INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

TEORÍA COMPUTACIONAL

2CM3

PROFESORA: LUZ MARÍA

PROYECTO MÁQUINA DE TURING

QUINTANA RUÍZ AJITZI RICARDO	**2017631261**
REYES MEDRANO ALEXIS DANIEL	**2013081006**
VÁZQUEZ MORENO MARCOS OSWALDO	**2016601777**

GRUPO 2CM4

FECHA DE ENTREGA: 13 DE JUNIO DE 2018

INTRODUCCIÓN

En el siguiente proyecto se desarrollará una codificación, simulación y ejecución de una máquina de Turing la cual va a resolver una suma de números binarios de hasta 8 bits, recorriendo una cinta limpia y bien formada teniendo el cabezal en el simulador de JFLAP al inicio del lado izquierdo de la cadena ingresada, en cuanto a nuestro programa desarrollado en Java el cabezal está localizado del lado derecho de la misma cinta.

DEFINICIÓN

La máquina de Turing es el modelo de autómata con máxima capacidad computacional, pues podemos desplazarnos tanto a la derecha como a la izquierda y sobre escribir símbolos en la cinta de entrada. Una MT es una séptupla M = (Q, q0, F, Σ, Γ, ă, δ), donde:

- ✓ **Q** es el conjunto finito de estados internos.
- ✓ **q0** ∈ Q es el estado inicial.
- ✓ **F** 6= φ es el conjunto finito de estados de aceptaci´on, donde F ⊆ Q. Σ es el alfabeto de entrada.
- ✓ **Γ** es el alfabeto de cinta tal que Σ ⊆ Γ.
- ✓ **B** ∈ Γ es el símbolo blanco tal que ă 6∈ Σ.
- ✓ **δ** es la función de transición, donde δ : Q × Γ → Q × Γ × {←, −, →}, es decir, recibe un estado y un símbolo de la cinta para devolver otro estado, otro símbolo y una dirección de movimiento. δ es una función parcial, pues puede que no esté definida en algunos elementos del dominio. La transición δ(q, a) = (p, b, d) significa que estando en el estado q escaneando el símbolo a: borramos a, escribimos b y nos movemos al estado p, además avanzamos a la cinta hacia la izquierda (si d =←), hacia la derecha (si d =→) o nos quedamos ahí (si d = −).

IMPLEMENTACIÓN:

Tendremos una aceptación de cadenas de 8 bits como máximo, la cual llevará a cabo una suma de binarios posición a posición, realizando si es necesario un acarreo como se ve en la materia de Diseño de Sistemas Digitales o Fundamentos de Diseño Digital.

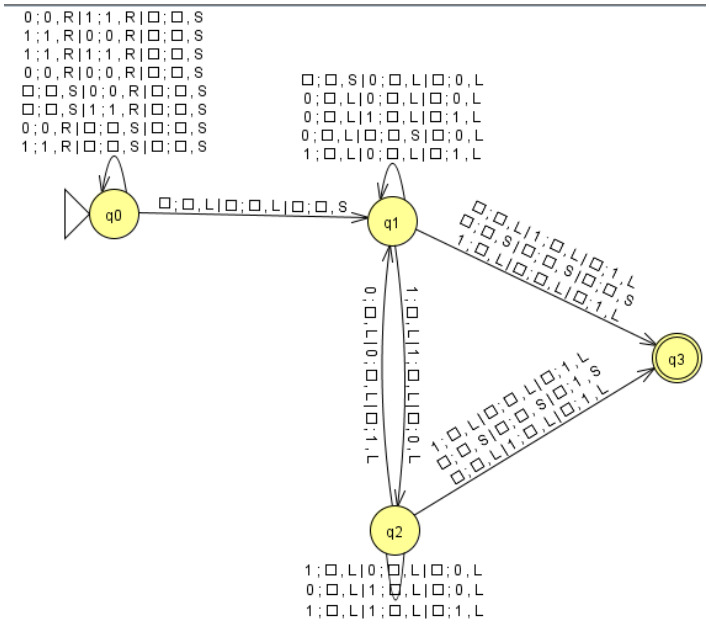
También se utilizará una función llamada *Gotoxy* para hacer la simulación de la cinta de transición dentro de la MT.

DIAGRAMA DE ESTADOS DE LA MÁQUINA DE TURING

(JFLAP)

A continuación, se muestra nuestro diagrama de estados, entradas y salidas, así como el movimiento del cabezal ya sea izquierdo o derecho.

