

Prácticas de Programación Resueltas: Máquinas de Turing

1. Realizar un programa que simule la actuación de una máquina de Turing que al serle introducida una sucesión de `1` escritos en casillas consecutivas en el estado de partida e_0 con la cabeza lectora-inscriptora sobre el cualquiera de los 1 de la sucesión, devuelva otra sucesión de `1` que tenga el triple de `1` que la de partida.

Solución. A la hora de diseñar el programa que nos piden nos damos cuenta que:

- Los datos de la cinta los podemos guardar en un array de tipo carácter.
- Para que nos aparezca el estado debajo de la casilla con el símbolo que se está examinando, crearemos otro array de tipo carácter que tenga en sus entradas ' ' y en la posición que nos interese el índice del estado que corresponda.
- Podemos crear unas subrutinas que simulen la acción de los cuádruples. De esta forma será más rápido realizar la programación.
- Será necesario diseñar los cuádruples que se van a utilizar. Esto es, se creará primero la máquina de Turing teórica y luego se implementará.

Un programa escrito en C que simula la actuación de la máquina es:

```
#include <stdio.h>
void cuadruple_cambio(char cinta[],char estado[],int posicion,char nuevo_estado,char
nuevo_simbolo)
{
    int i;
    char c;
    cinta[posicion]=nuevo_simbolo;
    estado[posicion]=nuevo_estado;
    printf("\n\n\n");
    for(i=0;i<=79;i++) printf("%c",cinta[i]);
    printf("\n");
    for(i=0;i<=79;i++) printf("%c",estado[i]);
}

int cuadruple_dra(char cinta[],char estado[], int posicion,char nuevo_estado)
{
    int i,d;
    estado[posicion]=' ';
    estado[posicion+1]=nuevo_estado;
    printf("\n\n\n");
    for(i=0;i<=79;i++) printf("%c",cinta[i]);
    printf("\n");
    for(i=0;i<=79;i++) printf("%c",estado[i]);
}
```

```

posicion=posicion+1;

return(posicion);
}

int cuadruple_izda(char cinta[],char estado[], int posicion,char nuevo_estado)
{
    int i,d;
    estado[posicion]=' ';
    estado[posicion-1]=nuevo_estado;
    printf("\n\n\n");
    for(i=0;i<=79;i++) printf("%c",cinta[i]);
    printf("\n");
    for(i=0;i<=79;i++) printf("%c",estado[i]);
    posicion=posicion-1;

    return(posicion);
}

void main(void)
{
    char cinta[80], estado[80],nuevo_simb, nuevo_est, est_partida, resp;
    int posicion,marcha,n,i,contador;
    /* La variable marcha es una variable auxiliar que toma el valor 1 para indicar que la maquina
    va a seguir actuando y 0 cuando se para. Se inicializa con el valor 1. */
    printf("\n Programa que simula la actuacion de una maquina");
    printf(" de Turing que dada una sucesion de 1 devuelve otra");
    printf(" con el triple de unos que la inicial. Las condiciones");
    printf(" iniciales para que funcione correctamente son: ");
    printf("\n Estado de partida: 0");
    printf("\n Cabeza lectora-inscriptora: Sobre una casilla no vacia");
    printf("\n\n Por problemas de espacio, se limita el numero maximo");
    printf(" de 1 a triplicar a 25");
    do
    {
        for(i=0;i<=79;i++)
        {
            cinta[i]=' ';
            estado[i]=' ';
        }
        printf("\n\n Introduce el numero que deseas triplicar. \n n=");
        scanf("%d", &n);
        while(n<1 || n>25)
        {
            printf("\n Numero no valido. Debe estar entre 1 y 25.");
            printf("\n Introduzca de nuevo. \n n=");
            scanf("%d",&n);
        }
        printf("\n\n Introduce el estado de partida.\n estado de partida=");
        scanf("%c",&est_partida);
        scanf("%c",&est_partida);
        printf("\n\n Introduce la posicion inicial\n posicion inicial=");

