Tablas de Transiciones

Constante Entera y Constante Real

**D0** [0]  
**D0c** [1-7]  
**D** [8-9]  
**Lx** [x X]  
**Lu** [u U]  
**Ll** [l L]  
**Le** [e E]  
**Lf** [f F]  
**LH** [a-d A-D]  
**PD** [.]  
**S** [\- \+]  
**#** [; , ( )]  
**Otro**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **D0** | **D0c** | **D** | **Lx** | **Lu** | **Ll** | **Le** | **Lf** | **LH** | **PD** | **S** | **#** | **Otro** | **Acciones** |
| **0** | 3 | 1 | 1 | - | - | - | - | - | - | 14 | - | - | - | Inicializar **j** |
| **1** | 1 | 1 | 1 | - | 2 | 4 | 11 | - | - | 10 | - | 8 | 8 | Guardar en buffer el carácter leído, incrementa **j** |
| **2** | - | - | - | - | - | 9 | - | - | - | - | - | 8 | 8 | Guardar en buffer el carácter leído, incrementa **j** |
| **3** | 6 | 6 | 15 | 5 | 2 | 4 | 11 | - | - | - | - | 8 | 8 | Guardar en buffer el carácter leído, incrementa **j** |
| **4** | - | - | - | - | 9 | - | - | - | - | - | - | 8 | 8 | Guardar en buffer el carácter leído, incrementa **j** |
| **5** | 7 | 7 | 7 | - | - | - | 7 | 7 | 7 | - | - | - | - | Guardar en buffer el carácter leído, incrementa **j** |
| **6** | 6 | 6 | 15 | - | 2 | 4 | 11 | - | - | 10 | - | 8 | 8 | Guardar en buffer el carácter leído, incrementa **j** |
| **7** | 7 | 7 | 7 | - | 2 | 4 | 7 | 7 | 7 | - | - | 8 | 8 | Guardar en buffer el carácter leído, incrementa **j** |
| **8** | - | - | - | - | - | - | - | - | - | - | - | - | - | Agregar el carácter nulo en buffer, muestra los resultados e inicializa **j** |
| **9** | - | - | - | - | - | - | - | - | - | - | - | - | - | Guardar en buffer el carácter leído, incrementa j, agrega el carácter nulo en buffer, muestra los resultados e inicializa **j** |
| **10** | 10 | 10 | 10 | - | - | 17 | 11 | 17 | - | - | - | - | 16 | Guardar en buffer el carácter leído, incrementa **j** |
| **11** | 12 | 12 | 12 | - | - | - | - | - | - | - | 13 | - | - | Guardar en buffer el carácter leído, incrementa **j** |
| **12** | 12 | 12 | 12 | - | - | 17 | - | 17 | - | - | - | - | 16 | Guardar en buffer el carácter leído, incrementa **j** |
| **13** | 12 | 12 | 12 | - | - | - | - | - | - | - | - | - | - | Guardar en buffer el carácter leído, incrementa **j** |
| **14** | 10 | 10 | 10 | - | - | - | - | - | - | - | - | - | - | Guardar en buffer el carácter leído, incrementa **j** |
| **15** | 15 | 15 | 15 | - | - | - | 11 | - | - | 10 | - | - | - | Guardar en buffer el carácter leído, incrementa **j** |
| **16** | - | - | - | - | - | - | - | - | - | - | - | - | - | Agregar el carácter nulo en buffer, muestra los resultados e inicializa **j** |
| **17** | - | - | - | - | - | - | - | - | - | - | - | - | - | Guardar en buffer el carácter leído, incrementa j, agrega el carácter nulo en buffer, muestra los resultados e inicializa **j** |

Identificador

**D0** [0]  
**D0c** [1-7]  
**D** [8-9]  
**Lx** [x X]  
**Lu** [u U]  
**Ll** [l L]  
**Le** [e E]  
**Lf** [f F]  
**LH** [a-d A-D]  
**PD** [.]  
**S** [\- \+]  
**Gb** [\_]  
**L1** [g-k G-K]  
**L2** [m-t M-T]  
**L3** [v V w W]  
**L4** [y Y z Z]  
**#** [; , ( )]  
**Otro**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **D0** | **D0c** | **D** | **Lx** | **Lu** | **Ll** | **Le** | **Lf** | **LH** | **PD** | **S** | **Gb** | **L1** | **L2** | **L3** | **L4** | **#** | **Otro** | **Acciones** |
| **0** | - | - | - | 1 | 1 | 1 | 1 | 1 | 1 | - | - | 1 | 1 | 1 | 1 | 1 | - | - | Inicializar **j** |
| **1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | Guardar en buffer el carácter leído,  incrementa **j** |
| **2** | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | Agregar el carácter nulo en buffer, muestra los resultados e inicializa **j** |

PuntoyComa, Coma, ParenDerecho, ParenIzquierdo y Otros

**#** [; , ( )]  
**Otro**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **#** | **Otro** | **Acciones** |
| **0** | 1 | 1 | Comparar caractér |
| **1** | - | - | Mostrar los resultados |

Función Autómata

int ColumnaEyR (char x)

