AUTORES :

Miguel Angel Califa Urquiza código : 1160950.

Dúmar Hidalgo Monsalve código : 1161079 .

function varargout = guiFinal(varargin)

% GUIFINAL MATLAB code for guiFinal.fig

% GUIFINAL, by itself, creates a new GUIFINAL or raises the existing

% singleton\*.

%

% H = GUIFINAL returns the handle to a new GUIFINAL or the handle to

% the existing singleton\*.

%

% GUIFINAL('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in GUIFINAL.M with the given input arguments.

%

% GUIFINAL('Property','Value',...) creates a new GUIFINAL or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before guiFinal\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to guiFinal\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help guiFinal

% Last Modified by GUIDE v2.5 05-Dec-2016 22:15:44

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @guiFinal\_OpeningFcn, ...

'gui\_OutputFcn', @guiFinal\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before guiFinal is made visible.

function guiFinal\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to guiFinal (see VARARGIN)

% Choose default command line output for guiFinal

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes guiFinal wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = guiFinal\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on slider movement.

function slider1\_Callback(hObject, eventdata, handles)

% hObject handle to slider1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.

function slider1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor',[.9 .9 .9]);

end

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

clc;

h = handles.(sprintf('edit%d', 1 ));

dato = get(h,'string');

variable = str2num(dato);

nombre\_var = ['A' ,'B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','a1','b1','c1','d1','e1','f1','g1','h1','i1','j1','k1','l1','m1','n1','o1','p1','q1','r1','s1','t1','u1','v1','w1','x1','y1','z1']

ancho = 2^variable;

mapaK = [];

colNombre = {};

data = {};

for i=1:1:ancho+1

valorACargar = '';

if i==1

colNombre(1,1)={'valor decimal'}

for k=1:1:variable

%colNombre = [colNombre ' ',nombre\_var(k)];

mapaK = [mapaK,nombre\_var(k)];

end

colNombre(2,1)={mapaK}

colNombre(3,1)= {'x'};

else

valor = i-2

valorBin = dec2bin(valor,variable);

%for j =1:1:variable

%valorACargar = [valorACargar ;valor;valorBin ];

data(i-1,1) = {(valor)}

data(i-1,2) = {char(valorBin)}

%end

data(i-1,3) = {'0'}

end

end

data

%mapaK(1,2)

%mapaK = [1 2 3 ; 4 5 6; 7 8 9]

%colNombre

%mapaK

%for(length(mapaK,1)

%data(1:3,1) = {}

%mapaK = ['a' 'b' ; 'c' 'd']

f = figure('units','normalized','outerposition',[0 0 0.9 0.9]);

set(f,'Color',[0.147 0.56 0.183])

t = uitable('Parent', f,'units','normalized', 'Position', [0 0 0.8 1],'ColumnName',colNombre, 'Data', data,'ColumnFormat',{'char'})

set(t,'ColumnEditable',true(1,3))

btn = uicontrol('Style', 'pushbutton','BackgroundColor',[1 1 0], 'String', 'Calular :)','Callback',@(src,event)myfun(t,variable,f),'units','normalized', 'Position', [0.8 0.9 0.1 0.1]);

%disp('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TABLA ENTREGADA AL USUARIO \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

%calva

function myfun(t,variable,f)

try

h100 = waitbar(0,'Calculando');

steps = 16+(40\*variable)+1000;

step = 1;

%disp(variable)

if variable>7

%msgbox('vas a tener que esperar almenos 2 minutos para que cargue la tabla :D');

else if variable>5

% msgbox('ten paciencia .... , cargando');

else if variable>10

% msgbox('esto es una locura ! ten paciencia .... , cargando');

end

end

end

cantidad = variable;

%minter

tableData = get(t, 'data');

preDatos = tableData(:,3)

largo = length(preDatos);

unos = [];

equivalentes = [];

for i=1:1:largo

if str2num(preDatos{i}) == 1

%disp('funciono');

equivalentes = [equivalentes ; i-1 ];

end

end

unos = [];

for i=0:1:length(equivalentes)-1

binario = dec2bin(equivalentes(i+1),variable)

unos = [ unos ; binario];

end

%disp('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MINTERMINOS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

unos

numerosABC = equivalentes;

equivalentes

largo = size(numerosABC)

steps = 16+(4\*variable)+largo(2)\*100;

close(f);

