**A simple CAD tool based on K -map**

E24082117 傅約榮

操作手冊:

1. 在input.txt(和程式同資料夾)輸入:

1. <number> Variable number, Range: 8~10

2. [index, index…] Minterm value index, Range: 0~(2N-1), N = Variable number, (The existed index stands for value 1)

3. (index, index…) Don’t care index, Range: 0~(2N-1), N = Variable number, (The existed index stands for value X)

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

1. 在資料夾中開啟cmd

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

3-執行.java/.class(輸入 java simulator.java)

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

(註:上面的文字是正常現象，不需要理它)

4-查看output.txt,就有結果了

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

程式撰寫流程:

1.先將程式的input,output寫完(創好)

2.接著是用.indexOf判斷程式為8,9或10 variable

3.將所有從input.txt得到的數字(value=1&X)存起來，同時將得到的數字(value=1&X) 轉換成二進位也存起來(存進二維字元陣列中)

4.比較兩個(組)數字之二進位有幾位數是一樣的，若兩個數字之二進位只有一位數是不同的的，則可組成一組兩個數字的prime implicant

5. 得出新的prime implicant(用二進位表示),並存入另一個二維字元陣列中

6. 將原本存的數字中被新的prime implicant給涵蓋到的數字刪除

7. 輸出第一排(column 1)

8. 重複4~7

9. 將(value=1)的數字存入新的二維字元陣列中,將全部不重複的prime implicant所指向的數字依次數加總，例如:有兩個 prime implicant有涵蓋"128",則將2存入陣列中

10.將所有的 prime implicant進行判斷,將一個prime implicant刪去,若沒有任何value 1的數被遺漏則該prime implicant為多餘的,應被刪除,進行完這個過程後,會得出最後的essential prime implicant

11. 輸出最終的essential prime implicant及標準格式的圖表

12.debug

作業心得

這次的作業難度比上次算是簡單了許多，我使用了最熟的java語言去寫；我的程式碼大部分都只是同一件事做很多次而已，所以實際花的時間算是不太多，小小的幾行程式碼就能處理1024個數著實令人驚異，也感謝助教的回答及老師上課的講解，我才能順利寫完。

Source code:

import java.util.\*;

import java.util.ArrayList;

import java.io.\*;

public class simulator{

public static void main(String args[]){

Scanner input=new Scanner(System.in);

try{

Scanner inputstream=new Scanner(new FileInputStream("input.txt"));

PrintWriter Writer = new PrintWriter(new FileOutputStream("output.txt"));

//====================================================================================//

int totalVariablesCount=0;

String trans="0";

int result=0;

String bit="0";

int a=0;

int b=0;

int[] numbers=new int[1024];

char[][] qc=new char[1024][10];

char[] used1=new char[1024];

int x=1;

int y=1;

String number= inputstream.nextLine();

if(number.indexOf("8")!=-1){

totalVariablesCount=8;

}else if(number.indexOf("9")!=-1){

totalVariablesCount=9;

}else{

totalVariablesCount=10;}

//確認variable有幾個(8~10)

String t1=inputstream.nextLine();

if(t1.equals("[")){

System.out.println("[");

}

do{

trans=inputstream.nextLine();

if(trans.equals("]")){

x=0;

}else if (trans.matches("^[0-9]+$")){

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

qc[a][j]='0';

}

result=Integer.parseInt(trans);

numbers[a]=result;//將取得的數字(value=1)本身存入numbers[]

bit=Integer.toBinaryString(result);

//System.out.println(bit);

for(int i=bit.length()-1;i>=0;i--){

qc[a][bit.length()-1-i] = bit.charAt(i);//將取得的數字(value=1)轉換成二進位存入qc[][]

}

a++; }

}while(x==1);//取得value為1的數字("[]"之間)

String t2=inputstream.nextLine();

if(t2.equals("(")){

System.out.println("(");

}

do{

trans=inputstream.nextLine();

if(trans.equals(")")){

y=0;

}else if (trans.matches("^[0-9]+$")){

for(int k=0;k<totalVariablesCount;k++){

qc[a+b][k]='0';

}

result=Integer.parseInt(trans);

numbers[a+b]=result;//將取得的數字(value=X)本身存入numbers[]

bit=Integer.toBinaryString(result);

used1[a+b]='d';

//System.out.println(bit);

for(int l=bit.length()-1;l>=0;l--){

qc[a+b][bit.length()-1-l] = bit.charAt(l);//將取得的數字(value=X)轉換成二進位存入qc[][]

}

b++; }

}while(y==1);//取得value為X的數字("()"之間)

/\*for(int xx=0;xx<a+b;xx++){

//System.out.print(numbers[xx]+":");

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc[xx][yy]+ " ");

}

System.out.println("");

} //確認是否有存入數字\*/

//====================================//

//=============c1~2===================//

int [][] c2=new int[1024][2];//紀錄column 2不重複的prime implicant的數字

int d2=0;//紀錄column 2不重複的有幾項

char[][] qc2=new char[1024][10];//紀錄column 2不重複的prime implicant的二進位

char[] used2=new char[1024];//紀錄column 2中的prime implicant是否有被更大的給涵蓋到

ArrayList delete2=new ArrayList();

for(int m=0;m<a+b;m++){

for(int n=0;n<a+b;n++){

int count=0;

for(int o=0;o<totalVariablesCount;o++){//比較兩個數字之二進位有幾位數是一樣的

if(qc[m][o]==qc[n][o]&&m!=n){

count++;

}

}

if(count==totalVariablesCount-1){//若兩個數字之二進位只有一位數是不同的的，則可組成一組兩個數字的prime implicant

if(numbers[n]>numbers[m]){//削去重複的

c2[d2][0]=numbers[m];

c2[d2][1]=numbers[n];

for(int p=0;p<totalVariablesCount;p++){

if(qc[m][p]==qc[n][p]){//得出新的prime implicant(用二進位表示),並存入qc2[][]

qc2[d2][p]=qc[m][p];

}else{

qc2[d2][p]='-';

}

}

delete2.add(m);

delete2.add(n);

d2++;

}

}

}

}

/\*for(int xx=0;xx<d2;xx++){

//System.out.print(numbers[xx]+":");

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc2[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d2;clk++){

System.out.println(c2[clk][0]+"+"+c2[clk][1]);}\*/

for(int e2=0;e2<delete2.size();e2++){//將 column 1中被新的prime implicant給涵蓋到的刪除

used1[(int)delete2.get(e2)]='X';

}

//輸出column 1

if(a+b!=0){

Writer.println("===========================");

Writer.println("Column 1");

Writer.println("===========================");

for(int z1z=0;z1z<=totalVariablesCount;z1z++){

int barrier1=0;

for(int x1x=0;x1x<a+b;x1x++){

int one1=0;

for(int y1y=totalVariablesCount-1;y1y>=0;y1y--){

if(qc[x1x][y1y]=='1'){

one1++;

}

}

if(one1==z1z){

if(used1[x1x]=='X'){

Writer.print("v ");

}else if(used1[x1x]=='d'){

Writer.print("d ");

}else{

Writer.print(" ");

}

for(int ss1=totalVariablesCount-1;ss1>=0;ss1--){

Writer.print(qc[x1x][ss1]);

}

Writer.println(" : "+numbers[x1x]);

barrier1++;

}

}

if(barrier1>0){

Writer.println("---------------------------");