Consta de 4 estados, en donde el inicial es el estado “q0” en donde se empezaran a introducir el numero binario y dependiendo de lo que se detecte se recorrerá a la izquierda llevándonos al estado “q1” donde se van detectando los números que prosiguen, pasando ya sea al estado “q2” para que se siga recorriendo los números o al “q3” donde se finaliza con la representación de la suma de los dos números en binario dados.



DEFINICIÓN FORMAL

(SÉPTUPLA)

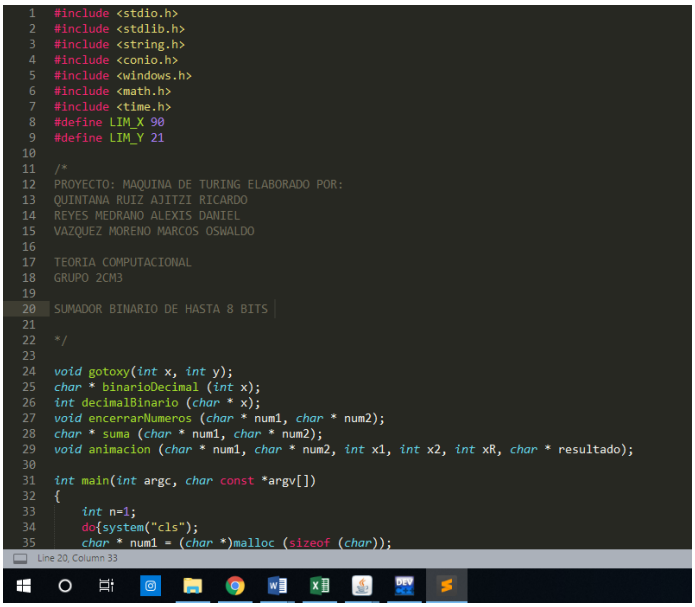
$$M= (\{q0,q1,q2,q3\}, \{0,1\}, \{0,1,B\}, p, q0, B, q3)$$

TABLA DE TRANSICIONES

Estado		Cinta 1			Cinta 2			Cinta 3	
	0	1	B	0	1	B	0	1	B
q0	(q0,0,R)	-	-	-	(q0,1,R)	-	-	-	(q0,B,S)
	-	(q0,1,R)	-	(q0,0,R)		-	-	-	(q0,B,S)
	-	(q0,1,R)	-	-	(q0,1,R)	-	-	-	(q0,B,S)
	(q0,0,R)		-	(q0,0,R)	-	-	-	-	(q0,B,S)
	-	-	(q0,B,S)	(q0,0,R)	-	-	-	-	(q0,B,S)
	-	-	(q0,B,S)	-	(q0,1,R)	-	-	-	(q0,B,S)
	(q0,0,R)	-	-	-	-	(q0,B,S)	-	-	(q0,B,S)
	-	(q0,1,R)	-	-	-	(q0,B,S)	-	-	(q0,B,S)
	-	-	(q1,B,L)	-	-	(q1,B,L)	-	-	(q1,B,L)
q1	(q1,B,L)	-	-	(q1,B,L)	-	-	-	-	(q1,0,L)
	(q1,B,L)			-	(q1,B,L)	-	-	-	(q1,1,L)
	-	-	(q1,B,S)	(q1,B,L)	-	-	-	-	(q1,0,L)
	(q1,B,L)	-	-	-	-	(q1,B,S)	-	-	(q1,0,L)
	-	(q1,B,L)	-	(q1,B,L)	-	-	-	-	(q1,1,L)
	-	(q2,B,L)	-	-	(q2,B,L)	-	-	-	(q2,0,L)
	-	-	(q3,B,L)	-	(q3,B,L)	-	-	-	(q3,1,L)
	-	-	(q3,B,S)	-	-	(q3,B,S)	-	-	(q3,B,S)
	-	(q3,B,L)	-	-	-	(q3,B,L)	-	-	(q3,1,L)
q2	-	(q2,B,L)	-	(q2,B,L)	-	-	-	-	(q2,0,L)
	(q2,B,L)	-	-	-	(q2,B,L)	-	-	-	(q2,0,L)
	-	(q2,B,L)	-	-	(q2,B,L)	-	-	-	(q2,1,L)
	(q1,B,L)	-	-	(q1,B,L)	-	-	-	-	(q1,1,L)
	-	(q3,B,L)	-	-	-	(q3,B,L)	-	-	(q3,1,L)
	-	-	(q3,B,L)	-	(q3,B,L)	-	-	-	(q3,1,L)
	-	-	(q3,B,S)	-	-	(q3,B,S)	-	-	(q3,1,S)

