1. 操作手冊: 本程式是以java去寫的。此程式在執行前需有一個名為input的txt檔，在此txt檔內輸入要變換成k map的東西，執行程式後，便會開始簡化並寫成minimum SOP，自動創立一個output.txt檔並將結果寫進裡面，程式即宣告結束。執行方式:打開cmd，輸入java.logic即可執行。
2. 程式撰寫流程:

基本上我分成4個步驟:

1. 將txt檔的東西分別拆開，以利之後運算:

先將檔案的內容抓出來存成String，並將這String以”+”號分解開來，到時候分別運算。

1. 將拆解的東西轉成k-map:

對拆解的東西個別運算:先設1個k-map，裡面所有東西目前為初始化階段。a、a’、b、b’、c、c’、d、d’分別對應k-map的其中1行或列，每當讀取到其中一個，這字母所對應的行列值+1，當抓完所有字母時，將最大值抓出來(其餘直接捨棄)，並將這最大值的位置當作1或don’t care(依拆解的東西而定)，到此完成拆解。接下來換下一個拆解的東西運算，重複上面過程，直到全部運算結束。

1. 開始化簡k-map(圈圈看):

從最大面積開始圈(example:4\*4、2\*4、2\*2)，以圈的左上角座標為基準點，將這個圈依序往右和往下平移，若圈裡皆不等於0且有1存在則圈起來，圈完後裡面的1變don’t care。對於當圈超過邊界時，超出的地方將直接移動到最左邊和最上面，超出多少則移動多少。當完成這個形狀的圈，會將圈的長與寬交換，再進行一次上面步驟，執行完後長會除2，之後再長與寬交換，長在除2，直到長與寬皆為1時停止。(此方法不會計算到1\*4和4\*1，因此這兩圖形獨立計算)

1. 將化簡後的k-map轉回字母和Minimum SOP:

依照圈起來的位置轉換成對應的數字，我有設好幾個function，其中一個是k-map的順序數字二維陣列，對應的位置上即為它代表的位置數字;而轉成字母也用了類似方法，對應的位置上即為代表它的字母。之後輸出所有結果，程式結束。

1. Source code:

import java.io.\*;

public class logic{

public static void CreateTxt(){ //create the output.txt

try{

File newTxt = new File("output.txt");

if( !newTxt.exists() ){

newTxt.createNewFile();

}

else{

System.out.println("The txt has already exist!");

}

} catch (IOException e) {

e.printStackTrace();

}

}

public static String input(){ //read the input.txt

String formula = " ";

try {

InputStreamReader reader = new InputStreamReader(new FileInputStream("input.txt"));

BufferedReader br = new BufferedReader(reader);

String line = " ";

while ((line= br.readLine()) != null) {

formula = line;

}

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return formula;

}

public static String[] vocabulary(int a){ //to let the k-map change into vocabulary (a,b,c,d)

String[] abcd;

if(a==0){

abcd=new String[8];

abcd[0]="a'";

abcd[1]="b";

abcd[2]="a";

abcd[3]="b'";

abcd[4]="c'";

abcd[5]="d";

abcd[6]="c";

abcd[7]="d'";

}

else{

if(a==1){

abcd=new String[6];

abcd[0]="a'";

abcd[1]="b";

abcd[2]="a";

abcd[3]="b'";

abcd[4]="c'";

abcd[5]="c";

}

else{

abcd=new String[4];

abcd[0]="a'";

abcd[1]="a";

abcd[2]="b'";

abcd[3]="b";

}

}

return abcd;

}

public static String[] vocabulary2(int a){ //to let the k-map change into vocabulary (a,b,c,d)

String[] abcd;

if(a==0){

abcd=new String[8];

abcd[0]="a'b'";

abcd[1]="a'b";

abcd[2]="ab";

abcd[3]="ab'";

abcd[4]="c'd'";

abcd[5]="c'd";

abcd[6]="cd";

abcd[7]="cd'";

}

else{

if(a==1){

abcd=new String[6];

abcd[0]="a'b'";

abcd[1]="a'b";

abcd[2]="ab";

abcd[3]="ab'";

abcd[4]="c'";

abcd[5]="c";