}

}

}

//=============c2~3===================//

int [][] c3=new int[1024][4];//紀錄column 3不重複的prime implicant的數字

char[][] qc3=new char[1024][10];//紀錄column 3不重複的prime implicant的二進位

int d3=0;//紀錄column 3不重複的有幾項

int v3=0;

int w3=0;

char[] used3=new char[1024];//紀錄column 3中的prime implicant是否有被更大的給涵蓋到

ArrayList delete3=new ArrayList();

for(int m3=0;m3<d2;m3++){

for(int n3=0;n3<d2;n3++){

int count3=0;

for(int o3=0;o3<totalVariablesCount;o3++){

if(qc2[m3][o3]==qc2[n3][o3]&&m3!=n3){

count3++;

}

}

if(count3==totalVariablesCount-1){

v3=0; w3=0;

for(int s3=0;s3<4;s3++){//將column 3的prime implicant的數字存入c3[][]

if(v3>=2&&w3==0){

v3=0;

w3=1;}

if(w3==0){

c3[d3][s3]=c2[m3][v3];

v3++;

}else{

c3[d3][s3]=c2[n3][v3];

v3++;

}

}

//在排序好的陣列中增加一個數字，增加後的數字插入到陣列合適的位置(c3[][]數字排序)

for (int x3=1;x3<4;x3++) //判斷迴圈

{

for (int y3=0;y3<4;y3++)

{

if (c3[d3][x3]<c3[d3][y3]) //數字從小到大排列

{

int temp3 =c3[d3][x3];

c3[d3][x3]=c3[d3][y3];

c3[d3][y3]= temp3;

}

}

}

for(int p3=0;p3<totalVariablesCount;p3++){//得出新的prime implicant(用二進位表示),並存入qc3[][]

if(qc2[m3][p3]==qc2[n3][p3]){

qc3[d3][p3]=qc2[m3][p3];

}else{

qc3[d3][p3]='-';

}

}

for(int q3=0;q3<d3;q3++){

int counter3=0;

for(int t3=0;t3<totalVariablesCount;t3++){//比較兩組數字之二進位有幾位數是一樣的

if(qc3[q3][t3]==qc3[d3][t3]){

counter3++;

}

if(counter3==totalVariablesCount){//若兩組數字之二進位只有一位數是不同的的，則可組成四個數字的prime implicant

d3--;

}

}

}

delete3.add(m3);

delete3.add(n3);

d3++;

}

}

}

/\*for(int xx=0;xx<d3;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc3[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d3;clk++){

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

System.out.println(c3[clk][ck]);}

}\*/

for(int e3=0;e3<delete3.size();e3++){//將 column 2中被新的prime implicant給涵蓋到的刪除

used2[(int)delete3.get(e3)]='X';

}

//輸出column 2

if(d2!=0){

Writer.println("===========================");

Writer.println("Column 2");

Writer.println("===========================");

for(int z2z=0;z2z<=totalVariablesCount;z2z++){

int barrier2=0;

for(int x2x=0;x2x<d2;x2x++){

int one2=0;

for(int y2y=totalVariablesCount-1;y2y>=0;y2y--){

if(qc2[x2x][y2y]=='1'){

one2++;

}

}

if(one2==z2z){

if(used2[x2x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss2=totalVariablesCount-1;ss2>=0;ss2--){

Writer.print(qc2[x2x][ss2]);

}

for(int tt2=0;tt2<2;tt2++){

if(tt2==0){

Writer.print(" : "+c2[x2x][tt2]);

}else{

Writer.print(", "+c2[x2x][tt2]);

}

}

Writer.println("");

barrier2++;

}

}

if(barrier2>0){

Writer.println("---------------------------");

}

}

}

//=============c3~4===================//

//(column 4之註解同column 3)

int [][] c4=new int[1024][8];

char[][] qc4=new char[1024][10];

int d4=0;

int v4=0;

int w4=0;

char[] used4=new char[1024];

ArrayList delete4=new ArrayList();

for(int m4=0;m4<d3;m4++){

for(int n4=0;n4<d3;n4++){

int count4=0;

for(int o4=0;o4<totalVariablesCount;o4++){

if(qc3[m4][o4]==qc3[n4][o4]&&m4!=n4){

count4++;

}

}

if(count4==totalVariablesCount-1){

v4=0; w4=0;

for(int s4=0;s4<8;s4++){

if(v4>=4&&w4==0){

v4=0;

w4=1;}

if(w4==0){

c4[d4][s4]=c3[m4][v4];

v4++;

}else{

c4[d4][s4]=c3[n4][v4];

v4++;

}

}

for (int x4=1;x4<8;x4++) //判斷迴圈

{

for (int y4=0;y4<8;y4++)

{

if (c4[d4][x4]<c4[d4][y4]) //數字從小到大排列

{

int temp4 =c4[d4][x4];

c4[d4][x4]=c4[d4][y4];

c4[d4][y4]= temp4;

}

}

}

for(int p4=0;p4<totalVariablesCount;p4++){

if(qc3[m4][p4]==qc3[n4][p4]){

qc4[d4][p4]=qc3[m4][p4];

}else{

qc4[d4][p4]='-';

}

}

for(int q4=0;q4<d4;q4++){

int counter4=0;

for(int t4=0;t4<totalVariablesCount;t4++){

if(qc4[q4][t4]==qc4[d4][t4]){

counter4++;

}

if(counter4==totalVariablesCount){

d4--;

}

}

}

delete4.add(m4);

delete4.add(n4);

d4++;

}

}

}

/\*for(int xx=0;xx<d4;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc4[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d4;clk++){

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

System.out.println(c4[clk][ck]);}

} \*/

for(int e4=0;e4<delete4.size();e4++){

used3[(int)delete4.get(e4)]='X';

}

if(d3!=0){

Writer.println("===========================");

Writer.println("Column 3");

Writer.println("===========================");

for(int z3z=0;z3z<=totalVariablesCount;z3z++){

int barrier3=0;

for(int x3x=0;x3x<d3;x3x++){

int one3=0;

for(int y3y=totalVariablesCount-1;y3y>=0;y3y--){

if(qc3[x3x][y3y]=='1'){

one3++;

}

}

if(one3==z3z){

for(int re3=0;re3<2;re3++){

if(re3==1){

Writer.print("x ");

}else if(used3[x3x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss3=totalVariablesCount-1;ss3>=0;ss3--){

Writer.print(qc3[x3x][ss3]);

}

for(int tt3=0;tt3<4;tt3++){

if(tt3==0){

Writer.print(" : "+c3[x3x][tt3]);

}else{

Writer.print(", "+c3[x3x][tt3]);

}

}

Writer.println("");

barrier3++;

}

}

}

if(barrier3>0){

Writer.println("---------------------------");

}

}

}

//=============c4~5===================//

//(column 5之註解同column 3)

int [][] c5=new int[128][16];

char[][] qc5=new char[1024][10];

int d5=0;

int v5=0;

int w5=0;

char[] used5=new char[128];

ArrayList delete5=new ArrayList();

for(int m5=0;m5<d4;m5++){

for(int n5=0;n5<d4;n5++){

int count5=0;

for(int o5=0;o5<totalVariablesCount;o5++){

if(qc4[m5][o5]==qc4[n5][o5]&&m5!=n5){

count5++;

}

}

if(count5==totalVariablesCount-1){

v5=0; w5=0;

for(int s5=0;s5<16;s5++){

if(v5>=8&&w5==0){

v5=0;

w5=1;}

if(w5==0){

c5[d5][s5]=c4[m5][v5];

v5++;

}else{

c5[d5][s5]=c4[n5][v5];

v5++;

}

}

for (int x5=1;x5<16;x5++) //判斷迴圈

{

for (int y5=0;y5<16;y5++)