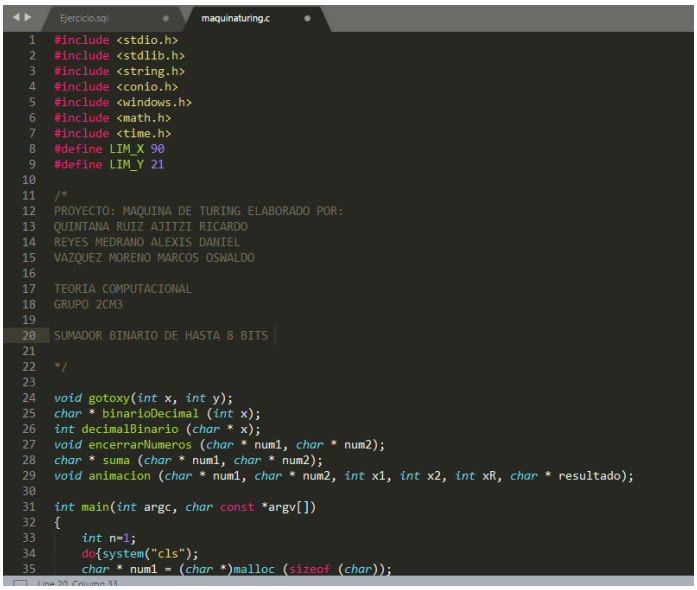
LENGUAJE DE PROGRAMACIÓN A UTILIZAR

Decidimos utilizar un lenguaje de programación cotidiano con el cual nos sentimos a gustos al trabajar, porque lo dominamos y sabemos que es uno de los que mejor se presta.



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <windows.h>
#include <math.h>
#include <time.h>
#define LIM_X 90
#define LIM_Y 21

/*
PROYECTO: MAQUINA DE TURING ELABORADO POR:
QUINTANA RUIZ AJITZI RICARDO
REYES MEDRANO ALEXIS DANIEL
VAZQUEZ MORENO MARCOS OSWALDO

TEORIA COMPUTACIONAL
GRUPO 2CM3

SUMADOR BINARIO DE HASTA 8 BITS

*/

void gotoxy(int x, int y);
char * binarioDecimal (int x);
int decimalBinario (char * x);
void encerrarNumeros (char * num1, char * num2);
char * suma (char * num1, char * num2);
void animacion (char * num1, char * num2, int x1, int x2, int xR, char * resultado);

int main(int argc, char const *argv[])
{
    int n=1;
    do{system("cls");
    char * num1 = (char *)malloc (sizeof (char));
```

Line 20, Column 33



Ejercicio.sql • maquinaturing.c •

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <windows.h>
#include <math.h>
#include <time.h>
#define LIM_X 90
#define LIM_Y 21

/*
PROYECTO: MAQUINA DE TURING ELABORADO POR:
QUINTANA RUIZ AJITZI RICARDO
REYES MEDRANO ALEXIS DANIEL
VAZQUEZ MORENO MARCOS OSWALDO

TEORIA COMPUTACIONAL
GRUPO 2CM3

SUMADOR BINARIO DE HASTA 8 BITS

*/

void gotoxy(int x, int y);
char * binarioDecimal (int x);
int decimalBinario (char * x);
void encerrarNumeros (char * num1, char * num2);
char * suma (char * num1, char * num2);
void animacion (char * num1, char * num2, int x1, int x2, int xR, char * resultado);

int main(int argc, char const *argv[])
{
    int n=1;
    do{system("cls");
    char * num1 = (char *)malloc (sizeof (char));
```

