

Para empezar, hicimos uso de un esqueleto para el TP brindado por la profesora **Leituz** para poder empezar a trabajar, el mismo consiste de dos “ifs”

```
if(!verifica(palabra)) {  
    printf("Error lexico :(\n");  
    return 0;  
}  
  
if(esPalabra(palabra, &contOctal, &contDecimal, &contHexa)) {  
    printf("La cadena pertenece al lenguaje! :)\n\n");  
    printf("Cantidad de octales: %d\n", contOctal);  
    printf("Cantidad de decimales: %d\n", contDecimal);  
    printf("Cantidad de hexadecimales: %d\n\n", contHexa);  
}  
else {  
    printf("La cadena insertada no pertenece al lenguaje :(\n");  
}  
  
return 0;  
}
```

Adentro de la función **esPalabra** creamos la matriz de transición

```
int esPalabra(char *cadena, int* contOctal, int* contDecimal, int* contHexa)
{
    const int tt[8][21] = {
        {1,1,7,7,3,2,2,2,2,2,2,2,2,7,7,7,7,7,7,7,7},
        {7,7,7,7,7,2,2,2,2,2,2,2,2,7,7,7,7,7,7,7,7},
        {7,7,7,7,2,2,2,2,2,2,2,2,2,7,7,7,7,7,7,0,0},
        {7,7,4,4,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7},
        {7,7,7,7,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,7},
        {7,7,7,7,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,0},
        {7,7,7,7,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,0},
        {7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7}
    };

    int estado = 0;
    int i = 0;
    int c = cadena[i];
```

El **if** que se ve adentro evalúa si el estado es final, y si es final, se fija que el siguiente carácter a recorrer en la cadena sea un **barra cero** o un **#**, cualquiera de estos dos casos, indica que debe incrementarse el contador a la base correspondiente

```
while(c != '\0' && estado != 7)
{
    estado = tt[estado][columna(c)];

    if(esEstadoFinal(estado) && (cadena[i + 1] == '\0' || cadena[i + 1] == '#'))
    {
        incrementarContadores(estado, contOctal, contDecimal, contHexa);
    }

    c = cadena[++i];
}
```

La función **incrementarContadores** se fija según el estado final, cualquier el contador de la base que debe incrementar

```
void incrementarContadores(int estado, int* contOctal, int* contDecimal, int* contHexa)
{
    switch (estado)
    {
        case 6:
            ++(*contOctal);
            break;
        case 2:
            ++(*contDecimal);
            break;
        case 5:
            ++(*contHexa);
            break;
    }
}
```

En la función **columna** se determina cual es la columna correspondiente en base al carácter leído.

Acá tuvimos que hacer uso de todos los números, pero para hacer la calculadora, los números que vayan del 1 al 9 simplemente los agruparemos como **D**, de esta forma podremos achicar la matriz de transición.

```
int columna(int c)
{
    switch(c)
    {
        case '+': return 0;
        case '-': return 1;
        case 'x': return 2;
        case 'X': return 3;
        case '0': return 4;
        case '1': return 5;
        case '2': return 6;
        case '3': return 7;
        case '4': return 8;
        case '5': return 9;
        case '6': return 10;
        case '7': return 11;
        case '8': return 12;
        case '9': return 13;
        case 'A': return 14;
        case 'B': return 15;
        case 'C': return 16;
        case 'D': return 17;
        case 'E': return 18;
        case 'F': return 19;
        case '#': return 20;
    }
}
```