{

if (c5[d5][x5]<c5[d5][y5]) //數字從小到大排列

{

int temp5 =c5[d5][x5];

c5[d5][x5]=c5[d5][y5];

c5[d5][y5]= temp5;

}

}

}

for(int p5=0;p5<totalVariablesCount;p5++){

if(qc4[m5][p5]==qc4[n5][p5]){

qc5[d5][p5]=qc4[m5][p5];

}else{

qc5[d5][p5]='-';

}

}

for(int q5=0;q5<d5;q5++){

int counter5=0;

for(int t5=0;t5<totalVariablesCount;t5++){

if(qc5[q5][t5]==qc5[d5][t5]){

counter5++;

}

if(counter5==totalVariablesCount){

d5--;

}

}

}

delete5.add(m5);

delete5.add(n5);

d5++;

}

}

}

/\*for(int xx=0;xx<d5;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc5[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d5;clk++){

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

System.out.println(c5[clk][ck]);}

} \*/

for(int e5=0;e5<delete5.size();e5++){

used4[(int)delete5.get(e5)]='X';

}

if(d4!=0){

Writer.println("===========================");

Writer.println("Column 4");

Writer.println("===========================");

for(int z4z=0;z4z<=totalVariablesCount;z4z++){

int barrier4=0;

for(int x4x=0;x4x<d4;x4x++){

int one4=0;

for(int y4y=totalVariablesCount-1;y4y>=0;y4y--){

if(qc4[x4x][y4y]=='1'){

one4++;

}

}

if(one4==z4z){

for(int re4=0;re4<2;re4++){

if(re4==1){

Writer.print("x ");

}else if(used4[x4x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss4=totalVariablesCount-1;ss4>=0;ss4--){

Writer.print(qc4[x4x][ss4]);

}

for(int tt4=0;tt4<8;tt4++){

if(tt4==0){

Writer.print(" : "+c4[x4x][tt4]);

}else{

Writer.print(", "+c4[x4x][tt4]);

}

}

Writer.println("");

barrier4++;

}

}

}

if(barrier4>0){

Writer.println("---------------------------");

}

}

}

//=============c5~6===================//

//(column 6之註解同column 3)

int [][] c6=new int[64][32];

char[][] qc6=new char[1024][10];

int d6=0;

int v6=0;

int w6=0;

char[] used6=new char[64];

ArrayList delete6=new ArrayList();

for(int m6=0;m6<d5;m6++){

for(int n6=0;n6<d5;n6++){

int count6=0;

for(int o6=0;o6<totalVariablesCount;o6++){

if(qc5[m6][o6]==qc5[n6][o6]&&m6!=n6){

count6++;

}

}

if(count6==totalVariablesCount-1){

v6=0; w6=0;

for(int s6=0;s6<32;s6++){

if(v6>=8&&w6==0){

v6=0;

w6=1;}

if(w6==0){

c6[d6][s6]=c5[m6][v6];

v6++;

}else{

c6[d6][s6]=c5[n6][v6];

v6++;

}

}

for (int x6=1;x6<32;x6++) //判斷迴圈

{

for (int y6=0;y6<32;y6++)

{

if (c6[d6][x6]<c6[d6][y6]) //數字從小到大排列

{

int temp6 =c6[d6][x6];

c6[d6][x6]=c6[d6][y6];

c6[d6][y6]= temp6;

}

}

}

for(int p6=0;p6<totalVariablesCount;p6++){

if(qc5[m6][p6]==qc5[n6][p6]){

qc6[d6][p6]=qc5[m6][p6];

}else{

qc6[d6][p6]='-';

}

}

for(int q6=0;q6<d6;q6++){

int counter6=0;

for(int t6=0;t6<totalVariablesCount;t6++){

if(qc6[q6][t6]==qc6[d6][t6]){

counter6++;

}

if(counter6==totalVariablesCount){

d6--;

}

}

}

delete6.add(m6);

delete6.add(n6);

d6++;

}

}

}

/\*for(int xx=0;xx<d6;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc6[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d6;clk++){

for(int ck=0;ck<32;ck++){

System.out.println(c6[clk][ck]);}

} \*/

for(int e6=0;e6<delete6.size();e6++){

used5[(int)delete6.get(e6)]='X';

}

if(d5!=0){

Writer.println("===========================");

Writer.println("Column 5");

Writer.println("===========================");

for(int z5z=0;z5z<=totalVariablesCount;z5z++){

int barrier5=0;

for(int x5x=0;x5x<d5;x5x++){

int one5=0;

for(int y5y=totalVariablesCount-1;y5y>=0;y5y--){

if(qc5[x5x][y5y]=='1'){

one5++;

}

}

if(one5==z5z){

for(int re5=0;re5<2;re5++){

if(re5==1){

Writer.print("x ");

}else if(used5[x5x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss5=totalVariablesCount-1;ss5>=0;ss5--){

Writer.print(qc5[x5x][ss5]);

}

for(int tt5=0;tt5<16;tt5++){

if(tt5==0){

Writer.print(" : "+c5[x5x][tt5]);

}else{

Writer.print(", "+c5[x5x][tt5]);

}

}

Writer.println("");

barrier5++;

}

}

}

if(barrier5>0){

Writer.println("---------------------------");

}

}

}

//=============c6~7===================//

//(column 7之註解同column 3)

int [][] c7=new int[32][64];

char[][] qc7=new char[1024][10];

int d7=0;

int v7=0;

int w7=0;

char[] used7=new char[32];

ArrayList delete7=new ArrayList();

for(int m7=0;m7<d6;m7++){

for(int n7=0;n7<d6;n7++){

int count7=0;

for(int o7=0;o7<totalVariablesCount;o7++){

if(qc6[m7][o7]==qc6[n7][o7]&&m7!=n7){

count7++;

}

}

if(count7==totalVariablesCount-1){

v7=0; w7=0;

for(int s7=0;s7<64;s7++){

if(v7>=8&&w7==0){

v7=0;

w7=1;}

if(w7==0){

c7[d7][s7]=c6[m7][v7];

v7++;

}else{

c7[d7][s7]=c6[n7][v7];

v7++;

}

}

for (int x7=1;x7<64;x7++) //判斷迴圈

{

for (int y7=0;y7<64;y7++)

{

if (c7[d7][x7]<c7[d7][y7]) //數字從小到大排列

{

int temp7 =c7[d7][x7];

c7[d7][x7]=c7[d7][y7];

c7[d7][y7]= temp7;

}

}

}

for(int p7=0;p7<totalVariablesCount;p7++){

if(qc6[m7][p7]==qc6[n7][p7]){

qc7[d7][p7]=qc6[m7][p7];

}else{

qc7[d7][p7]='-';

}

}

for(int q7=0;q7<d7;q7++){

int counter7=0;

for(int t7=0;t7<totalVariablesCount;t7++){

if(qc7[q7][t7]==qc7[d7][t7]){

counter7++;

}

if(counter7==totalVariablesCount){

d7--;

}

}

}

delete7.add(m7);

delete7.add(n7);

d7++;

}

}

}

/\*for(int xx=0;xx<d7;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc7[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d7;clk++){

for(int ck=0;ck<64;ck++){

System.out.println(c7[clk][ck]);}

} \*/

for(int e7=0;e7<delete7.size();e7++){

used6[(int)delete7.get(e7)]='X';

}

if(d6!=0){

Writer.println("===========================");

Writer.println("Column 6");