}

else{

abcd=new String[4];

abcd[0]="a'";

abcd[1]="a";

abcd[2]="b'";

abcd[3]="b";

}

}

return abcd;

}

public static int[][] array(int a){ //create k-map

int[][] abcd;

if(a==0){ //0 represent 4\*4 array

abcd = new int[4][4];

abcd[0][0] = 0;

abcd[0][1] = 4;

abcd[0][2] = 12;

abcd[0][3] = 8;

abcd[1][0] = 1;

abcd[1][1] = 5;

abcd[1][2] = 13;

abcd[1][3] = 9;

abcd[2][0] = 3;

abcd[2][1] = 7;

abcd[2][2] = 15;

abcd[2][3] = 11;

abcd[3][0] = 2;

abcd[3][1] = 6;

abcd[3][2] = 14;

abcd[3][3] = 10;

}

else{

if(a==1){ //1 represent 2\*4 array

abcd = new int[2][4];

abcd[0][0] = 0;

abcd[0][1] = 2;

abcd[0][2] = 6;

abcd[0][3] = 4;

abcd[1][0] = 1;

abcd[1][1] = 3;

abcd[1][2] = 7;

abcd[1][3] = 5;

}

else{ //2 represent 2\*2 array

abcd = new int[2][2];

abcd[0][0] = 0;

abcd[0][1] = 2;

abcd[1][0] = 1;

abcd[1][1] = 3;

}

}

return abcd;

}

public static void main(String args[]){

CreateTxt();

try{ //write things in the output.txt

File file = new File("output.txt");

BufferedWriter fw = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(file, true), "UTF-8"));

int distinguish = 9;

String[] fourfour = {"00","01","11","10"};

String[] twotwo = {"0","1"};

String formula = input();

for(int i=0;i<formula.length();i++){ //distinguish k-map is 4\*4 2\*4 2\*2

if(formula.contains("b")){

if(distinguish>2){

distinguish = 2; //represent 2\*2

}

}

if(formula.contains("c")){

if(distinguish>1){

distinguish = 1; //represent 2\*4

}

}

if(formula.contains("d")){

distinguish = 0; //represent 4\*4

}

}

String[] voctwo = vocabulary(distinguish);

String[] vocone = vocabulary2(distinguish);

int[][] truth = array(distinguish);

int[][] test = array(distinguish);

String replaced = formula.replaceAll("\\+"," "); //let char '+' change into " "

String[] one = replaced.split(" "); //split the vocabulary in the input.txt

for(int i=0;i<test.length;i++){ //initialize the int test[][]

for(int j=0;j<test[0].length;j++){

test[i][j]=0;

}

}

// create k-map

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

int write = 30; //30 represent 1

if(one[i].charAt(0)=='('){

write = 33; //33 represent X

}

int big=1;

for(int a=0;a<one[i].length();a++){

if(one[i].charAt(a)=='a'){

if(a+1<one[i].length() && one[i].charAt(a+1)!=')'&& one[i].charAt(a+1)!='b'&& one[i].charAt(a+1)!='c'&& one[i].charAt(a+1)!='d'){

for(int e=0;e<((test[0].length)/2);e++){

for(int f=0;f<test.length;f++){

test[f][e]+=1;

if(test[f][e]>big){

big=test[f][e];

}

}

}

}

else{

for(int c=((test[0].length)/2);c<test[0].length;c++){

for(int d=0;d<test.length;d++){

test[d][c]+=1;

if(test[d][c]>big){

big=test[d][c];

}

}

}

}

}

else{

if(one[i].charAt(a)=='b'){

if(test[0].length==2){

if(a+1<one[i].length() && one[i].charAt(a+1)!=')'&& one[i].charAt(a+1)!='a'&& one[i].charAt(a+1)!='c'&& one[i].charAt(a+1)!='d'){

for(int e=0;e<((test.length)/2);e++){

for(int f=0;f<test[0].length;f++){

test[e][f]+=1;

if(test[e][f]>big){

big=test[e][f];

}

}

}

}

else{

for(int c=((test.length)/2);c<test.length;c++){

for(int d=0;d<test[0].length;d++){

test[c][d]+=1;

if(test[c][d]>big){

big=test[c][d];