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
}

void encerrarNumeros (char * num1, char * num2)
{
    system("cls");
    char * res = (char *)malloc (sizeof (char));
    int longitudPalabra1, numGuiones1, x1, i;
    int longitudPalabra2, numGuiones2, x2;
    int longitudResultado, numGuionesRes, xR;
    longitudPalabra1 = strlen(num1);                    //Obtenemos la longitud de la cadena
    longitudPalabra2 = strlen(num2);                    //Obtenemos la longitud de la cadena 2
    numGuiones1 = (longitudPalabra1 * 2) + 1;           //Calculamos los guiones para simular la cinta
    numGuiones2 = (longitudPalabra2 * 2) + 1;           //Calculamos los guiones para simular la cinta 2
    x1 = (LIM_X - (longitudPalabra1 * 2)) / 2;          //Centramos la cadena
    x2 = (LIM_X - (longitudPalabra2 * 2)) / 2;          //Centramos la cadena 2
    if (longitudPalabra2 > longitudPalabra1)
    {
        x2 -= (longitudPalabra2 - longitudPalabra1);        //Alineamos con la cadena 1
        longitudResultado = (longitudPalabra2 + 1);
        numGuionesRes = (longitudResultado * 2) + 1;
        xR = (LIM_X - (longitudResultado * 2))/2;
        xR -= (longitudResultado - longitudPalabra1);
    }
    else
    {
        x1 -= (longitudPalabra1 - longitudPalabra2);
        longitudResultado = (longitudPalabra1 + 1);
        numGuionesRes = (longitudResultado * 2) + 1;
        xR = (LIM_X - (longitudResultado * 2))/2;
        xR -= (longitudResultado - longitudPalabra2);
    }
    //ENCERRAMOS EL PRIMER NÚMERO
    gotoxy(6,5);
    for (int i = 0; i < 79; ++i)
    {
```

Line 20, Column 33

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
    for (int i = 0; i < 79; ++i)
    {
        printf("%c",196);
    }
    gotoxy(6,6);
    for (int i = 0; i < 40; ++i)
    {
        printf("| ");
    }
    gotoxy(x1,6);
    for (i = 0; i < longitudPalabra1; i ++)
    {
        printf("|%c", num1[i]);
    }
    printf("|");
    gotoxy(6,7);
    for (int i = 0; i < 79; ++i)
    {
        printf("%c",196);
    }
    //ENCERRAMOS EL SEGUNDO NÚMERO
    gotoxy(6,10);
    for (int i = 0; i < 79; ++i)
    {
        printf("%c",196);
    }
    gotoxy(6,11);
    for (int i = 0; i < 40; ++i)
    {
        printf("| ");
    }
    gotoxy(x2,11);
    for (i = 0; i < longitudPalabra2; i ++)
    {
        printf("|%c", num2[i]);
```