```

```

scanf("%d",&posicion);
while(posicion<0 || posicion>79)
{
    printf("\n Posicion no valida. Debe estar entre 0 y 79.");
    printf("\n Introduzca nueva posicion inicial.\n posicion inicial=");
    scanf("%d",&posicion);
}
for (i=1;i<=n;i++) cinta[i]='1';
estado[posicion]=est_partida;
printf("\n Inicializacion de la maquina\n\n");
for(i=0;i<=79;i++) printf("%c",cinta[i]);
printf("\n");
for(i=0;i<=79;i++) printf("%c",estado[i]);
contador=0;
do
{
    marcha=0;
    if (cinta[posicion]=='1' && estado[posicion]=='0')
    {
        marcha=1;
        nuevo_est='0';
        contador=contador+1;
        posicion=cuadruple_izda(cinta,estado,posicion,nuevo_est);
    } /* Cuadruple e0 1 l e0 */
    if(cinta[posicion]==' ' && estado[posicion]=='0')
    {
        marcha=1;
        nuevo_est='1';
        contador=contador+1;
        posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
    } /* Cuadruple e0 s0 D e1 */
    if(cinta[posicion]=='1' && estado[posicion]=='1')
    {
        marcha=1;
        nuevo_simb='a';
        nuevo_est='2';
        contador=contador+1;
        cuadruple_cambio(cinta,estado,posicion,nuevo_est,nuevo_simb);
    } /* Cuadruple e1 1 a e2 */
    if(cinta[posicion]=='a' && estado[posicion]=='2')
    {
        marcha=1;
        nuevo_est='2';
        contador=contador+1;
        posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
    } /* Cuadruple e2 a D e2 */
    if(cinta[posicion]=='1' && estado[posicion]=='2')
    {
        marcha=1;
        nuevo_est='2';
        contador=contador+1;
        posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
    }
}

```

```

}/* Cuadruple e2 1 D e2*/
if(cinta[posicion]=='b'&&estado[posicion]=='2')
{
    marcha=1;
    nuevo_est='2';
    contador=contador+1;
    posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e2 b D e2*/
if(cinta[posicion]==' '&&estado[posicion]=='2')
{
    marcha=1;
    nuevo_simb='b';
    nuevo_est='3';
    contador=contador+1;
    cuadruple_cambio(cinta,estado,posicion,nuevo_est,nuevo_simb);
}/* Cuadruple e2 s0 b e3*/
if(cinta[posicion]=='b'&&estado[posicion]=='3')
{
    marcha=1;
    nuevo_est='4';
    contador=contador+1;
    posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e3 b D e4*/
if(cinta[posicion]==' '&&estado[posicion]=='4')
{
    marcha=1;
    nuevo_simb='b';
    nuevo_est='5';
    contador=contador+1;
    cuadruple_cambio(cinta,estado,posicion,nuevo_est,nuevo_simb);
}/* Cuadruple e4 s0 b e5*/
if (cinta[posicion]=='b'&&estado[posicion]=='5')
{
    marcha=1;
    nuevo_est='5';
    contador=contador+1;
    posicion=cuadruple_izda(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e5 b l e5*/
if (cinta[posicion]=='1'&&estado[posicion]=='5')
{
    marcha=1;
    nuevo_est='5';
    contador=contador+1;
    posicion=cuadruple_izda(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e5 1 l e5*/
if (cinta[posicion]=='a'&&estado[posicion]=='5')
{
    marcha=1;
    nuevo_est='1';
    contador=contador+1;
    posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e5 a D e1*/

```

```

if (cinta[posicion]=='b'&&estado[posicion]=='1')
{
    marcha=1;
    nuevo_est='6';
    contador=contador+1;
    posicion=cuadruple_izda(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e1 b l e6*/
if (cinta[posicion]=='a'&&estado[posicion]=='6')
{
    marcha=1;
    nuevo_est='6';
    contador=contador+1;
    posicion=cuadruple_izda(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e6 a l e6*/
if (cinta[posicion]==' '&&estado[posicion]=='6')
{
    marcha=1;
    nuevo_est='7';
    contador=contador+1;
    posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e6 s0 D e7*/
if(cinta[posicion]=='a'&&estado[posicion]=='7')
{
    marcha=1;
    nuevo_simb='1';
    contador=contador+1;
    nuevo_est='7';
    cuadruple_cambio(cinta,estado,posicion,nuevo_est,nuevo_simb);
}/* Cuadruple e7 a 1 e7*/
if (cinta[posicion]=='1'&&estado[posicion]=='7')
{
    marcha=1;
    nuevo_est='7';
    contador=contador+1;
    posicion=cuadruple_dra(cinta,estado,posicion,nuevo_est);
}/* Cuadruple e7 1 D e7*/
if(cinta[posicion]=='b'&&estado[posicion]=='7')
{
    marcha=1;
    nuevo_simb='1';
    contador=contador+1;
    nuevo_est='7';
    cuadruple_cambio(cinta,estado,posicion,nuevo_est,nuevo_simb);
}/* Cuadruple e7 b 1 e7*/
}while(marcha==1);
printf("\n \n\n Fin de la actuacion de la maquina.");
printf("\n La maquina ha realizado %d movimientos.",contador);
printf("\n\n Deseas repetir el proceso?(s/n): ");
scanf("%c",&resp);
scanf("%c",&resp);
}while(resp=='s' | resp=='S');
}

```