}

}

}

}

}

else{

if(a+1<one[i].length() && one[i].charAt(a+1)!=')'&& one[i].charAt(a+1)!='a'&& one[i].charAt(a+1)!='c'&& one[i].charAt(a+1)!='d'){

for(int e= 0;e<test[0].length;e+=3){

for(int f=0;f<test.length;f++){

test[f][e]+=1;

if(test[f][e]>big){

big=test[f][e];

}

}

}

}

else{

for(int c=(((test[0].length)/2)-1);c<(((test[0].length)/2)+1);c++){

for(int d=0;d<test.length;d++){

test[d][c]+=1;

if(test[d][c]>big){

big=test[d][c];

}

}

}

}

}

}

else{

if(one[i].charAt(a)=='c'){

if(a+1<one[i].length() && one[i].charAt(a+1)!=')'&& one[i].charAt(a+1)!='a'&& one[i].charAt(a+1)!='b'&& one[i].charAt(a+1)!='d'){

for(int e=0;e<((test.length)/2);e++){

for(int f=0;f<test[0].length;f++){

test[e][f]+=1;

if(test[e][f]>big){

big=test[e][f];

}

}

}

}

else{

for(int c=(test.length/2);c<test.length;c++){

for(int d=0;d<test[0].length;d++){

test[c][d]+=1;

if(test[c][d]>big){

big=test[c][d];

}

}

}

}

}

else{

if(one[i].charAt(a)=='d'){

if(a+1<one[i].length() && one[i].charAt(a+1)!=')'&& one[i].charAt(a+1)!='a'&& one[i].charAt(a+1)!='b'&& one[i].charAt(a+1)!='c'){

for(int e=0;e<test.length;e+=3){

for(int f=0;f<test[0].length;f++){

test[e][f]+=1;

if(test[e][f]>big){

big=test[e][f];

}

}

}

}

else{

for(int c=(((test.length)/2)-1);c<(((test.length)/2)+1);c++){

for(int d=0;d<test[0].length;d++){

test[c][d]+=1;

if(test[c][d]>big){

big=test[c][d];

}

}

}

}

}

}

}

}

}

for(int c=0;c<truth.length;c++){

for(int d=0;d<truth[0].length;d++){

if(test[c][d]==big){

truth[c][d]=write;

}

}

}

big=1;

for(int c=0;c<test.length;c++){ //initialize the test[][]

for(int d=0;d<test[0].length;d++){

test[c][d]=0;

}

}

}

if(distinguish==0){ //draw the k-map in ouput.txt(the first three row)

fw.append("========== K Map==========");

fw.newLine();

fw.append(" AB|");

fw.newLine();

fw.append("CD | 00 | 01 | 11 | 10");

fw.newLine();

fw.append("-----|----|----|----|----|");

fw.newLine();

}

else{

if(distinguish==1){

fw.append("========== K Map==========");

fw.newLine();

fw.append(" AB|");

fw.newLine();

fw.append("C | 00 | 01 | 11 | 10");

fw.newLine();

fw.append("-----|----|----|----|----|");

fw.newLine();

}

else{

fw.append("======K Map=====");

fw.newLine();

fw.append(" A |");

fw.newLine();

fw.append("B | 0 | 1 |");

fw.newLine();

fw.append("-----|----|----|");

fw.newLine();

}

}

for(int i=0;i<truth.length;i++){ //draw the k-map in ouput.txt

if(distinguish==0){

fw.append(" "+fourfour[i]+" |");

}

else{

fw.append(" "+twotwo[i]+" |");

}

for(int j=0;j<truth[0].length;j++){

if(truth[i][j]==30){ //change 30 to 1

truth[i][j]=1;

}

else{

if(truth[i][j]==33){ //change 33 to don't care

truth[i][j]=2; //2 represents dont't care(X)

}

else{

truth[i][j]=0;

}

}

if(truth[i][j]!=2){

fw.append(" "+truth[i][j]+" |"); //draw the k-map in output.txt

}

else{

fw.append(" X |"); //write X(don't care) in output.txt

}

}

fw.newLine();

if(distinguish==2){

fw.append("-----|----|----|");

fw.newLine();

