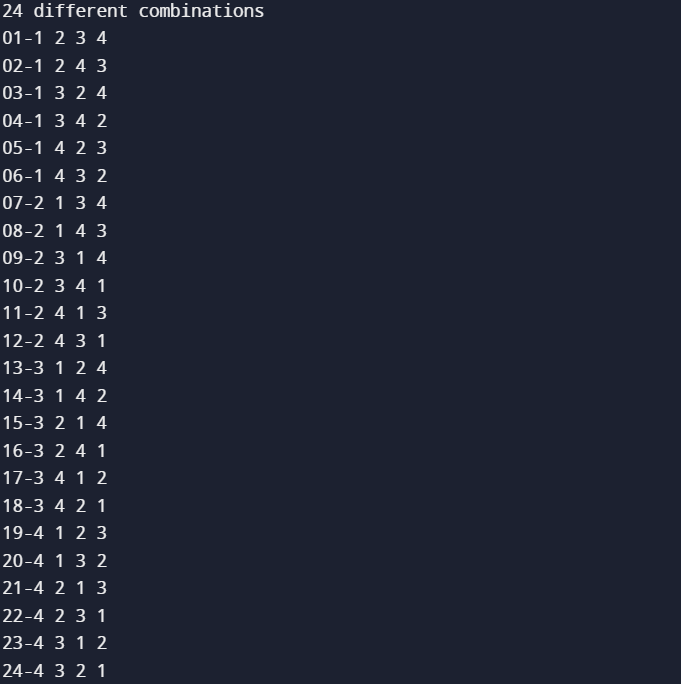
}

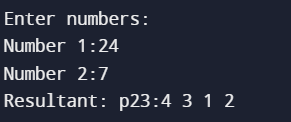
for(l=0;l<4;l++)

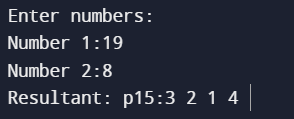
{ cout<<b[l]<<" ";}

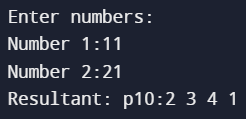
return 0;

}



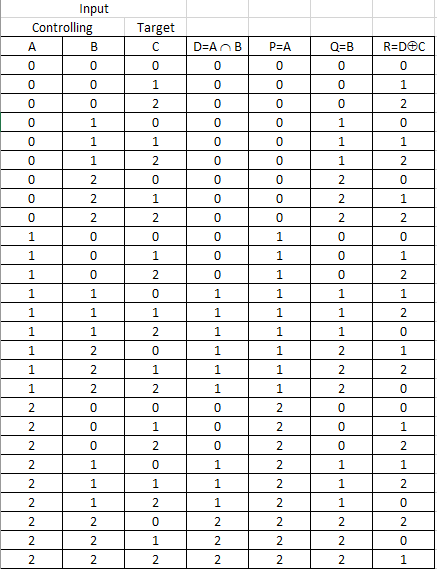






1. A binary x-or gate is reversible, if one of the inputs is made available at the output. What about Mod 3 gate? Do we have this property?

Assuming exor as mod 3 gate



Assuming AB’ +A’B as true in ternary realm and we define negation as

0->2

1->1

2->0