Writer.println("===========================");

for(int z6z=0;z6z<=totalVariablesCount;z6z++){

int barrier6=0;

for(int x6x=0;x6x<d6;x6x++){

int one6=0;

for(int y6y=totalVariablesCount-1;y6y>=0;y6y--){

if(qc6[x6x][y6y]=='1'){

one6++;

}

}

if(one6==z6z){

for(int re6=0;re6<2;re6++){

if(re6==1){

Writer.print("x ");

}else if(used6[x6x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss6=totalVariablesCount-1;ss6>=0;ss6--){

Writer.print(qc6[x6x][ss6]);

}

for(int tt6=0;tt6<32;tt6++){

if(tt6==0){

Writer.print(" : "+c6[x6x][tt6]);

}else{

Writer.print(", "+c6[x6x][tt6]);

}

}

Writer.println("");

barrier6++;

}

}

}

if(barrier6>0){

Writer.println("---------------------------");

}

}

}

//=============c7~8===================//

//(column 8之註解同column 3)

int [][] c8=new int[16][128];

char[][] qc8=new char[1024][10];

int d8=0;

int v8=0;

int w8=0;

char[] used8=new char[16];

ArrayList delete8=new ArrayList();

for(int m8=0;m8<d7;m8++){

for(int n8=0;n8<d7;n8++){

int count8=0;

for(int o8=0;o8<totalVariablesCount;o8++){

if(qc7[m8][o8]==qc7[n8][o8]&&m8!=n8){

count8++;

}

}

if(count8==totalVariablesCount-1){

v8=0; w8=0;

for(int s8=0;s8<128;s8++){

if(v8>=8&&w8==0){

v8=0;

w8=1;}

if(w8==0){

c8[d8][s8]=c7[m8][v8];

v8++;

}else{

c8[d8][s8]=c7[n8][v8];

v8++;

}

}

for (int x8=1;x8<128;x8++) //判斷迴圈

{

for (int y8=0;y8<128;y8++)

{

if (c8[d8][x8]<c8[d8][y8]) //數字從小到大排列

{

int temp8 =c8[d8][x8];

c8[d8][x8]=c8[d8][y8];

c8[d8][y8]= temp8;

}

}

}

for(int p8=0;p8<totalVariablesCount;p8++){

if(qc7[m8][p8]==qc7[n8][p8]){

qc8[d8][p8]=qc7[m8][p8];

}else{

qc8[d8][p8]='-';

}

}

for(int q8=0;q8<d8;q8++){

int counter8=0;

for(int t8=0;t8<totalVariablesCount;t8++){

if(qc8[q8][t8]==qc8[d8][t8]){

counter8++;

}

if(counter8==totalVariablesCount){

d8--;

}

}

}

delete8.add(m8);

delete8.add(n8);

d8++;

}

}

}

/\*for(int xx=0;xx<d8;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc8[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d8;clk++){

for(int ck=0;ck<128;ck++){

System.out.println(c8[clk][ck]);}

} \*/

for(int e8=0;e8<delete8.size();e8++){

used7[(int)delete8.get(e8)]='X';

}

if(d7!=0){

Writer.println("===========================");

Writer.println("Column 7");

Writer.println("===========================");

for(int z7z=0;z7z<=totalVariablesCount;z7z++){

int barrier7=0;

for(int x7x=0;x7x<d7;x7x++){

int one7=0;

for(int y7y=totalVariablesCount-1;y7y>=0;y7y--){

if(qc7[x7x][y7y]=='1'){

one7++;

}

}

if(one7==z7z){

for(int re7=0;re7<2;re7++){

if(re7==1){

Writer.print("x ");

}else if(used7[x7x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss7=totalVariablesCount-1;ss7>=0;ss7--){

Writer.print(qc7[x7x][ss7]);

}

for(int tt7=0;tt7<64;tt7++){

if(tt7==0){

Writer.print(" : "+c7[x7x][tt7]);

}else{

Writer.print(", "+c7[x7x][tt7]);

}

}

Writer.println("");

barrier7++;

}

}

}

if(barrier7>0){

Writer.println("---------------------------");

}

}

}

//=============c8~9===================//

//(column 9之註解同column 3)

int [][] c9=new int[8][256];

char[][] qc9=new char[1024][10];

int d9=0;

int v9=0;

int w9=0;

char[] used9=new char[8];

ArrayList delete9=new ArrayList();

for(int m9=0;m9<d8;m9++){

for(int n9=0;n9<d8;n9++){

int count9=0;

for(int o9=0;o9<totalVariablesCount;o9++){

if(qc8[m9][o9]==qc8[n9][o9]&&m9!=n9){

count9++;

}

}

if(count9==totalVariablesCount-1){

v9=0; w9=0;

for(int s9=0;s9<256;s9++){

if(v9>=9&&w9==0){

v9=0;

w9=1;}

if(w9==0){

c9[d9][s9]=c8[m9][v9];

v9++;

}else{

c9[d9][s9]=c8[n9][v9];

v9++;

}

}

for (int x9=1;x9<256;x9++) //判斷迴圈

{

for (int y9=0;y9<256;y9++)

{

if (c9[d9][x9]<c9[d9][y9]) //數字從小到大排列

{

int temp9 =c9[d9][x9];

c9[d9][x9]=c9[d9][y9];

c9[d9][y9]= temp9;

}

}

}

for(int p9=0;p9<totalVariablesCount;p9++){

if(qc8[m9][p9]==qc8[n9][p9]){

qc9[d9][p9]=qc8[m9][p9];

}else{

qc9[d9][p9]='-';

}

}

for(int q9=0;q9<d9;q9++){

int counter9=0;

for(int t9=0;t9<totalVariablesCount;t9++){

if(qc9[q9][t9]==qc9[d9][t9]){

counter9++;

}

if(counter9==totalVariablesCount){

d9--;

}

}

}

delete9.add(m9);

delete9.add(n9);

d9++;

}

}

}

/\*for(int xx=0;xx<d9;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc9[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d9;clk++){

for(int ck=0;ck<256;ck++){

System.out.println(c9[clk][ck]);}

} \*/

for(int e9=0;e9<delete9.size();e9++){

used8[(int)delete9.get(e9)]='X';

}

if(d8!=0){

Writer.println("===========================");

Writer.println("Column 8");

Writer.println("===========================");

for(int z8z=0;z8z<=totalVariablesCount;z8z++){

int barrier8=0;

for(int x8x=0;x8x<d8;x8x++){

int one8=0;

for(int y8y=totalVariablesCount-1;y8y>=0;y8y--){

if(qc8[x8x][y8y]=='1'){

one8++;

}

}

if(one8==z8z){

for(int re8=0;re8<2;re8++){

if(re8==1){

Writer.print("x ");

}else if(used8[x8x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss8=totalVariablesCount-1;ss8>=0;ss8--){

Writer.print(qc8[x8x][ss8]);

}

for(int tt8=0;tt8<128;tt8++){

if(tt8==0){

Writer.print(" : "+c8[x8x][tt8]);

}else{

Writer.print(", "+c8[x8x][tt8]);

}

}

Writer.println("");

barrier8++;

}

}

}

if(barrier8>0){

Writer.println("---------------------------");

}

}

}

//=============c9~10===================//

//(column 10之註解同column 3)

int [][] c10=new int[4][512];

char[][] qc10=new char[1024][10];

int d10=0;

int v10=0;

int w10=0;

char[] used10=new char[4];

ArrayList delete10=new ArrayList();

for(int m10=0;m10<d9;m10++){

for(int n10=0;n10<d9;n10++){

int count10=0;

for(int o10=0;o10<totalVariablesCount;o10++){

if(qc9[m10][o10]==qc9[n10][o10]&&m10!=n10){

count10++;