%procesamiento

%declaracion de variables y bloques anFonimos

%disp('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*visualizacion de pares \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for i=1:1:variable+5

eval(['pares' num2str(i) num2str(i+1) ' =[];']);

eval(['data\_pares' num2str(i) num2str(i+1) ' =[];']);

eval(['data\_pares\_c' num2str(i) num2str(i+1) ' =[];']);

eval(['data\_pares\_complete' num2str(i) num2str(i+1) ' =[];']);

eval(['temporal' num2str(i) num2str(i+1) ' =[];']);

eval(['definitivoK' num2str(i) num2str(i+1) ' =[];']);

eval(['resultadottal' num2str(i) ' = [];']);

step = step+1

waitbar(step/steps )

end

%pause(100);

for i=1:1:variable+10

eval([' zero' num2str(i) ' = [];' ]);

end

%disp('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*NO TOCAR NADA DESDE AQUI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

%ya tengo cargado el mapa , ahora hare la tabla de verdad y hallare los

%unos

%tabla de verdad

datos = [];

datos2 = [];

largo = size(unos);

largo = largo(1)

%grupos

zero1 = [];

zero2 = [];

zero3 = [];

zero4 = [];

zero5 = [];

for largo=1:1:largo

numero = dec2bin(equivalentes(largo));

largoVector2 = length (numero);

cuenta = 0;

for largos=1:1:largoVector2

step = step+1

waitbar(step/steps )

if numero(largos)=='1'

cuenta=cuenta+1;

end

step = step+1

waitbar(step/steps );

waitbar(step/steps );

end

%disp('proceso el numero:');

%disp(unos(largo));

%disp(cuenta);

cuenta = num2str(cuenta);

if cuenta=='0'

%disp('cuenta tiene un uno');

zero1 = [zero1 ; unos(largo,:)];

else if cuenta == '1'

zero2 = [zero2;unos(largo,:)];

else if cuenta == '2'

zero3 = [zero3;unos(largo,:)];

else if cuenta == '3'

zero4 = [zero4;unos(largo,:)];

else if cuenta == '4'

zero5 = [zero5;unos(largo,:)];

else if cuenta == '5'

zero6 = [zero6;unos(largo,:)];

else if cuenta == '6'

zero7 = [zero7;unos(largo,:)];

else if cuenta == '7'

zero8 = [zero8;unos(largo,:)];

else if cuenta == '8'

zero9 = [zero9;unos(largo,:)];

else if cuenta == '9'

zero10 = [zero10;unos(largo,:)];

else if cuenta == '10'

zero11 = [zero11;unos(largo,:)];

else if cuenta == '11'

zero12 = [zero12;unos(largo,:)];

else if cuenta == '12'

zero13 = [zero13;unos(largo,:)];

else if cuenta == '13'

zero14 = [zero14;unos(largo,:)];

else if cuenta == '14'

zero15 = [zero15;unos(largo,:)];

else if cuenta == '15'

zero16 = [zero16;unos(largo,:)];

else if cuenta == '16'

zero17 = [zero17;unos(largo,:)];

else if cuenta == '17'

zero18 = [zero18;unos(largo,:)];

else if cuenta == '18'

zero19 = [zero19;unos(largo,:)];

else if cuenta == '19'

zero20 = [zero20;unos(largo,:)];

else if cuenta == '20'

zero21 = [zero21;unos(largo,:)];

else if cuenta == '21'

zero22 = [zero22;unos(largo,:)];

else if cuenta == '22'

zero23 = [zero23;unos(largo,:)];

else if cuenta == '23'

zero24 = [zero24;unos(largo,:)];

else if cuenta == '24'

zero25 = [zero25;unos(largo,:)];

else if cuenta == '25'

zero26 = [zero26;unos(largo,:)];

else if cuenta == '26'

zero27 = [zero27;unos(largo,:)];

else if cuenta == '27'

zero28 = [zero28;unos(largo,:)];

else if cuenta == '28'

zero29 = [zero29;unos(largo,:)];

else if cuenta == '29'

zero30 = [zero30;unos(largo,:)];

else if cuenta == '30'

zero31 = [zero31;unos(largo,:)];

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

end

%disp('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*grupos por unos \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

for i=1:1:variable+3

eval(['zero' num2str(i) ]);

step = step+1

waitbar(step/steps )

end

%pause(100);

%msgbox('aqui');

%pause(100);