Line 20, Column 33

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
130     {
131         printf("%c",196);
132     }
133     //ENCERRAMOS EL RESULTADO
134     gotoxy(6,15);
135     for (int i = 0; i < 79; ++i)
136     {
137         printf("%c",196);
138     }
139     gotoxy(6,16);
140     for (int i = 0; i < 40; ++i)
141     {
142         printf("| ");
143     }
144     gotoxy(xR,16);
145     for (i = 0; i < longitudResultado; i ++)
146     {
147         printf("| ");
148     }
149     printf("|");
150     gotoxy(6,17);
151     for (int i = 0; i < 79; ++i)
152     {
153         printf("%c",196);
154     }
155     //res = suma (num1, num2);
156     int numero1 = decimalBinario (num1);
157     int numero2 = decimalBinario (num2);
158     int result = numero1 + numero2;
159     res = binarioDecimal (result);
160     animacion(num1, num2, x1, x2, xR, res);
161 }
162
163 char * suma (char * num1, char * num2)
164 {
```

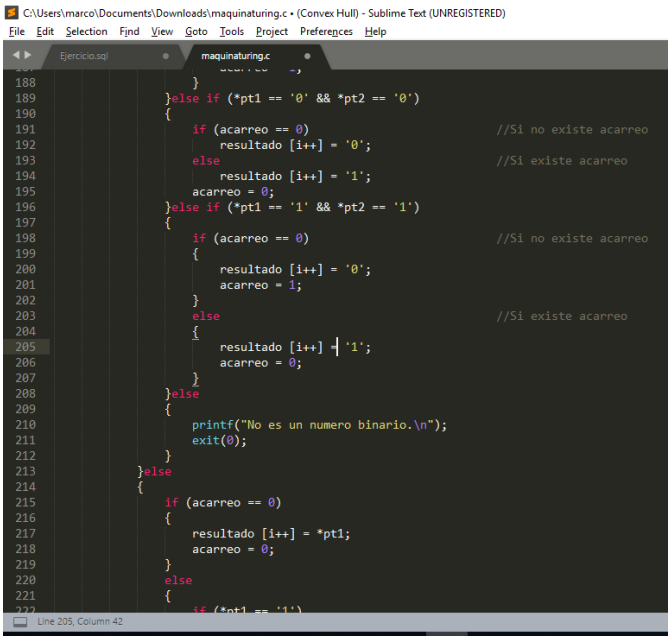
C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

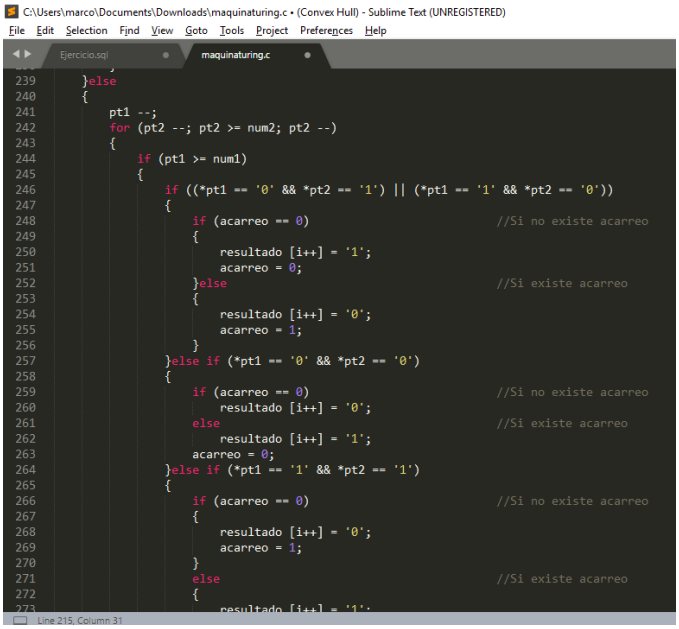
File Edit Selection Find View Goto Tools Project Preferences Help

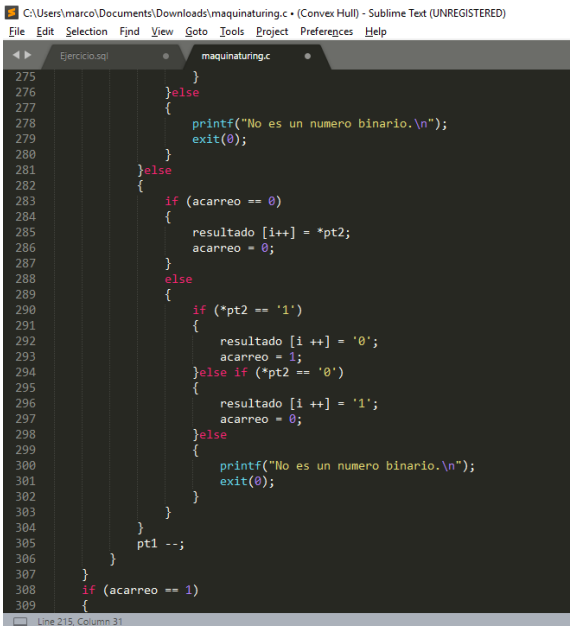
Ejercicio.sql | maquinaturing.c

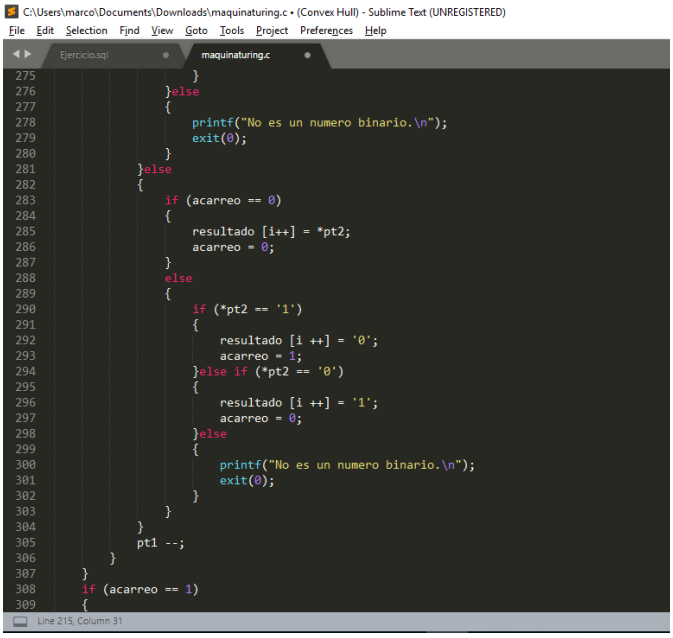
```
163 char * suma (char * num1, char * num2)
164 {
165     int acarreo = 0, i = 0;
166     char * resultado = (char *)malloc (sizeof (char));
167     char * pt1 = num1;
168     char * pt2 = num2;
169     for (; *pt1 != '\0'; pt1 ++);
170     for (; *pt2 != '\0'; pt2 ++);
171     if (strlen (num1) > strlen (num2))                //El número 1 es más grande
172     {
173         pt2--;
174         for (pt1 --; pt1 >= num1; pt1 --)
175         {
176             if (pt2 >= num2)
177             {
178                 if ((*pt1 == '0' && *pt2 == '1') || (*pt1 == '1' && *pt2 == '0'))
179                 {
180                     if (acarreo == 0)                          //Si no existe acarreo
181                     {
182                         resultado [i++] = '1';
183                         acarreo = 0;
184                     }else                                      //Si existe acarreo
185                     {
186                         resultado [i++] = '0';
187                         acarreo = 1;
188                     }
189                 }else if (*pt1 == '0' && *pt2 == '0')
190                 {
191                     if (acarreo == 0)                          //Si no existe acarreo
192                         resultado [i++] = '0';
193                     else                                       //Si existe acarreo
194                         resultado [i++] = '1';
195                     acarreo = 0;
196                 }else if (*pt1 == '1' && *pt2 == '1')
```

Line 20, Column 33


C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Ejercicio.sql
maquinaturing.c
}
}else if (*pt1 == '0' && *pt2 == '0')
{
if (acarreo == 0) //Si no existe acarreo
resultado [i++] = '0';
else //Si existe acarreo
resultado [i++] = '1';
acarreo = 0;
}else if (*pt1 == '1' && *pt2 == '1')