}

}

if(count10==totalVariablesCount-1){

v10=0; w10=0;

for(int s10=0;s10<512;s10++){

if(v10>=10&&w10==0){

v10=0;

w10=1;}

if(w10==0){

c10[d10][s10]=c9[m10][v10];

v10++;

}else{

c10[d10][s10]=c9[n10][v10];

v10++;

}

}

for (int x10=1;x10<512;x10++) //判斷迴圈

{

for (int y10=0;y10<512;y10++)

{

if (c10[d10][x10]<c10[d10][y10]) //數字從小到大排列

{

int temp10 =c10[d10][x10];

c10[d10][x10]=c10[d10][y10];

c10[d10][y10]= temp10;

}

}

}

for(int p10=0;p10<totalVariablesCount;p10++){

if(qc9[m10][p10]==qc9[n10][p10]){

qc10[d10][p10]=qc9[m10][p10];

}else{

qc10[d10][p10]='-';

}

}

for(int q10=0;q10<d10;q10++){

int counter10=0;

for(int t10=0;t10<totalVariablesCount;t10++){

if(qc10[q10][t10]==qc10[d10][t10]){

counter10++;

}

if(counter10==totalVariablesCount){

d10--;

}

}

}

delete10.add(m10);

delete10.add(n10);

d10++;

}

}

}

/\*for(int xx=0;xx<d10;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc10[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d10;clk++){

for(int ck=0;ck<512;ck++){

System.out.println(c10[clk][ck]);}

} \*/

for(int e10=0;e10<delete10.size();e10++){

used9[(int)delete10.get(e10)]='X';

}

if(d9!=0){

Writer.println("===========================");

Writer.println("Column 9");

Writer.println("===========================");

for(int z9z=0;z9z<=totalVariablesCount;z9z++){

int barrier9=0;

for(int x9x=0;x9x<d9;x9x++){

int one9=0;

for(int y9y=totalVariablesCount-1;y9y>=0;y9y--){

if(qc9[x9x][y9y]=='1'){

one9++;

}

}

if(one9==z9z){

for(int re9=0;re9<2;re9++){

if(re9==1){

Writer.print("x ");

}else if(used9[x9x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss9=totalVariablesCount-1;ss9>=0;ss9--){

Writer.print(qc9[x9x][ss9]);

}

for(int tt9=0;tt9<256;tt9++){

if(tt9==0){

Writer.print(" : "+c9[x9x][tt9]);

}else{

Writer.print(", "+c9[x9x][tt9]);

}

}

Writer.println("");

barrier9++;

}

}

}

if(barrier9>0){

Writer.println("---------------------------");

}

}

}

//=============c10~11===================//

//(column 11之註解同column 3)

int [][] c11=new int[2][1024];

char[][] qc11=new char[1024][10];

int d11=0;

int v11=0;

int w11=0;

char[] used11=new char[2];

ArrayList delete11=new ArrayList();

for(int m11=0;m11<d10;m11++){

for(int n11=0;n11<d10;n11++){

int count11=0;

for(int o11=0;o11<totalVariablesCount;o11++){

if(qc10[m11][o11]==qc10[n11][o11]&&m11!=n11){

count11++;

}

}

if(count11==totalVariablesCount-1){

v11=0; w11=0;

for(int s11=0;s11<1024;s11++){

if(v11>=11&&w11==0){

v11=0;

w11=1;}

if(w11==0){

c11[d11][s11]=c10[m11][v11];

v11++;

}else{

c11[d11][s11]=c10[n11][v11];

v11++;

}

}

for (int x11=1;x11<1024;x11++) //判斷迴圈

{

for (int y11=0;y11<1024;y11++)

{

if (c11[d11][x11]<c11[d11][y11]) //數字從小到大排列

{

int temp11 =c11[d11][x11];

c11[d11][x11]=c11[d11][y11];

c11[d11][y11]= temp11;

}

}

}

for(int p11=0;p11<totalVariablesCount;p11++){

if(qc10[m11][p11]==qc10[n11][p11]){

qc11[d11][p11]=qc10[m11][p11];

}else{

qc11[d11][p11]='-';

}

}

for(int q11=0;q11<d11;q11++){

int counter11=0;

for(int t11=0;t11<totalVariablesCount;t11++){

if(qc11[q11][t11]==qc11[d11][t11]){

counter11++;

}

if(counter11==totalVariablesCount){

d11--;

}

}

}

delete11.add(m11);

delete11.add(n11);

d11++;

}

}

}

/\*for(int xx=0;xx<d11;xx++){

for(int yy=totalVariablesCount-1;yy>=0;yy--){

System.out.print(qc11[xx][yy]+ " ");

}

System.out.println("");

}\*/

/\*for(int clk=0;clk<d11;clk++){

for(int ck=0;ck<1024;ck++){

System.out.println(c11[clk][ck]);}

} \*/

for(int e11=0;e11<delete11.size();e11++){

used10[(int)delete11.get(e11)]='X';

}

if(d10!=0){

Writer.println("===========================");

Writer.println("Column 10");

Writer.println("===========================");

for(int z10z=0;z10z<=totalVariablesCount;z10z++){

int barrier10=0;

for(int x10x=0;x10x<d10;x10x++){

int one10=0;

for(int y10y=totalVariablesCount-1;y10y>=0;y10y--){

if(qc10[x10x][y10y]=='1'){

one10++;

}

}

if(one10==z10z){

for(int re10=0;re10<2;re10++){

if(re10==1){

Writer.print("x ");

}else if(used10[x10x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss10=totalVariablesCount-1;ss10>=0;ss10--){

Writer.print(qc10[x10x][ss10]);

}

for(int tt10=0;tt10<512;tt10++){

if(tt10==0){

Writer.print(" : "+c10[x10x][tt10]);

}else{

Writer.print(", "+c10[x10x][tt10]);

}

}

Writer.println("");

barrier10++;

}

}

}

if(barrier10>0){

Writer.println("---------------------------");

}

}

}

//=============c11======================//

//輸出column 11

if(d11!=0){

Writer.println("===========================");

Writer.println("Column 11");

Writer.println("===========================");

for(int z11z=0;z11z<=totalVariablesCount;z11z++){

int barrier11=0;

for(int x11x=0;x11x<d11;x11x++){

int one11=0;

for(int y11y=totalVariablesCount-1;y11y>=0;y11y--){

if(qc11[x11x][y11y]=='1'){

one11++;

}

}

if(one11==z11z){

for(int re11=0;re11<2;re11++){

if(re11==1){

Writer.print("x ");

}else if(used11[x11x]=='X'){

Writer.print("v ");

}else{

Writer.print(" ");

}

for(int ss11=totalVariablesCount-1;ss11>=0;ss11--){

Writer.print(qc11[x11x][ss11]);

}

for(int tt11=0;tt11<1024;tt11++){

if(tt11==0){

Writer.print(" : "+c11[x11x][tt11]);

}else{

Writer.print(", "+c11[x11x][tt11]);

}

}

Writer.println("");

barrier11++;

}

}

}

if(barrier11>0){

Writer.println("---------------------------");

}

}

}

//======================================================//

//輸出Result

Writer.println("================================================================================================");

Writer.println("Result");

Writer.println("================================================================================================");

int[][] fin=new int[1024][2];