}

else{

fw.append("-----|----|----|----|----|");

fw.newLine();

}

}

test = array(distinguish);

int existone = 0;

int number = 0;

int length\_a=truth[0].length;

int length\_b=truth.length;

int change =0;

int output\_num = 0;

boolean conti =true;

boolean outrow = false;

boolean outcolume = false;

String[] output\_voc=new String[20];

// begin to circle the k-map

while(conti){

if(length\_a<=truth[0].length && length\_b<=truth.length){

for(int i=0;i<truth[0].length;i++){ // i,j choose the location

for(int j=0;j<truth.length;j++){

for(int k=i;k<(length\_a+i);k++){ //k,l decide the area of circle (distinguish the circle is correct)

for(int l=j;l<(length\_b+j);l++){

if(k>=truth[0].length&& k<(length\_a+i)){ //if k over the array,it will return to k-array[0].length

k-=truth[0].length;

outrow = true;

}

if(l>=truth.length&& l<(length\_b+j)){ //if l over the array,it will return to l-array.length

l-=truth.length;

outcolume = true;

}

if(truth[l][k]!=0){

if(truth[l][k]==1){

existone=1;

}

number+=1;

}

if(outrow){

k+=truth[0].length;

outrow = false;

}

if(outcolume){

l+=truth.length;

outcolume = false;

}

}

}

if(number==(length\_a\*length\_b) && existone==1){ //if the circle is correct, then..

existone=0;

fw.append("group "+(output\_num+1)+":");

for(int k=i;k<(length\_a+i);k++){

for(int l=j;l<(length\_b+j);l++){

if(k>=truth[0].length&& k<(length\_a+i)){

k-=truth[0].length;

outrow = true;

}

if(l>=truth.length&& l<(length\_b+j)){

l-=truth.length;

outcolume = true;

}

fw.append("("+test[l][k]+")"); // write the number of circle in k-map

truth[l][k]+=1;

number=0;

if(outrow){

k+=truth[0].length;

outrow = false;

}

if(outcolume){

l+=truth.length;

outcolume = false;

}

}

}

fw.newLine();

output\_voc[output\_num]="";

fw.append("simplification of group "+(output\_num+1)+" -> ");

if(length\_a!=truth[0].length){ //change number to vocabulary

if(length\_a==2){

fw.append(voctwo[i]);

output\_voc[output\_num]+=voctwo[i];

}

else{

fw.append(vocone[i]);

output\_voc[output\_num]+=vocone[i];

}

}

if(length\_b!=truth.length){ //change number to vocabulary

if(length\_b==2){

fw.append(voctwo[j+(voctwo.length/2)]);

fw.newLine();

output\_voc[output\_num]+=voctwo[j+(voctwo.length/2)];

}

else{ //change number to vocabulary

fw.append(vocone[j+(vocone.length/2)]);

fw.newLine();

output\_voc[output\_num]+=vocone[j+(vocone.length/2)];

}

}

output\_num+=1;

}

else{

number=0;

existone=0;

}

}

}

if(length\_a==1 &&length\_b==1){ //if the circle area=1\*1 then out the while()

conti=false;

}

}

if(change==0 && length\_a!=length\_b){ //let length\_a and length\_b change(let the area a\*b be area b\*a)

change=length\_a;

length\_a=length\_b;

length\_b=change;

}

else{

change=0;

if(length\_a\*length\_b==4 &&length\_a==length\_b){

length\_a=1;

length\_b=4;

}

else{

if((length\_a/2)!=0){ //let length\_a/2 (let the area a\*b be area (a/2)\*b)

length\_a/=2;

}

}

}

}

fw.newLine();

if(distinguish==0){

fw.append("F(A,B,C,D) = ");

}

else{

if(distinguish==1){

fw.append("F(A,B,C) = ");

}

else{

fw.append("F(A,B) = ");

}

}

for(int i=0;i<output\_num;i++){ //write the changed vocabulary in output.txt

fw.append(output\_voc[i]);

if(i+1<output\_num){

fw.append("+");

}

}

fw.newLine();

fw.flush();

fw.close();

}catch (IOException e) {

e.printStackTrace();

}

}

}