{
if (acarreo == 0) //Si no existe acarreo
{
resultado [i++] = '0';
acarreo = 1;
}
else //Si existe acarreo
{
resultado [i++] = '1';
acarreo = 0;
}
}else
{
printf("No es un numero binario.\n");
exit(0);
}
}else
{
if (acarreo == 0)
{
resultado [i++] = *pt1;
acarreo = 0;
}
else
{
Line 205, Column 42


C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Ejercicio.sql
maquinaturing.c
}else
{
pt1 --;
for (pt2 --; pt2 >= num2; pt2 --)
{
if (pt1 >= num1)
{
if ((*pt1 == '0' && *pt2 == '1') || (*pt1 == '1' && *pt2 == '0'))
{
if (acarreo == 0) //Si no existe acarreo
{
resultado [i++] = '1';
acarreo = 0;
}else //Si existe acarreo
{
resultado [i++] = '0';
acarreo = 1;
}
}else if (*pt1 == '0' && *pt2 == '0')
{
if (acarreo == 0) //Si no existe acarreo
resultado [i++] = '0';
else //Si existe acarreo
resultado [i++] = '1';
acarreo = 0;
}else if (*pt1 == '1' && *pt2 == '1')
{
if (acarreo == 0) //Si no existe acarreo
{
resultado [i++] = '0';
acarreo = 1;
}
else //Si existe acarreo
{
Line 215, Column 31


C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Ejercicio.sql
maquinaturing.c
}
}else
{
printf("No es un numero binario.\n");
exit(0);
}
}else
{
if (acarreo == 0)
{
resultado [i++] = *pt2;
acarreo = 0;
}
else
{
if (*pt2 == '1')
{
resultado [i ++] = '0';
acarreo = 1;
}else if (*pt2 == '0')
{
resultado [i ++] = '1';
acarreo = 0;
}else
{
printf("No es un numero binario.\n");
exit(0);
}
}
}
pt1 --;
}
}
if (acarreo == 1)
{
Line 215, Column 31


C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Ejercicio.sql
maquinaturing.c
}
}else
{
printf("No es un numero binario.\n");
exit(0);
}
}else
{
if (acarreo == 0)
{
resultado [i++] = *pt2;
acarreo = 0;
}
else
{
if (*pt2 == '1')
{
resultado [i ++] = '0';
acarreo = 1;
}else if (*pt2 == '0')
{
resultado [i ++] = '1';
acarreo = 0;
}else
{
printf("No es un numero binario.\n");
exit(0);
}
}
}
pt1 --;
}
}
if (acarreo == 1)
{
Line 215, Column 31

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
276                 }else
277                 {
278                     printf("No es un numero binario.\n");
279                     exit(0);
280                 }
281             }else
282             {
283                 if (acarreo == 0)
284                 {
285                     resultado [i++] = *pt2;
286                     acarreo = 0;
287                 }
288                 else
289                 {
290                     if (*pt2 == '1')
291                     {
292                         resultado [i ++] = '0';
293                         acarreo = 1;
294                     }else if (*pt2 == '0')
295                     {
296                         resultado [i ++] = '1';
297                         acarreo = 0;
298                     }else
299                     {
300                         printf("No es un numero binario.\n");
301                         exit(0);
302                     }
303                 }
304             }
305             pt1 --;
306         }
307     }
308     if (acarreo == 1)
309     {
310         resultado [i ++] = '1';
```