/\*將(value=1)的數字存入fin[][0],將全部不重複的prime implicant所指向的數字依次數加總

例如:有兩個 prime implicant有涵蓋"128",則將2存入fin[][1]\*/

for(int f=0;f<a;f++){

fin[f][0]=numbers[f];

fin[f][1]=0;

}

for(int f11=0;f11<d11;f11++){//將column 11的prime implicant所指向的數字依次數加總

if(used11[f11]!='X'){

for(int z11=0;z11<1024;z11++){

for(int u11=0;u11<a;u11++){

if(c11[f11][z11]==fin[u11][0]){

fin[u11][1]++;

}

}

}

}

}

for(int f10=0;f10<d10;f10++){//將column 10的prime implicant所指向的數字依次數加總

if(used10[f10]!='X'){

for(int z10=0;z10<512;z10++){

for(int u10=0;u10<a;u10++){

if(c10[f10][z10]==fin[u10][0]){

fin[u10][1]++;

}

}

}

}

}

for(int f9=0;f9<d9;f9++){//將column 9的prime implicant所指向的數字依次數加總

if(used9[f9]!='X'){

for(int z9=0;z9<256;z9++){

for(int u9=0;u9<a;u9++){

if(c9[f9][z9]==fin[u9][0]){

fin[u9][1]++;

}

}

}

}

}

for(int f8=0;f8<d8;f8++){//將column 8的prime implicant所指向的數字依次數加總

if(used8[f8]!='X'){

for(int z8=0;z8<128;z8++){

for(int u8=0;u8<a;u8++){

if(c8[f8][z8]==fin[u8][0]){

fin[u8][1]++;

}

}

}

}

}

for(int f7=0;f7<d7;f7++){//將column 7的prime implicant所指向的數字依次數加總

if(used7[f7]!='X'){

for(int z7=0;z7<64;z7++){

for(int u7=0;u7<a;u7++){

if(c7[f7][z7]==fin[u7][0]){

fin[u7][1]++;

}

}

}

}

}

for(int f6=0;f6<d6;f6++){//將column 6的prime implicant所指向的數字依次數加總

if(used6[f6]!='X'){

for(int z6=0;z6<32;z6++){

for(int u6=0;u6<a;u6++){

if(c6[f6][z6]==fin[u6][0]){

fin[u6][1]++;

}

}

}

}

}

for(int f5=0;f5<d5;f5++){//將column 5的prime implicant所指向的數字依次數加總

if(used5[f5]!='X'){

for(int z5=0;z5<16;z5++){

for(int u5=0;u5<a;u5++){

if(c5[f5][z5]==fin[u5][0]){

fin[u5][1]++;

}

}

}

}

}

for(int f4=0;f4<d4;f4++){//將column 4的prime implicant所指向的數字依次數加總

if(used4[f4]!='X'){

for(int z4=0;z4<8;z4++){

for(int u4=0;u4<a;u4++){

if(c4[f4][z4]==fin[u4][0]){

fin[u4][1]++;

}

}

}

}

}

for(int f3=0;f3<d3;f3++){//將column 3的prime implicant所指向的數字依次數加總

if(used3[f3]!='X'){

for(int z3=0;z3<4;z3++){

for(int u3=0;u3<a;u3++){

if(c3[f3][z3]==fin[u3][0]){

fin[u3][1]++;

}

}

}

}

}

for(int f2=0;f2<d2;f2++){//將column 2的prime implicant所指向的數字依次數加總

if(used2[f2]!='X'){

for(int z2=0;z2<2;z2++){

for(int u2=0;u2<a;u2++){

if(c2[f2][z2]==fin[u2][0]){

fin[u2][1]++;

}

}

}

}

}

for(int f1=0;f1<a;f1++){//將column 1的prime implicant所指向的數字依次數加總

if(used1[f1]!='X'){

for(int u1=0;u1<a;u1++){

if(numbers[f1]==fin[u1][0]){

fin[u1][1]++;

}

}

}

}

//===============================================//

Writer.print(" |");

for(int ff=0;ff<a;ff++){//調整輸出output.txt的格式

if(ff==0){

if(fin[ff][0]<10){

Writer.print(" "+fin[ff][0]);

}else if(fin[ff][0]>=10&&fin[ff][0]<100){

Writer.print(" "+fin[ff][0]);

}else if(fin[ff][0]>=100&&fin[ff][0]<1000){

Writer.print(" "+fin[ff][0]);

}else{

Writer.print(" "+fin[ff][0]);

}

}else{

if(fin[ff][0]<10){

Writer.print(", "+fin[ff][0]);

}else if(fin[ff][0]>=10&&fin[ff][0]<100){

Writer.print(", "+fin[ff][0]);

}else if(fin[ff][0]>=100&&fin[ff][0]<1000){

Writer.print(", "+fin[ff][0]);

}else{

Writer.print(", "+fin[ff][0]);

}

}

}

Writer.println("");

Writer.println("---------------------+--------------------------------------------------------------------------");

//========================================================//

/\*將所有的 prime implicant進行判斷,將一個prime implicant刪去,若沒有任何value 1的數被遺漏則該prime implicant為多餘的

,應被刪除,進行完這個過程後,會得出最後的essential prime implicant\*/

int error=0;

char[][] truth=new char[1024][10];

int truly=0;

int[][] Result=new int[1024][1024];

for(int ff1=a-1;ff1>=0;ff1--){

if(used1[ff1]!='X'){

for(int uu1=0;uu1<a;uu1++){

if(numbers[ff1]==fin[uu1][0]){

fin[uu1][1]--;

}

}

for(int xx1=0;xx1<a;xx1++){

if(fin[xx1][1]==0){

error++;

}

}

if(error!=0){

for(int ee1=0;ee1<totalVariablesCount;ee1++){

truth[truly][ee1]=qc[ff1][ee1];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int vv1=0;vv1<a;vv1++){

if(numbers[ff1]==fin[vv1][0]){

fin[vv1][1]++;

}

}

error=0;

}

}

}

for(int ff2=d2-1;ff2>=0;ff2--){

if(used2[ff2]!='X'){

for(int zz2=0;zz2<2;zz2++){

for(int uu2=0;uu2<a;uu2++){

if(c2[ff2][zz2]==fin[uu2][0]){

fin[uu2][1]--;

}

}

}

for(int xx2=0;xx2<a;xx2++){

if(fin[xx2][1]==0){

error++;

}

}

if(error!=0){

for(int ee2=0;ee2<totalVariablesCount;ee2++){

truth[truly][ee2]=qc2[ff2][ee2];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy2=0;yy2<2;yy2++){

for(int vv2=0;vv2<a;vv2++){

if(c2[ff2][yy2]==fin[vv2][0]){

fin[vv2][1]++;

}

}

}

error=0;

}

}

}

for(int ff3=d3-1;ff3>=0;ff3--){

if(used3[ff3]!='X'){

for(int zz3=0;zz3<4;zz3++){

for(int uu3=0;uu3<a;uu3++){

if(c3[ff3][zz3]==fin[uu3][0]){

fin[uu3][1]--;

}

}

}

for(int xx3=0;xx3<a;xx3++){

if(fin[xx3][1]==0){

error++;

}

}

if(error!=0){

for(int ee3=0;ee3<totalVariablesCount;ee3++){

truth[truly][ee3]=qc3[ff3][ee3];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy3=0;yy3<4;yy3++){

for(int vv3=0;vv3<a;vv3++){

if(c3[ff3][yy3]==fin[vv3][0]){

fin[vv3][1]++;

}

}

}

error=0;