{

if(x=='0') return **0**;

if (x>='1' && x<='7') return **1**;

if (x>='8' && x<='9') return **2**;

if (x>='a' && x<='d') return **8**;

if (x>='A' && x<='D') return **8**;

if (x=='.') return **9**;

if (x=='\-' || x<='\+') return **10**;

switch(x)

{

case 'x': case 'X': return **3**; break;

case 'u': case 'U': return **4**; break;

case 'l': case 'L': return **5**; break;

case 'e': case 'E': return **6**; break;

case 'f': case 'F': return **7**; break;

case ';': case ',': case '(': case ')': return **11**; break;

default: return **12**; break;

}

}

int ColumnaI (char x)

{

if(x=='0') return **0**;

if (x>='1' && x<='7') return **1**;

if (x>='8' && x<='9') return **2**;

if (x>='a' && x<='d') return **8**;

if (x>='A' && x<='D') return **8**;

if (x=='.') return **9**;

if (x=='\-' || x<='\+') return **10**;

if (x=='\_') return **11**;

if (x>='g' && x<='k') return **12**;

if (x>='G' && x<='K') return **12**;

if (x>='m' && x<='t') return **13**;

if (x>='M' && x<='T') return **13**;

if (x>='v' && x<='w') return **14**;

if (x>='V' && x<='W') return **14**;

if (x>='y' && x<='z') return **15**;

if (x>='Y' && x<='Z') return **15**;

switch(x)

{

case 'x': case 'X': return **3**; break;

case 'u': case 'U': return **4**; break;

case 'l': case 'L': return **5**; break;

case 'e': case 'E': return **6**; break;

case 'f': case 'F': return **7**; break;

case ';': case ',': case '(': case ')': return **16**; break;

default: return **17**; break;

}

}

automataOtro(char texto)

{

switch(texto)

{

case ';': printf(*"%cPunto y Coma\n"*, texto); break;

case ',': printf(*"%cComa\n"*, texto); break;

case '(': printf(*"%cParentesis Derecho\n"*, texto); break;

case ')': printf(*"%cParentesis Izquierdo\n"*, texto); break;

default: printf(*"%cOtro\n"*,texto); break;

}

return ++i;

}

automataEyR(char \* texto, int i)

{

static Tabla[**16**][**12**] = {

{**3,1,1,0,0,0,0,0,0,14,0,0,0**},

{**1,1,1,0,2,4,11,0,0,10,0,8,8**},

{**0,0,0,0,0,9,0,0,0,0,0,8,8**},

{**6,6,15,5,2,4,11,0,0,0,0,8,8**},

{**0,0,0,0,9,0,0,0,0,0,0,8,8**},

{**7,7,7,0,0,0,7,7,7,0,0,0,0**},

{**6,6,15,0,2,4,11,0,0,10,0,8,8**},

{**7,7,7,0,2,4,7,7,7,0,0,8,8**},

{**0,0,0,0,0,0,0,0,0,0,0,0,0**},

{**0,0,0,0,0,0,0,0,0,0,0,0,0**},

{**10,10,10,0,0,17,11,17,0,0,0,0,16**},

{**12,12,12,0,0,0,0,0,0,0,13,0,0**},

{**12,12,12,0,0,17,0,17,0,0,0,0,16**},

{**12,12,12,0,0,0,0,0,0,0,0,0,0**},

{**10,10,10,0,0,0,0,0,0,0,0,0,0**},

{**15,15,15,0,0,0,11,0,0,10,0,0,0**},

{**0,0,0,0,0,0,0,0,0,0,0,0,0**},

{**0,0,0,0,0,0,0,0,0,0,0,0,0**}

};

int j=**0**;char buffer[**100**+**1**];bool flag = true;int estadoActual = **0**;

while(flag)

{

estadoActual = Tabla[estadoActual][ColumnaEyR(texto[i])];

switch(estadoActual)

{

case **9**: buffer[j++] = texto[i];

case **8**: buffer[j] = '\0';printf(*"%sConstante Entera\n"*,buffer );j=**0**;flag=false; break;

case **17**: buffer[j++] = texto[i];

case **16**: buffer[j] = '\0';printf(*"%sConstante Real\n"*,buffer ); j=**0**; flag=false; break;

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

}

i++;

}

return --i;

}

automataI(char \* texto, int i)  
{

static Tabla[**2**][**15**] = {

{**0,0,0,1,1,1,1,1,1,1,1,1,1,1,0,0**},

{**1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2**},

{**0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0**}

};

int j=**0**;char buffer[**100**+**1**];bool flag = true;int estadoActual = **0**;

while(flag)

{

estadoActual = Tabla[estadoActual][ColumnaI(texto[i])];

switch(estadoActual)

{

case **2**: buffer[j] = '\0';printf(*"%sIdentificador\n"*,buffer );j=**0**;flag=false;break;

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

}

i++;

}

return --i;

}

automata (char \* texto)

{

int i = **0**;char fdt ='\0';

while(texto[i]!=fdt)

{

if(isDigit(texto[i]) || texto[i]=='.'){

i = automataEyR(texto,i);

}else{

if(isAlpha(texto[i]) || texto[i]=='\_'){

i = automataI(texto,i);

}else{

i = automataOtro(texto[i]);

}

}

}

}