%ahroa que he procesado cuantos unos tienen y los he agrupado ahora

%empiezo a procesar los grupos que tengan mas de un item

%iterando entre uno y dos

pares12=[];%pareja de iteraciones entre el grupo 1 y 2

pares23=[];%pareja de iteraciones entre el grupo 1 y 2

pares34=[];%pareja de iteraciones entre el grupo 1 y 2

pares45=[];%pareja de iteraciones entre el grupo 1 y 2

for grupo=1:1:variable% aqui estaba hasta 4 lo pongo hasta N ojo <----

largo = eval(['size(zero' num2str(grupo) ',1);'])%1

largo2 = eval(['size(zero' num2str(grupo+1) ',1);']) %1 con dos

%debo recorrer el uno dentro del dos

for cuenta2 = 1 : 1 : largo

for cuenta =1:1:largo2

step = step+1

waitbar(step/steps )

variable

numero = eval(['zero' num2str(grupo+1) '(cuenta,:' ');']);

numeroDec = cuenta;

numero1 = eval(['zero' num2str(grupo) '(cuenta2,:' ' );']);

numero1Dec = cuenta2;

numero1Decimal = bin2dec(numero1);

numero2Decimal = bin2dec(numero);

%disp('/-------------------------- NUMEROS ----------------------/');

%disp(numero);

%disp(numero1);

%aqui es donde debo comparar cad numero1 con su numero respectivo

num\_diferencia = 0 ;

numero\_definitivo\_precocina = [];

for bit = 1:1:variable

if numero1(bit)==numero(bit)

%no pasa nada , son iguales bit a bit

numero\_definitivo\_precocina = [numero\_definitivo\_precocina,numero1(bit)];

else

num\_diferencia = num\_diferencia+1;

numero\_definitivo\_precocina = [numero\_definitivo\_precocina,'-'];

%aqui hay que guardar los datos de los pares

end

end

if num\_diferencia==1

eval(['pares' num2str(grupo) num2str(grupo+1) '= [pares' num2str(grupo) num2str(grupo+1) ';numero\_definitivo\_precocina];' ]);

%disp('data pares');

%concatenador = 'y';

eval(['data\_pares' num2str(grupo) num2str(grupo+1) '= [data\_pares' num2str(grupo) num2str(grupo+1) '; ' num2str(numero1Decimal) '];'])

eval(['data\_pares\_c' num2str(grupo) num2str(grupo+1) '= [data\_pares\_c' num2str(grupo) num2str(grupo+1) '; ' num2str(numero2Decimal) '];'])

end

%msgbox('fallo');

%pause(5);

%disp('el numero definitivo es :');%disp(numero\_definitivo\_precocina);

%disp('el numero de contradicciones es :');

%disp(num\_diferencia);

%disp(numero1);

%disp(numero);

%disp(strcat('los pares ' ,num2str(grupo) ,' y ', num2str(grupo+1), ' son : '));

%disp(eval(['pares' ,num2str(grupo), num2str(grupo+1)]));

end

end

end

%recorro el vector de unos del dos y lo comparo con los unos del

%primero itero para saber cuantos dash(-) colocar

%h = msgbox('resultados parciales');

%pause(3);

%close(h);

%disp('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* resultados parciales \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

%for i=1:1:variable+1

%msgbox('aqui?');

% mensaje = 'los pares ';

% mensaje2 = 'son --> ' ;

% eval(['A' num2str(i) '= pares' num2str(i) num2str(i+1) ';']);

%eval([ 'if length(A' num2str(i) ') > 1, pares' num2str(i) num2str(i+1) ',data\_pares' num2str(i) num2str(i+1) ',data\_pares\_c' num2str(i) num2str(i+1) ',end']);

%eval(['%disp( pares' num2str(i) num2str(i+1) ');']);

%end

%pause(1000);

% for i=1:1:variable+1

% eval(['pares' num2str(i) num2str(i+1) ]);

%end

%pause(100);

%ahora debo iterar entre los grupos que quedan

%disp(size(pares12));

%disp(pares12);

%disp(size(pares12,1));

%ahora hago la misma iteracion pero con los preresultados

%seccion resultado final

resultadottal = [];

resultadottal1 = [];

resultadottal2 = [];

resultadottal3 = [];

%disp('---------------------------------------------');

%disp('segundo analisis de grupo');

% añadido el 11/19/2016 por miguell y dumar :p

step = 0.3\*steps

for cuenta1 = 1 : 1 : variable-1 % numero de iteraciones AQUI TAMBIEN IVA 3 pongo N-1