Line 295, Column 22

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
316 void animacion (char * num1, char * num2, int x1, int x2, int xR, char * resultado)
317 {
318     int j1 = (x1 + (strlen(num1) * 2) - 1);
319     int j2 = (x2 + (strlen (num2) * 2) - 1);
320     int j3 = (xR + (strlen(resultado) * 2) - 1);
321     char * pt1 = num1;
322     char * pt2 = num2;
323     char * pt3 = resultado;
324     int i = 0;
325     for (; *pt1 != '\0'; pt1 ++);
326     for (; *pt2 != '\0'; pt2 ++);
327     if (strlen (num1) > strlen (num2))
328     {
329         for (pt1 --; pt1 >= num1; pt1 --)
330         {
331             gotoxy (j1, 8);
332             printf ("%c", 94);
333             if (pt2 > num2)
334             {
335                 gotoxy (j2, 13);
336                 printf ("%c", 94);
337             }
338             gotoxy (j3, 18);
339             printf("%c\n", 94);
340             Sleep (700);
341             gotoxy (j1, 8);
342             printf (" ");
343             if (pt2 > num2)
344             {
345                 gotoxy (j2, 13);
346                 printf (" ");
347             }
348             gotoxy(j3, 18);
349             printf(" ");
350             gotoxy (j3, 16);
```

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
                gotoxy (j2, 11);
                printf(" ");
            }
            j1--;j1--;j2--;j2--;pt2--;j3--;j3--;pt3++;
        }
        if (*pt3 != '\0')
        {
            gotoxy (j3, 18);
            printf("%c\n", 94);
            Sleep (700);
            gotoxy (j3, 18);
            printf(" ");
            gotoxy (j3, 16);
            printf("%c", *pt3);
        }
    }else
    {
        for (pt2 --; pt2 >= num2; pt2 --)
        {
            gotoxy (j2, 13);
            printf ("%c", 94);
            if (pt1 > num1)
            {
                gotoxy (j1, 8);
                printf ("%c", 94);
            }
            gotoxy (j3, 18);
            printf("%c", 94);
            Sleep (700);
            gotoxy (j2, 13);
            printf (" ");
            if (pt1 > num1)
            {
                gotoxy (j1, 8);
                printf (" ");
```

Line 295, Column 22

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
        }
    }
}

char * binarioDecimal (int x)
{
    char * c = (char *) malloc (sizeof (char));
    int i = 0;
    while (x >= 1)
    {
        if (x % 2 == 1)
        {
            c [i++] = '1';
            x /= 2;
        }else
        {
            c [i++] = '0';
            x /= 2;
        }
    }
    c [i] = '\0';
    return c;
}