}

}

}

for(int ff4=d4-1;ff4>=0;ff4--){

if(used4[ff4]!='X'){

for(int zz4=0;zz4<8;zz4++){

for(int uu4=0;uu4<a;uu4++){

if(c4[ff4][zz4]==fin[uu4][0]){

fin[uu4][1]--;

}

}

}

for(int xx4=0;xx4<a;xx4++){

if(fin[xx4][1]==0){

error++;

}

}

if(error!=0){

for(int ee4=0;ee4<totalVariablesCount;ee4++){

truth[truly][ee4]=qc4[ff4][ee4];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy4=0;yy4<8;yy4++){

for(int vv4=0;vv4<a;vv4++){

if(c4[ff4][yy4]==fin[vv4][0]){

fin[vv4][1]++;

}

}

}

error=0;

}

}

}

for(int ff5=d5-1;ff5>=0;ff5--){

if(used5[ff5]!='X'){

for(int zz5=0;zz5<16;zz5++){

for(int uu5=0;uu5<a;uu5++){

if(c5[ff5][zz5]==fin[uu5][0]){

fin[uu5][1]--;

}

}

}

for(int xx5=0;xx5<a;xx5++){

if(fin[xx5][1]==0){

error++;

}

}

if(error!=0){

for(int ee5=0;ee5<totalVariablesCount;ee5++){

truth[truly][ee5]=qc5[ff5][ee5];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy5=0;yy5<16;yy5++){

for(int vv5=0;vv5<a;vv5++){

if(c5[ff5][yy5]==fin[vv5][0]){

fin[vv5][1]++;

}

}

}

error=0;

}

}

}

for(int ff6=d6-1;ff6>=0;ff6--){

if(used6[ff6]!='X'){

for(int zz6=0;zz6<32;zz6++){

for(int uu6=0;uu6<a;uu6++){

if(c6[ff6][zz6]==fin[uu6][0]){

fin[uu6][1]--;

}

}

}

for(int xx6=0;xx6<a;xx6++){

if(fin[xx6][1]==0){

error++;

}

}

if(error!=0){

for(int ee6=0;ee6<totalVariablesCount;ee6++){

truth[truly][ee6]=qc6[ff6][ee6];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy6=0;yy6<32;yy6++){

for(int vv6=0;vv6<a;vv6++){

if(c6[ff6][yy6]==fin[vv6][0]){

fin[vv6][1]++;

}

}

}

error=0;

}

}

}

for(int ff7=d7-1;ff7>=0;ff7--){

if(used7[ff7]!='X'){

for(int zz7=0;zz7<64;zz7++){

for(int uu7=0;uu7<a;uu7++){

if(c7[ff7][zz7]==fin[uu7][0]){

fin[uu7][1]--;

}

}

}

for(int xx7=0;xx7<a;xx7++){

if(fin[xx7][1]==0){

error++;

}

}

if(error!=0){

for(int ee7=0;ee7<totalVariablesCount;ee7++){

truth[truly][ee7]=qc7[ff7][ee7];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy7=0;yy7<64;yy7++){

for(int vv7=0;vv7<a;vv7++){

if(c7[ff7][yy7]==fin[vv7][0]){

fin[vv7][1]++;

}

}

}

error=0;

}

}

}

for(int ff8=d8-1;ff8>=0;ff8--){

if(used8[ff8]!='X'){

for(int zz8=0;zz8<128;zz8++){

for(int uu8=0;uu8<a;uu8++){

if(c8[ff8][zz8]==fin[uu8][0]){

fin[uu8][1]--;

}

}

}

for(int xx8=0;xx8<a;xx8++){

if(fin[xx8][1]==0){

error++;

}

}

if(error!=0){

for(int ee8=0;ee8<totalVariablesCount;ee8++){

truth[truly][ee8]=qc8[ff8][ee8];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy8=0;yy8<128;yy8++){

for(int vv8=0;vv8<a;vv8++){

if(c8[ff8][yy8]==fin[vv8][0]){

fin[vv8][1]++;

}

}

}

error=0;

}

}

}

for(int ff9=d9-1;ff9>=0;ff9--){

if(used9[ff9]!='X'){

for(int zz9=0;zz9<256;zz9++){

for(int uu9=0;uu9<a;uu9++){

if(c9[ff9][zz9]==fin[uu9][0]){

fin[uu9][1]--;

}

}

}

for(int xx9=0;xx9<a;xx9++){

if(fin[xx9][1]==0){

error++;

}

}

if(error!=0){

for(int ee9=0;ee9<totalVariablesCount;ee9++){

truth[truly][ee9]=qc9[ff9][ee9];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy9=0;yy9<256;yy9++){

for(int vv9=0;vv9<a;vv9++){

if(c9[ff9][yy9]==fin[vv9][0]){

fin[vv9][1]++;

}

}

}

error=0;

}

}

}

for(int ff10=d10-1;ff10>=0;ff10--){

if(used10[ff10]!='X'){

for(int zz10=0;zz10<512;zz10++){

for(int uu10=0;uu10<a;uu10++){

if(c10[ff10][zz10]==fin[uu10][0]){

fin[uu10][1]--;

}

}

}

for(int xx10=0;xx10<a;xx10++){

if(fin[xx10][1]==0){

error++;

}

}

if(error!=0){

for(int ee10=0;ee10<totalVariablesCount;ee10++){

truth[truly][ee10]=qc10[ff10][ee10];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy10=0;yy10<512;yy10++){

for(int vv10=0;vv10<a;vv10++){

if(c10[ff10][yy10]==fin[vv10][0]){

fin[vv10][1]++;

}

}

}

error=0;

}

}

}

for(int ff11=d11-1;ff11>=0;ff11--){

if(used11[ff11]!='X'){

for(int zz11=0;zz11<1024;zz11++){

for(int uu11=0;uu11<a;uu11++){

if(c11[ff11][zz11]==fin[uu11][0]){

fin[uu11][1]--;

}

}

}

for(int xx11=0;xx11<a;xx11++){

if(fin[xx11][1]==0){

error++;

}

}

if(error!=0){

for(int ee11=0;ee11<totalVariablesCount;ee11++){

truth[truly][ee11]=qc11[ff11][ee11];//將所有essential prime implicant給放入truth[][],並用整數truly紀錄其個數

}

truly++;

error=0;

}else{

for(int yy11=0;yy11<1024;yy11++){

for(int vv11=0;vv11<a;vv11++){

if(c11[ff11][yy11]==fin[vv11][0]){

fin[vv11][1]++;

}

}

}

error=0;

}

}

}

//========================================//

//將所有不重複的prime implicant的數字組合和二進位表示法放在同一個array

int prime=0;

char[][] primei=new char[1024][10];//debug專用

for(int debug=0;debug<a+d2+d3+d4+d5+d6+d7+d8+d9+d10+d11;debug++){

for(int de=0;de<1024;de++){

Result[debug][de]=-1;}

}

for(int fff1=a-1;fff1>=0;fff1--){

if(used1[fff1]!='X'){

Result[prime][0]=numbers[fff1];

for(int pp1=0;pp1<totalVariablesCount;pp1++){

primei[prime][pp1]=qc[fff1][pp1];

}

prime++;}}

for(int fff2=d2-1;fff2>=0;fff2--){

if(used2[fff2]!='X'){

for(int rr2=0;rr2<2;rr2++){

Result[prime][rr2]=c2[fff2][rr2];

}

for(int pp2=0;pp2<totalVariablesCount;pp2++){

primei[prime][pp2]=qc2[fff2][pp2];

}prime++;}}

for(int fff3=d3-1;fff3>=0;fff3--){

if(used3[fff3]!='X'){

for(int rr3=0;rr3<4;rr3++){

Result[prime][rr3]=c3[fff3][rr3];