%analizare 3 grupos

%12 con 23

%23 con 34

%34 con 45

%disp('for');

vector1 = eval(['pares' num2str(cuenta1) num2str(cuenta1+1) ';'])%1

vector2 = eval(['pares' num2str(cuenta1+1) num2str(cuenta1+2) ';'])%1

%vector2\_eqDec = eval(['pares' num2str(cuenta1) num2str(cuenta1+1) ''])%1

largo1 = eval(['size(pares' num2str(cuenta1) num2str(cuenta1+1) ',2)' ';'])%1

largo2 = eval(['size(pares' num2str(cuenta1+1) num2str(cuenta1+2) ',2)' ';'])%1

ancho1 = eval(['size(pares' num2str(cuenta1) num2str(cuenta1+1) ',1)' ';'])%1

ancho2 = eval(['size(pares' num2str(cuenta1+1) num2str(cuenta1+2) ',1)' ';'])%1

error = 0;

temporal = [];

temporal2 = [];

for vector1Vertical = 1 : 1 : ancho1

for vector2Vertical = 1 : 1 : ancho2

%disp('-------------------vectores-------------------------------');

vector1(vector1Vertical,:);

vector2(vector2Vertical,:);

numeroPrecocina = [];

pretemp1=[];

pretemp2=[];

pretemp3=[];

pretemp4=[];

step = step+1

waitbar(step/steps )

%ahora evaluo el vector 1 punto a punto con el vector 2 teniendo en cuenta que juntos tengan el mismo tamaño

if largo1 == largo2

for i = 1 : 1 : largo1

punto1 = vector1(vector1Vertical,i);

punto2 = vector2(vector2Vertical,i);

if punto1 == punto2

numeroPrecocina = [numeroPrecocina , punto1];

%disp('iguales');

else

numeroPrecocina = [numeroPrecocina , '-'];

%disp('diferentes');

%disp('y el error es ');

error = error + 1

%disp(error);

end

end

if error>=2

%disp(numeroPrecocina);

%disp('el numero NO pasa :( y el numero de error es ');

%disp(error);

error = 0;

else

error = 0;

%disp('el numero si pasa y es : ');

numeroPrecocina

variable = '';

variable = mat2str(numeroPrecocina);

%disp('----------------------RESULTADO TTAL -----------------');

eval(['resultadottal' num2str(cuenta1) ]);

eval(['resultadottal' num2str(cuenta1) ' = [' 'resultadottal' num2str(cuenta1) ';' variable ']' ';' ])%1

eval(['data\_pares' num2str(cuenta1) num2str(cuenta1+1)]);

eval(['data\_pares\_c' num2str(cuenta1) num2str(cuenta1+1)]);

eval(['data\_pares' num2str(cuenta1+1) num2str(cuenta1+2)]);

eval(['data\_pares\_c' num2str(cuenta1+1) num2str(cuenta1+2)]);

pretemp1 = [eval(['data\_pares' num2str(cuenta1) num2str(cuenta1+1) '(' num2str(vector1Vertical) ',:);' ])]

pretemp2 = [eval(['data\_pares\_c' num2str(cuenta1) num2str(cuenta1+1) '(' num2str(vector1Vertical) ',:);'])]

vector2Vertical

pretemp3 = [eval(['data\_pares' num2str(cuenta1+1) num2str(cuenta1+2) '(' num2str(vector2Vertical) ',:);'])]

pretemp4 = [eval(['data\_pares\_c' num2str(cuenta1+1) num2str(cuenta1+2) '(' num2str(vector2Vertical) ',:);' ])]

A = [ pretemp1;pretemp2 ; pretemp3 ; pretemp4];

B = eval(['temporal' num2str(cuenta1) num2str(cuenta1+1) ';']);

Concatenada = horzcat(A,B);

eval(['temporal' num2str(cuenta1) num2str(cuenta1+1) '= Concatenada ;']);

eval(['definitivoK' num2str(cuenta1) num2str(cuenta1+1) '= Concatenada ;']);

%pause(11000);

end

end

end

end

end

%disp('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*preresultados2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

%disp(cantidad);

%for i = 1 : 1 : cantidad-1 % <-- AQUI TAMBIEN ESTABA 3 pongo N-1

% eval(['if size(temporal' num2str(i) num2str(i+1) '),1>0,temporal' num2str(i) num2str(i+1) ',end' ]);

% eval(['size(resultadottal' num2str(i) ')' ]);

% eval(['if size(resultadottal' num2str(i) '),2>0,resultadottal' num2str(i) ',end' ]);

%end

% HASTA AQUI FUNCIONA NO TOCAR !!!!