int decimalBinario (char * x)
{
    int numeroDecimal = 0;
    int flag = 0;
    char *pt = x, *ptr = x;
    for (; *pt != '\0'; pt++); pt--;                    //Mandamos un apuntador al final de la cadena, o a la posicion 2^0
    for (; pt >= ptr; pt--)
    {
        if (*pt == '1')
        {
            numeroDecimal += pow (2, flag);             //Si encuentra un 1, sumamos a la variable 2^flag
```

C:\Users\marco\Documents\Downloads\maquinaturing.c • (Convex Hull) - Sublime Text (UNREGISTERED)

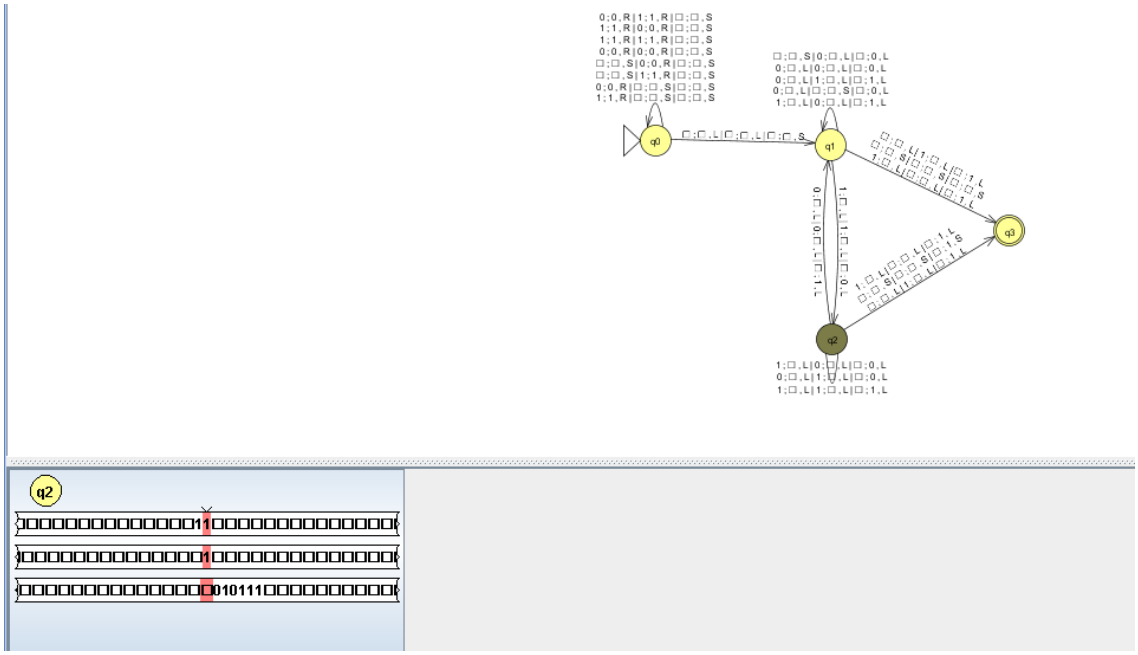
File Edit Selection Find View Goto Tools Project Preferences Help

Ejercicio.sql | maquinaturing.c

```
414         }
415     }
416 }
417
418 char * binarioDecimal (int x)
419 {
420     char * c = (char *) malloc (sizeof (char));
421     int i = 0;
422     while (x >= 1)
423     {
424         if (x % 2 == 1)
425         {
426             c [i++] = '1';
427             x /= 2;
428         }else
429         {
430             c [i++] = '0';
431             x /= 2;
432         }
433     }
434     c [i] = '\0';
435     return c;
436 }
437
438 int decimalBinario (char * x)
439 {
440     int numeroDecimal = 0;
441     int flag = 0;
442     char *pt = x, *ptr = x;
443     for (; *pt != '\0'; pt++); pt--;                    //Mandamos un apuntador al final de la cadena, o a la posicion 2^0
444     for (; pt >= ptr; pt--)
445     {
446         if (*pt == '1')
447         {
448             numeroDecimal += pow (2, flag);             //Si encuentra un 1, sumamos a la variable 2^flag
```

Line 295, Column 22

FUNCIONAMIENTO



CONCLUSIONES

- ✓ **Quintana** Considero teoría computacional como la unidad de aprendizaje que permite transferir todas las matemáticas a la realidad por medio de máquinas. Un gran aporte que nos dejá Alan Turing fue el hecho de que pudiéramos resolver cualquier algoritmo computable con una máquina. La máquina de Turing es el antecedente de lo que hoy llamamos computadoras, en la práctica final nos dimos cuenta de su poder computacional. Podemos considerar la función delta de la MT como el primer lenguaje de programación, vimos que las instrucciones que contiene la función son las que permiten comportarse de una u otra manera.

- ✓ **Reyes** La Máquina de Turing es el autómata más poderoso, pues a pesar de agregarle elementos (más cintas, pistas o no determinismo) no aumenta ni disminuye su capacidad computacional. Tuvimos la oportunidad de adaptar un algoritmo muy conocido, que es la conversión de decimal a binario, a una máquina de Turing determinista. De esta forma verificamos una vez más que la tesis de Church-Turing está lejos de ser falsa, pues todo algoritmo computable tiene su equivalente en una MT, claro, asumiendo que la memoria y el tiempo de ejecución tienden a infinito.

- ✓ **Vázquez** La máquina en sí es una herramienta demasiado importantes e impactantes, Finalmente quiero decir que Teoría Computacional es una materia muy interesante porque no solo se trata de programación y estructura de datos con arbole, pilas, etc., sino que también se utilizan matemáticas, tal es el caso de Matemáticas Discretas con los diagramas de estados y la toma de decisiones para brincar de un estado a otro u otros. Cabe mencionar que al igual que muchos de mis compañeros no tenía muy claro de qué iba a tratar la materia, pero poco a poco fui conociendo, aprendiendo y entendiendo bien lo que la profesora nos enseñaba.