



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

---

## ESTRUCTURA DE DATOS Y ALGORITMOS I

Actividad 2: Algoritmo operaciones calculadora

Laura Mildred Moreno Razo

**FECHA: 18/06/2021**

## Algoritmo

1. Escribir la cadena
2. Identificar cuantos elementos tiene la cadena y guardarlos en variable "num".
3. Restar uno a "num", el resultado dividirlo entre 2 y asignarlo a variable "nOp"
4. Crear un arreglo de enteros con tamaño "num-nOp" y un arreglo de caracteres con tamaño "nOp".
5. Recorrer cada uno de los elementos del arreglo, si están entre 48 y 57 (código ascii) asignar una posición dentro del arreglo de enteros menos '0' (código ascii) para así obtener el valor en enteros.
6. Si no están entre 48 y 57 guardar dentro del arreglo de caracteres (aumentar el contador de posición en 1 cada que se asigne un valor).
7. Al terminar de recorrer todos los elementos evaluar los valores dentro del arreglo de caracteres para operar.

Comenzando por buscar todos los \* y /

Si  $op(x) = 42$

La posición anterior a la operación será el valor de la multiplicación del número anterior al símbolo por el número siguiente, es decir:

$enteros[x] = enteros[x] * enteros[x+1];$

Desplazar los valores del arreglo de enteros hacia la izquierda para llenar entero (x+1)

Desplazar los valores del arreglo de caracteres hacia la izquierda para llenar operaciones (x)

Si  $op(x) = 47$

La posición anterior a la operación será el valor de la división del número anterior al símbolo por el número siguiente, es decir:

$enteros[x] = enteros[x] / enteros[x+1];$

Desplazar los valores del arreglo de enteros hacia la izquierda para llenar entero (x+1)

Desplazar los valores del arreglo de caracteres hacia la izquierda para llenar operaciones (x)

Después buscar los + y -

Si  $op(x) = 43$

La posición anterior a la operación será el valor de la suma del número anterior al símbolo con el número siguiente, es decir:

$enteros[x] = enteros[x] + enteros[x+1];$

Desplazar los valores del arreglo de enteros hacia la izquierda para llenar entero (x+1)

Desplazar los valores del arreglo de caracteres hacia la izquierda para llenar operaciones (x)

Si  $op(x) = 45$

La posición anterior a la operación será el valor de la resta del número anterior al símbolo con el número siguiente, es decir:

$enteros[x] = enteros[x] - enteros[x+1];$

Desplazar los valores del arreglo de enteros hacia la izquierda para llenar entero (x+1)

Desplazar los valores del arreglo de caracteres hacia la izquierda para llenar operaciones (x)

Si no es ninguno hay un error.

9. Imprimir el valor de  $enteros[0]$  donde ese encuentra el valor de la operación.

## Prueba del algoritmo

1. La cadena es: '1/3+7-2\*4'
2. num = 9
3. nOp = (unm-1)/2 = 4
4. num-nOp = 5, enteros[5] , operador[4]
5. enteros[0] = cadena[0] - '0'    49-48= 1  
operador[0] = cadena[1] = /  
enteros[1]= cadena[2] - '0'    51-48= 3  
operador[1]= cadena[3] = +  
enteros[2]= cadena[4] - '0'    55-48= 7  
operador[2]= cadena[5] = -  
enteros[3]= cadena[6] - '0'    50-48= 2  
operador[3]= cadena[7] = \*  
enteros[4]= cadena[8] - '0'    52-48= 4

Enteros

0	1	2	3	4
1	3	7	2	4

Operadores

0	1	2	3
/	+	-	*

8. 1/3=0.3333

Enteros

0	1	2	3	4
0.3333	7	2	4	0

Operadores

0	1	2	3
+	-	*	0

$$2*4=8$$

Enteros

0	1	2	3	4
0.3333	7	8	0	0

Operadores

0	1	2	3
+	-	0	0

$$0.33333+7=7.3333$$

Enteros

0	1	2	3	4
7.3333	8	0	0	0

Operadores

0	1	2	3
-	0	0	0

$$7.33333-8=-0.66666$$

Enteros

0	1	2	3	4
-0.66666	0	0	0	0

## Código en lenguaje C:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main (){
    int num,nOp,i,z=0;
    char* cadena;
    cadena = malloc(sizeof(char)*128);
    printf ("Escribe la operacion a realizar\n");
    fgets (cadena, 128, stdin);
    num = strlen(cadena); //Define el numero de caracteres
    nOp = (num-1)/2;    //Nos dice cuantas operaciones hay
    float enteros [num-nOp];
    char operaciones [nOp];

    for (i=0 ; i<num ; i++){
        if ('0'<=cadena[i] && cadena[i]<='9'){
            enteros [z] = cadena[i]- '0' ; //Transforma los valores a enteros y los guarda en un arreglo int

        } else {
            operaciones[z]= cadena[i]; //Guarda las operaciones en orden
            z= z+1;
        }
    }

    for (i=0 ; i<nOp ; i++){    //Ciclo para evaluar * y /
        switch (operaciones[i]){
            case 42:
                enteros [i+1]= enteros [i] * enteros [i+1];
                enteros [i]= enteros [i+1];
                break;
        }
    }
}
```

```

        case 47:
enteros [i+1]= enteros [i]/ enteros [i+1];
enteros [i]=enteros [i+1];
break;
default:printf ("\n");
    }
}

for (i=0 ; i<nOp ; i++){
    switch (operaciones[i]){ //Ciclo para evaluar + y -
        case 43:
enteros [i+1]= enteros [i] + enteros [i+1];
enteros [i]= enteros [i+1];
printf ("\n %f",enteros [i]);
break;

        case 45:
enteros [i+1]= enteros [i]- enteros [i+1];
enteros [i]=enteros [i+1];
printf ("\n %f",enteros [i]);
break;
        default:printf ("\n");
    }
}

printf ("\nEl ultimo numero es el resultado");

return 0;
}

```

```
C:\Users\mildr\Desktop>prueba.exe
Escribe la operacion a realizar
3+7
```

```
10.000000
```

```
El ultimo numero es el resultado
```

```
C:\Users\mildr\Desktop>prueba.exe
Escribe la operacion a realizar
3+7*4
```

```
31.000000
```

```
El ultimo numero es el resultado
```

```
C:\Users\mildr\Desktop>prueba.exe
Escribe la operacion a realizar
1/3+7-2*4
```

```
7.333333
-0.666667
```

```
El ultimo numero es el resultado
```

```
C:\Users\mildr\Desktop>_
```

El programa presenta excepciones con algunas operaciones debido a que no logré realizar el desplazamiento de los arreglos para cubrir los espacios ya operados.