%size(resultadottal1)

%%disp(resultadottal1)

%temporal12

%%disp(resultadottal2);

%%disp(resultadottal3);

% analisis de pares y ultima tabla dinamica :p

paresTotal = [];

temporalPares = [];

aAnalizar = [];

rtattal = [];

for i=1:1:cantidad-1

step = step+1

waitbar(step/steps );

waitbar(step/steps )

temporalPares = eval(['temporal' num2str(i) num2str(i+1)]);

largo = size(temporalPares);

largo = largo(1);

if(largo >0)

rtattal=horzcat(rtattal,temporalPares);

end

end

rtattal

temporalComparacion=[];

temporal = [];

datos\_padre = [];

temporal\_padre = [];

for i=1:1:cantidad

temporalPares = eval(['pares' num2str(i) num2str(i+1)]);

largo = size(temporalPares)

largo = largo(1)

for j=1:1:largo

numero = eval(['data\_pares' num2str(i) num2str(i+1) ]);

numero = numero(j);

numero1 = eval(['data\_pares\_c' num2str(i) num2str(i+1) ]);

numero1=numero1(j);

numeros = horzcat(num2str(i), num2str(i+1));

temporal\_padre = [str2num(numeros) ,(j)];

temporal = horzcat(numero,numero1);

temporalComparacion = vertcat(temporalComparacion,temporal);

datos\_padre = vertcat(datos\_padre,temporal\_padre);

end

end

temporalComparacion;

datos\_padre; % contiene la info que encuentro en cada uno de los temporales para comparacion

rtattal;

%disp('aca estoy');

rttattalbck = rtattal;

rtattalNew =[];rtattalNew2 =[];

temporalMuevertattal = [];

largo = size(rtattal);

for i=1:1:largo(2)

for j=1:1:largo(1)/2

step = step+1

waitbar(step/steps );

numero1 = num2str(rtattal(j,i));

numero2 = num2str(rtattal(j+2,i));

rtattalNew2 = eval(['cat(2,[' numero1 '],' '[' numero2 '])' ]);

rtattalNew = [rtattalNew;rtattalNew2];

end

end

%disp('aqui')

rtattal = rtattalNew;

% pause(1000);

largo = size(rtattal);

largo = largo(1);

largo2 = size(temporalComparacion);

largo2 = largo2(1);

eliminado = [];

%disp('sigo al for');

for i=1:1:largo

for j=1:1:largo2

step = step+1

waitbar(step/steps )

rtattal(i,:);

temporalComparacion(j,:);

largo3 = size(rtattal(i,:));

largo3 = largo3(2);

contador =0;

for k=1:1:largo3

rtattal(i,k)

temporalComparacion(j,k)

if rtattal(i,k) == temporalComparacion(j,k)

contador = contador+1

end

end

if contador==2

matriz = [];

%disp('elimino');

num2str(datos\_padre(j));

selector = (datos\_padre(j,2));

matriz = eval(['pares' num2str(datos\_padre(j)) ]);

matriz = matriz(selector,:) ;

eliminado = vertcat(eliminado,matriz);

%eval(['pares' num2str(datos\_padre(j)) '(' num2str(datos\_padre(j,2)) ',:)' ])

%eval(['pares' num2str(datos\_padre(j)) '(' num2str(datos\_padre(j,2)) ')' '=[]'])

end

end

end

eliminado % lo que debo eliminar

total = [];

par = [];

par3 = [];

par4= [];

par\_total = [];

tempPar = [];

for i=1:1:cantidad+1

par = eval(['pares' num2str(i) num2str(i+1)]);

par3 = eval(['data\_pares' num2str(i) num2str(i+1)]);

par4= eval(['data\_pares\_c' num2str(i) num2str(i+1)]);

largo = size(par);

largo = largo(1);

if largo>0

total = vertcat(total,par);

par\_total = vertcat(par\_total,horzcat(par3,par4));

end

end

total

total2 = [];

%elimino el mugre

largo = size(total);

largo2 = size(eliminado);

contador = 0;

par\_total\_2 =[];

par\_total\_2\_E =[];

for i=1:1:largo(1)

contador2 = 0;

for j=1:1:largo2(1)

contador = 0;

step = step+1

waitbar(step/steps )

%total(i,:)

%eliminado(i,:)

largos = size(total);

for k=1:1:largos(2) %cantidad

if total(i,k) == eliminado(j,k);

contador = contador+1;

else

end

end

if contador<cantidad

contador2 = contador2+1;

end

end

largo2(1)

if contador2==largo2(1)

total2 = vertcat(total2,total(i,:))

par\_total\_2 = vertcat(par\_total\_2,par\_total(i,:));

end

end

%disp('-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PRE ANTESALA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

step = 0.5\*steps

total2

par\_total\_2

binario = []

pares = []

par = []

%definitivoK12

par\_total\_2

for i=1:1:cantidad+1

binario = eval(['resultadottal' num2str(i)]);

%disp('largo del binario <--- ');

largo = size(binario)

if largo(1)>0

total2 = [total2;binario];

pares = [];

step = step+1

waitbar(step/steps )

