**EJERCICIO EJEMPLO PARA EL TP SOBRE AUTÓMATA RECONOCEDOR/ACCIONADOR**

Sea una secuencia de caracteres formada por letras minúsculas del alfabeto inglés y/o dígitos del 0 al 9. Esta secuencia contiene diferentes códigos de artículos, separados uno del otro por el carácter especial #. Este texto termina con el carácter especial fdt.

🡺 Se desea mostrar por pantalla cada uno de los códigos detectados con la descripción del artículo al cual

pertenece, en función de las siguientes características:

\* comienza con una letra minúscula entre las letras **c** y **h** seguida de tres dígitos entre **3** y **5**, seguido

por la letra **z** 🡪 artículo *zapato-vestir*

\* comienza con una letra minúscula **a**, **b**, **w** seguida de dos dígitos entre **7** y **9**, seguido por la letra

**z** 🡪 artículo *zapato-deportivo*

Ejemplos: Son códigos válidos **c334z d343z f555z h345z**

**a87z a89z b77z w97z**

Aclaración: Puede haber varios **#** seguidos, empezar con uno o varios **#** y siempre termina con **#fdt**. El carácter especial **#** del final, es para indicar la finalización del último código.

Pueden venir códigos que no sean los buscados.

Ejemplo de un texto:

##d435z#b77z###a74z#c453##h345z##w32z#j877z#w97z#6454##fdt

Por pantalla debe mostrarse:

d435z zapato-vestir

b77z zapato-deportivo

c453z zapato-vestir

h345z zapato-vestir

w97z zapato-deportivo

1° Es necesarios determinar las categorías para cada columna de la Tabla de Transiciones, que son los caracteres que pueden provocar el cambio de un estado a otro o el quedarse en el mismo estado, a medida que va reconociendo un código. Es importante tener en cuenta que no puede estar en más de una categoría una misma letra o dígito.

Entonces, las categorías son:

**Lch** [c-h]

**D35** [3-5]

**Labw** [abw]

**D79** [7-9]

**fdt**

**#**

**Otros** [abi-vyz0-26]

**Zeta** z

Tengan en cuenta que en este tipo de AFDs no pueden hallar el mínimo. En este caso, en particular, tienen los estados 6 y 11 que deben indicar acciones diferentes: el estado 6 🡺 corresponde a zapatos-vestir; mientras que el estado 11 🡺 corresponde a zapatos-deportivos.

De los estados 6, 11 y 12 se puede volver a los estados: 1 (si el carácter corresponde a la categoría Lch) y 7 (si el carácter corresponde a la categoría Labw). Nunca se mezclan los dos tipos de códigos posibles.

Del estado 13 solo se puede salir cuando lea un # (que va al estado 12)

2° Armar la Tabla de transiciones completa, excepto la columna del fdt.

La Tabla de Transiciones es:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Lch** | **D35** | **Labw** | **D79** | **Zeta** | **#** | **Otros** | **fdt** | **Acciones** |
| 0- | 1 | 13 | 7 | 13 | 13 | 12 | 13 | - | Inicializar **i** |
| 1 | 13 | 2 | 13 | 13 | 13 | 13 | 13 | - | Guardar en buffer el carácter leído, incrementa **i** |
| 2 | 13 | 3 | 13 | 13 | 13 | 13 | 13 | - | Idem anterior |
| 3 | 13 | 4 | 13 | 13 | 13 | 13 | 13 | - | Idem anterior |
| 4 | 13 | 13 | 13 | 13 | 5 | 13 | 13 | - | Idem anterior |
| 5 | 13 | 13 | 13 | 13 | 13 | 6 | 13 | - | Idem anterior |
| 6 | 1 | 13 | 7 | 13 | 13 | 12 | 13 | 14 | Agregar el carácter nulo en buffer, muestra los resultados e inicializa **i** |
| 7 | 13 | 13 | 13 | 8 | 13 | 13 | 13 | - | Guarda en buffer el carácter leído, incrementa **i** |
| 8 | 13 | 13 | 13 | 9 | 13 | 13 | 13 | - | Idem anterior |
| 9 | 13 | 13 | 13 | 13 | 10 | 13 | 13 | - | Idem anterior |
| 10 | 13 | 13 | 13 | 13 | 13 | 11 | 13 | - | Idem anterior |
| 11 | 1 | 13 | 7 | 13 | 13 | 12 | 13 | 14 | Agregar el carácter nulo en buffer, muestra los resultados e inicializa **i** |
| 12 | 1 | 13 | 7 | 13 | 13 | 12 | 13 | 14 |  |
| 13 | 13 | 13 | 13 | 13 | 13 | 12 | 13 | 14 |  |
| 14+ | - | - | - | - | - | - | - | - |  |

Antes de escribir la Automata, desarrollo la función auxiliar Columna. Recuerden que esta función es sumamente importante porque determina la columna que le corresponde a cada carácter leído.

int Columna (char x) {

if (x>=’c’ && x<=’h’) return 0;

if (x>=’3’ && x<=’5’) return 1;

if (x>=’7’ && x<=’9’) return 3;

switch(x) {

case ‘a’: case ‘b’: case ‘w’: return 2; break;

case ‘z’: return 4; break;

case ‘#’: return 5; break;

}

return 6;

} /\* fin-funcion-Columna \*/

Aclaración: la columna que corresponde al carácter especial fdt, no se completa ni se agrega en la Tabla de la función Automata ya que nunca entra al while.

La variable buffer es una variable auxiliar donde se van guardando los caracteres a medida que se leen del texto para poder tener el código y mostrarlo por pantalla. Luego de hacerlo se debe volver a inicializar la variable i que se usa como subíndice para buffer.

int Automata (char \*texto) {

static Tabla[14][7] = {{1,13,7,13,13,12,13}, {13,2,13,13,13,13,13}, {13,3,13,13,13,13,13},

{13,4,13,13,13,13,13}, {13,13,13,13,5,13,13}, {13,13,13,13,13,6,13},

{1,13,7,13,13,12,13}, {13,8,13,13,13,13,13}, {13,9,13,13,13,13,13},

{13,13,13,13,10,13,13}, {13,13,13,13,13,11,13}, {1,13,7,13,13,12,13},

{1,13,7,13,13,12,13}, {13,13,13,13,13,12,13}};

char buffer[100+1];

int i = 0, j = 0;

char fdt= ‘\0’;

estadoActual = 0;

while (texto[j] != fdt) {

estadoActual = Tabla[estadoActual][Columna(texto[j])];

switch (estadoActual) {

case 1: case 2: case 3: case 4: case 5: case 7:case 8: case 9: case 10:

buffer[i++] = texto[j]; break;

case 6: buffer[i] = ‘\0’; printf(“%s zapato-vestir\n”, buffer); i=0; break;

case 11: buffer[i] = ‘\0’; printf(“%s zapato-deportivo\n”,buffer); i=0; break;

}

j++;

}

if (texto[j] == fdt) return 1;

return 0;

} /\* fin-funcion-Automata \*/