}

for(int pp3=0;pp3<totalVariablesCount;pp3++){

primei[prime][pp3]=qc3[fff3][pp3];

}prime++;}}

for(int fff4=d4-1;fff4>=0;fff4--){

if(used4[fff4]!='X'){

for(int rr4=0;rr4<8;rr4++){

Result[prime][rr4]=c4[fff4][rr4];

}for(int pp4=0;pp4<totalVariablesCount;pp4++){

primei[prime][pp4]=qc4[fff4][pp4];

}prime++;}}

for(int fff5=d5-1;fff5>=0;fff5--){

if(used5[fff5]!='X'){

for(int rr5=0;rr5<16;rr5++){

Result[prime][rr5]=c5[fff5][rr5];

}

for(int pp5=0;pp5<totalVariablesCount;pp5++){

primei[prime][pp5]=qc5[fff5][pp5];

}prime++;}}

for(int fff6=d6-1;fff6>=0;fff6--){

if(used6[fff6]!='X'){

for(int rr6=0;rr6<32;rr6++){

Result[prime][rr6]=c6[fff6][rr6];

}

for(int pp6=0;pp6<totalVariablesCount;pp6++){

primei[prime][pp6]=qc6[fff6][pp6];

}prime++;}}

for(int fff7=d7-1;fff7>=0;fff7--){

if(used7[fff7]!='X'){

for(int rr7=0;rr7<64;rr7++){

Result[prime][rr7]=c7[fff7][rr7];

}

for(int pp7=0;pp7<totalVariablesCount;pp7++){

primei[prime][pp7]=qc7[fff7][pp7];

}prime++;}}

for(int fff8=d8-1;fff8>=0;fff8--){

if(used8[fff8]!='X'){

for(int rr8=0;rr8<128;rr8++){

Result[prime][rr8]=c8[fff8][rr8];

}for(int pp8=0;pp8<totalVariablesCount;pp8++){

primei[prime][pp8]=qc8[fff8][pp8];

}prime++;}}

for(int fff9=d9-1;fff9>=0;fff9--){

if(used9[fff9]!='X'){

for(int rr9=0;rr9<256;rr9++){

Result[prime][rr9]=c9[fff9][rr9];

}

for(int pp9=0;pp9<totalVariablesCount;pp9++){

primei[prime][pp9]=qc9[fff9][pp9];

}prime++;}}

for(int fff10=d10-1;fff10>=0;fff10--){

if(used10[fff10]!='X'){

for(int rr10=0;rr10<512;rr10++){

Result[prime][rr10]=c10[fff10][rr10];

}for(int pp10=0;pp10<totalVariablesCount;pp10++){

primei[prime][pp10]=qc10[fff10][pp10];

}prime++;}}

for(int fff11=d11-1;fff11>=0;fff11--){

if(used11[fff11]!='X'){

for(int rr11=0;rr11<1024;rr11++){

Result[prime][rr11]=c11[fff11][rr11];

}

for(int pp11=0;pp11<totalVariablesCount;pp11++){

primei[prime][pp11]=qc11[fff11][pp11];

}prime++;}}

/\*System.out.println(prime);

for(int xxx=0;xxx<prime;xxx++){

for(int yyy=totalVariablesCount-1;yyy>=0;yyy--){

System.out.print(primei[xxx][yyy]+ " ");

}

System.out.println("");

}

for(int fff=0;fff<a;fff++){

System.out.println(fin[fff][0]+":"+fin[fff][1]);

}\*/

//依輸出格式輸出abcdefgh(i)(j)

for(int xx=prime-1;xx>=0;xx--){

int space=0;

for(int yy=totalVariablesCount-1;yy>=0;yy--){

if(yy==totalVariablesCount-1&&primei[xx][yy]!='-'){

Writer.print("a");

space++;}

if(yy==totalVariablesCount-2&&primei[xx][yy]!='-'){

Writer.print("b");

space++;}

if(yy==totalVariablesCount-3&&primei[xx][yy]!='-'){

Writer.print("c");

space++;}

if(yy==totalVariablesCount-4&&primei[xx][yy]!='-'){

Writer.print("d");

space++;}

if(yy==totalVariablesCount-5&&primei[xx][yy]!='-'){

Writer.print("e");

space++;}

if(yy==totalVariablesCount-6&&primei[xx][yy]!='-'){

Writer.print("f");

space++;}

if(yy==totalVariablesCount-7&&primei[xx][yy]!='-'){

Writer.print("g");

space++;}

if(yy==totalVariablesCount-8&&primei[xx][yy]!='-'){

Writer.print("h");

space++;}

if(yy==totalVariablesCount-9&&primei[xx][yy]!='-'){

Writer.print("i");

space++;}

if(yy==totalVariablesCount-10&&primei[xx][yy]!='-'){

Writer.print("j");

space++;}

if(primei[xx][yy]=='1'){

Writer.print(" ");

}

if(primei[xx][yy]=='0'){

Writer.print("'");

}

}

for(int sss=0;sss<=(10-space)\*2;sss++){//調整輸出的空格數

Writer.print(" ");

}

Writer.print("|");

for(int fff=0;fff<a;fff++){

if(fff==0){

int tru=0;

for(int tr=0;tr<1024;tr++){

if(Result[xx][tr]==fin[fff][0]){

tru++;

}

}

if(tru==0){

Writer.print(" ");

}else{

Writer.print(" x");}

}else{

int trut=0;

for(int trul=0;trul<1024;trul++){

if(Result[xx][trul]==fin[fff][0]){

trut++;

}

}

if(trut==0){

Writer.print(" ");

}else{

Writer.print(" x");}

}

}

Writer.println("");

}

Writer.println("---------------------+--------------------------------------------------------------------------");

Writer.print("F(A,B,C,D,E,F,G,H,I,J) = ");

//輸出最終的essential prime implicant

for(int xxx=truly-1;xxx>=0;xxx--){

if(xxx!=truly-1){

Writer.print(" + ");

}

for(int yyy=totalVariablesCount-1;yyy>=0;yyy--){

if(yyy==totalVariablesCount-1&&truth[xxx][yyy]!='-'){

Writer.print("a");

}

if(yyy==totalVariablesCount-2&&truth[xxx][yyy]!='-'){

Writer.print("b");

}

if(yyy==totalVariablesCount-3&&truth[xxx][yyy]!='-'){

Writer.print("c");

}

if(yyy==totalVariablesCount-4&&truth[xxx][yyy]!='-'){

Writer.print("d");

}

if(yyy==totalVariablesCount-5&&truth[xxx][yyy]!='-'){

Writer.print("e");

}

if(yyy==totalVariablesCount-6&&truth[xxx][yyy]!='-'){

Writer.print("f");

}

if(yyy==totalVariablesCount-7&&truth[xxx][yyy]!='-'){

Writer.print("g");

}

if(yyy==totalVariablesCount-8&&truth[xxx][yyy]!='-'){

Writer.print("h");

}

if(yyy==totalVariablesCount-9&&truth[xxx][yyy]!='-'){

Writer.print("i");

}

if(yyy==totalVariablesCount-10&&truth[xxx][yyy]!='-'){

Writer.print("j");

}

if(truth[xxx][yyy]=='1'){

Writer.print(" ");

}

if(truth[xxx][yyy]=='0'){

Writer.print("'");

}

}

}

Writer.flush();

Writer.close();

}catch(FileNotFoundException e){

System.out.println("IO Exception Happened");

}

}

}