%disp('evaluando')

par = eval(['definitivoK' num2str(i) num2str(i+1)]);

largoPar = size(par);

largoPar = largoPar(2);

a = par;

at = [];

[r,c] = size(a);

for j=1:1:c

for K = 1 : r

at(j,K) = a(K,j);

end

end

at

%disp('listos para unir la ultima parte');

par\_total\_2\_E = [par\_total\_2\_E ; at]

end

end

%disp('preparados para matriz');

numerosABC;

a = numerosABC;

at = [];

[r,c] = size(a);

for i = 1 : r

at(1,i) = a(i,1);

end

at

numerosABC =at ;

par\_total\_2;

par\_total\_2\_E;

total2;

largoE = size(par\_total\_2\_E); %no tocar!

%pause(1000);

largoParE=size(par\_total\_2\_E) ; % debo concatenar ahora todos los valores de los pares y eliminar a los

largoParN = size(par\_total\_2);

paresCompletos = [];

if largoParN(1)==0 && largoParN(2)==0

paresCompletos = [par\_total\_2\_E];

else

for b=1:1:largoParE(2)-largoParN(2)

%disp('------------numeroE----------------------');

matrizVacia = [];

for x=1:1:largoParN(1);

step = step+1

waitbar(step/steps )

matrizVacia = [matrizVacia;'Þ'];

end

matrizVacia;

par\_total\_2(:,largoParE(2)-largoParN(2));

par\_total\_2(:,largoParE(2)-largoParN(2)+b)= matrizVacia;

end

par\_total\_2

paresCompletos = [par\_total\_2 ; par\_total\_2\_E ];

end

disp('antesala del final ');

paresCompletos

numerosABC

faltantes=[]

%faltantes = numerosABC

largoPares= size(paresCompletos) % largo de las parejas

largoMinTer = size(numerosABC)% largo de los minterminos

temporal = 0;

for k=1:1:largoMinTer(2)

temporal=0

contador=0

for i=1:1:largoPares(1)

for j =1:1:largoPares(2)

disp('comparando')

disp(numerosABC(k))

disp(paresCompletos(i,j))

temporal = temporal+1

if numerosABC(k)~=paresCompletos(i,j)

contador = contador+1

end

% debo comparar cada par con cada uno de los 4 faltantes

end

end

if contador==temporal

faltantes= [faltantes;numerosABC(k)]

end

end

if largoPares(1)==0

faltantes= numerosABC

end

faltantes

largo = size(faltantes)

minTerminosSinPar = []

valores = [ 'A';'B';'C';'D';'E';'F';'G';'H';'I';'J';'K';'L';'M';'N';'O';'P';'Q';'R';'S';'T';'U';'V';'W';'X';'Y';'Z' ];

for i=1:1:largo(1)

binario = dec2bin(faltantes(i),cantidad)

for j =1:1:cantidad

if binario(j)=='1'

minTerminosSinPar = horzcat(minTerminosSinPar,valores(j,:))

else

minTerminosSinPar = horzcat(minTerminosSinPar,valores(j,:),char(39))

end

end

if i<largo(1)

minTerminosSinPar = horzcat(minTerminosSinPar,'+')

end

end

minTerminosSinPar

%momento de analizar y hacer la matriz completa

matrizCompletita = {} ;

largo = size(numerosABC);

%matrizCompletita(1,1) = [char('-')];

for i=1:1:cantidad

step = step+1

waitbar(step/steps )

matrizCompletita(1,i)={char('-')};

end

for i=cantidad:1:largo(2)+cantidad

tamano = size(numerosABC);

i+1-cantidad;

step = step+1

waitbar(step/steps )

if tamano(2)>= (i+1-cantidad);

char(num2str(numerosABC(1,1+i-cantidad)));

matrizCompletita(1,i+1) = {(numerosABC(1,1+i-cantidad))};

end

end

largo2= size(paresCompletos);

largo5 = size(total2);

largo = size(numerosABC);

for i=2:1:largo5(1)+1

%filas

for j=1:1:largo(2)+1

%columna

%empiezo en la fila 2 :p

%disp('imprimiendo fila2');

if j==1

largosE22 = size(total2);

for n=1:1:largosE22(2)

n

i-1

% char(total2(n,i-1))

% matrizCompletita(n,i)

matrizCompletita(i,n) = num2cell(total2(i-1,n));

end

else

largo3 = size(paresCompletos);

largo3 = largo3(2);

%disp('analizo')

numerosABC(1,j-1);

%disp('contra')

paresCompletos(i-1,:);

estado='0';

for k=1:1:largo3

%disp('analizo2')

numerosABC(1,j-1);

