

De ser correcta la cadena ingresada, dado que la operación, tal y como la reconoce el autómata, está en notación infija, se procede a convertirla a notación postfija hicimos uso del algoritmo de **shunting yard**.

La notación postfija tiene la ventaja de ser más fácil de evaluar usando una pila, pudiendo apilar operandos y aplicar operadores cuando aparezcan.

Finalmente, al tener la cadena en notación postfija, se procede a calcular su resultado, convirtiendo sus caracteres en enteros.

Tanto como para convertir de notación infija a postfija, como para calcular el resultado de la operación, se hizo uso de funciones conseguidas de Internet.

**(Para convertir infija en postfija)**

<https://www.geeksforgeeks.org/convert-infix-expression-to-postfix-expression/>

**(Para conseguir el resultado de una operación en notación postfija)**

<https://www.geeksforgeeks.org/evaluation-of-postfix-expression/>

Para las columnas, los números del 1 al 9 se agruparon bajo la letra **D**

```
int columna(int c)
{
    switch(c)
    {
        case '0': return 0;
        case '1': return 1;
        case '2': return 1;
        case '3': return 1;
        case '4': return 1;
        case '5': return 1;
        case '6': return 1;
        case '7': return 1;
        case '8': return 1;
        case '9': return 1;
        case '+': return 2;
        case '-': return 3;
        case '*': return 4;
        case '/': return 5;
    }
}
```

Consiguiendo así una reducción notable de la matriz de transición

```
int esPalabra(char *cadena)
{
    const int tt[6][6] = {
        {2,2,1,1,5,5},
        {2,2,5,5,5,5},
        {5,5,1,1,1,3},
        {5,4,5,5,5,5},
        {5,5,1,1,1,3},
        {5,5,5,5,5,5}
    };
}
```

Modularizando, “**convertirNotacion.h**” almacena toda la lógica para poder convertir de infija a posfija, y “**resultadoPosfija.h**” guarda toda la lógica para poder pasar de postfija al resultado de la operación

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#include "convertirNotacion.h"
#include "resultadoPosfija.h"
```

**infixToPostfix( )** convierte la cadena ingresada de notación infija a posfija

**evaluatePostfix( )** evalúa la notación posfija y obtiene su resultado

```
if (!verifica(operacion)) {

printf("Error lexico :(\n");

return 0;
}

if (esPalabra(operacion)) {
printf("\nLa cadena pertenece al lenguaje! :)\n\n");

char* notacionPosfija = infixToPostfix(operacion);

printf("La notacion postfija es: %s\n\n", notacionPosfija);

int resultado = evaluatePostfix(notacionPosfija);

printf("El resultado de la operacion es: %d\n", resultado);

free(notacionPosfija);

return 0;
}

else
{
printf("La cadena insertada no pertenece al lenguaje :(\n");
return 0;
}
```

Como pide el **punto 2**, creamos una función que convierte caracteres en enteros:

```
int charToInt(char c) {  
    return c - '0';  
}
```

Y la usamos para convertir a enteros todos los caracteres que fueran dígitos dentro de la función **evaluatePostfix( )**

```
for (i = 0; exp[i]; ++i) {  
    if (isdigit(exp[i]))  
        push(stack, charToInt(exp[i]));  
    else {  
        int val1 = pop(stack);  
        int val2 = pop(stack);  
        switch (exp[i]) {
```