%disp('contra2')

paresCompletos(i-1,k);

step = step+1

waitbar(step/steps )

if numerosABC(1,j-1)==paresCompletos(i-1,k)

matrizCompletita(i,j+cantidad-1) = num2cell('x');

estado='1'

%disp('cambio de estado');

else if estado=='0'

matrizCompletita(i,j+cantidad-1) = num2cell('-');

end

end

end

end

end

end

matrizCompletita

largo = size(matrizCompletita);

cantidad;

for i=3:2:largo(1)

%fila

for j=cantidad+1:1:largo(2)

%columna

%pregunta por el vecino de arriba

%disp(' voy en i=');

%disp(i);

%disp('y j=');

%disp(j);

step = step+1

waitbar(step/steps )

if (i-1)<=largo(1)

%disp(' voy en i=');

%disp(i);

%disp('y j=');

%disp(j);

A = matrizCompletita(i,j);

b = matrizCompletita(i-1,j);

%disp('antes')

strcmp(A,b)

%disp('despues')

if strcmp(A,b)

%disp('he pasado');

matrizCompletita(i-1,j) = cellstr('-') ;% iva == -

%disp(':)');

end

end

if (i+1)<=largo(1)

matrizCompletita(i,j);

matrizCompletita(i+1,j);

if strcmp(cell2mat(matrizCompletita(i,j)),cell2mat(matrizCompletita(i+1,j)));

matrizCompletita(i+1,j) = cellstr('-');% iva == -

end

end

end

end

matrizCompletita

%pause(1000);

%momento de reorganizar

matrizTemporal = {};

largoE2 = size(matrizCompletita);

for i=largoE(1)+2:1:largoE2(1)

step = step+1

waitbar(step/steps )

matrizTemporal(i,:) = matrizCompletita(i-largoE(1),:);

end

for i=1:1:1

step = step+1

waitbar(step/steps )

matrizTemporal(i,:) = matrizCompletita(i,:);

end

largoE2(1)-i+1

for i=2:1:largoE(1)+1

step = step+1

waitbar(step/steps )

matrizTemporal(i,:) = matrizCompletita(largoE2(1)-i+2,:);

end

matrizTemporal;

matrizCompletita =matrizTemporal

step = 0.7\*steps

valoresN = [ '-A';'-B';'-C';'-D';'-E';'-F';'-G';'-H';'-I';'-J';'-K';'-L';'-M';'-N';'-O';'-P';'-Q';'-R';'-S';'-T';'-U';'-V';'-W';'-X';'-Y';'-Z' ];

valores = [ 'A';'B';'C';'D';'E';'F';'G';'H';'I';'J';'K';'L';'M';'N';'O';'P';'Q';'R';'S';'T';'U';'V';'W';'X';'Y';'Z' ];

solucion = [];

repeticion = 0;

largo

for i=2:1:largo(1)

%fila

contador = 0;

%disp('cantidad de contador');

largo(2)-cantidad

for j=cantidad+1:1:largo(2)

%columna

%pregunta por el vecino de arriba

%disp('columna es -->');

matrizCompletita(i,:);

%matrizCompletita(i,j)

step = step+1

waitbar(step/steps )

if (cell2mat(matrizCompletita(i,j))=='-')

else

contador = contador+1;

end

end

solutemp = [];

solucion2 = [];

%disp('momento de pensar');

if contador < largo(2)-cantidad

%disp('creando una solucion');

if repeticion == 0

repeticion = repeticion+1;

else

solucion = horzcat(solucion,'+')

solutemp = vertcat(solutemp,solucion2);

solucion2 = [];

% solutemp = vertcat(solutemp,solucion);

end

for m=1:1:cantidad

if cell2mat(matrizCompletita(i,m))=='0'

solucion = horzcat(solucion,valores(m,:),char(39));

solucion2 = horzcat(solucion2,valores(m,:),char(39));

else

if cell2mat(matrizCompletita(i,m))=='1'

solucion = horzcat(solucion,valores(m,:));

solucion2 = horzcat(solucion2,valores(m,:));

end

end

end

end

end

solucion

solutemp = {};

largo = size(solucion);

largo2 = size(minTerminosSinPar)

if largo2(2)~=0 && largo(2)~=0

solucion=horzcat(solucion,'+',minTerminosSinPar);

disp('concatenando lo que falta')

else if largo2(2)~=0

disp('concatenando lo unico q hay')

solucion=horzcat(minTerminosSinPar);

end

end

solutemp;

largo = size(solucion);

if largo(2)~=0

h = msgbox(solucion,'Solucion','info');

set(h, 'position', [100 440 400 100]);

ah = get( h, 'CurrentAxes' );

ch = get( ah, 'Children' );

set( ch, 'FontSize', 20 ); %makes text bigger

else

msgbox('solucion vacia , consulta al administrador utilizare el algoritmo de reserva');

numerosABC

%disp(variable)

syms A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

variable2 = [ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z]

largo = size(numerosABC)

ecuacion = '';

for i=1:1:largo(2)

numero = numerosABC(1,i)

numeroBin = dec2bin((numero),(variable))

largo = size(numeroBin)

%ecuacion = horzcat(ecuacion,'+')

if i>1

ecuacion = strcat(ecuacion,'+');

end

for i=1:1:largo(2)

if(numeroBin(i)=='1')

%disp('uno encontrado');

if i>1

ecuacion = strcat(ecuacion,'\*',variable2(i))

else

ecuacion = strcat(ecuacion,variable2(i))

end

else if(numeroBin(i)=='0')

if i>1

ecuacion = strcat(ecuacion,'\*',strcat('(1-', variable2(i),')'))

else

ecuacion = strcat(ecuacion,strcat('(1-', variable2(i),')'))

end

end

end

end

ecuacion

solucion = eval(['simplify(' ecuacion ')'])

end

end

disp('pasos');

disp('hora cuchi cuchi');

par\_total\_2

step

waitbar(1)

pause(2);

close(h100)

largo = size(solucion)

%if largo(2)~=0

% h = msgbox(char(solucion),'Solucion','info');

% set(h, 'position', [100 440 800 100]);

% ah = get( h, 'CurrentAxes' );

%ch = get( ah, 'Children' );

%set( ch, 'FontSize', 20 ); %makes text bigger

%pause(20)

%close(h)

%end

catch

%ME.identifier

%exception = MException.last()

%getReport(exception,'basic')

%pause(1000)

%close(h)

msgbox('algo salio mal , utilizare el algoritmo de reserva');

numerosABC

%disp(variable)

syms A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

variable2 = [ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z]

largo = size(numerosABC)

ecuacion = '';

for i=1:1:largo(2)

numero = numerosABC(1,i)

numeroBin = dec2bin((numero),(variable))

largo = size(numeroBin)

%ecuacion = horzcat(ecuacion,'+')

if i>1

ecuacion = strcat(ecuacion,'+');

end

for i=1:1:largo(2)

if(numeroBin(i)=='1')

%disp('uno encontrado');

if i>1

ecuacion = strcat(ecuacion,'\*',variable2(i))

else

ecuacion = strcat(ecuacion,variable2(i))

end

else if(numeroBin(i)=='0')

if i>1

ecuacion = strcat(ecuacion,'\*',strcat('(1-', variable2(i),')'))

else

ecuacion = strcat(ecuacion,strcat('(1-', variable2(i),')'))

end

end

end

end

ecuacion

solucion = eval(['simplify(' ecuacion ')'])

end

end

largo = size(solucion)

if largo(2)~=0

h = msgbox(char(solucion),'Solucion','info');

set(h, 'position', [100 440 800 100]);

ah = get( h, 'CurrentAxes' );

ch = get( ah, 'Children' );

set( ch, 'FontSize', 20 ); %makes text bigger

end

%pause(10000);

% temporalComparacion vs aAnalizar

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

web('http://www.facebook.com/onmotica')

% --- Executes on button press in pushbutton3.

function pushbutton3\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

msgbox('buscanos en facebok como Miguel Califa , Dumar hidaldo , Onmotica.com');

web('https://github.com/miguel5612/solucionadorMapaK')

% --- Executes on button press in pushbutton4.

function pushbutton4\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

msgbox('whatsapp info : +57 3192597748 Colombia');

%web('https://www.facebook.com/miguel.califa');

web('https://github.com/miguel5612/solucionadorMapaK')

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.

function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- If Enable == 'on', executes on mouse press in 5 pixel border.c

% --- Otherwise, executes on mouse press in 5 pixel border or over pushbutton1.

function pushbutton1\_ButtonDownFcn(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton5.

function pushbutton5\_Callback(hObject, eventdata, handles)

web('http://www.miguelcalifa.me');

% hObject handle to pushbutton5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.

function axes1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to axes1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

% --- If Enable == 'on', executes on mouse press in 5 pixel border.

% --- Otherwise, executes on mouse press in 5 pixel border or over text3.

function text3\_ButtonDownFcn(hObject, eventdata, handles)

% hObject handle to text3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.

function axes2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to axes2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2

axes(hObject)

imshow('base2.png');