



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Estructura de Datos y Algoritmos I**

**Actividad Asíncrona #05 Viernes**

Algoritmo para solución de Operaciones Matemáticas

Alumno:

**Flores Ramírez Eduardo Amilcar**

**No. Cuenta:**

31732297-4



**18 de junio del 2021**

## Algoritmo para resolver problemas matemáticos.

Para comenzar a trabajar en el Pseudocódigo, primero debemos de conocer las librerías que se van a emplear en este trabajo y con las cuales vamos a poder realizar operaciones básicas de matemáticas (suma, resta y división).

```
1  #include <stdio.h>
```

Siguiendo de esto, se declara el `-int main-` para posteriormente poder empezar a declarar nuestras variables, declaramos 3 variables y 1 arreglo. Las variables “x”, “y”, y “z” nos servirán mas adelante para poder almacenar el valor de los resultados de nuestras operaciones, mientras que por otro lado el arreglo es declarado de tamaño 10, en donde se puede apreciar una numeración del 1 al 10, y estos números funcionaran como valores funcionales con los cuales se realizaran las operaciones.

Cabe señalar que todas las variables tienen un tipo de dato, en donde el dato `-int-` se refiere a un valor entero y el dato `-float-` de refiere a un dato con muchas decimales.

```
3  int main()
4  {
5      int x;
6      int y;
7      float z;
8      float OP [10] = {1,2,3,4,5,6,7,8,9,10};
9  }
```

Después, imprimimos en la consola la palabra “OPERACIONES”, es en este apartado en donde se van a colocar las operaciones que se nos solicitan.

```
10  printf("\t OPERACIONES");
11
```

Ahora bien, comenzamos por determinar la primera operación, de forma que asignamos a “x” el resultado de dicha operación, de tal forma que con ayuda de los datos almacenados en el arreglo de una dimensión pudimos realizar la suma de  $3 + 7$  dando como resultado 10, una vez realizado esto con ayuda de un `-printf()-` imprimimos la operación y resultado almacenado en la variable “x”.

```
12  x = OP[2] + OP[6];
13  printf("\n 3 + 7 = %d",x);
14
```

La siguiente operación es muy parecida a la anterior, primero agregamos la variable “y”, en la cual se va a almacenar el dato resultante de dicha operación, en la cual se realizó la operación de  $3+7*4$  dando como resultado 31, y de igual manera imprimimos el resultado con una función -printf()-, el cual quedó almacenado en la variable “y”.

```
15     y = OP[2] + OP[6] * OP[3];
16     printf("\n 3 + 7 * 4 = %d",y);
17
```

Y finalmente, volvimos a realizar el mismo procedimiento solo que esta vez la operación se almacenará en la variable “z” y también se ocuparán números fraccionarios, de tal modo que debemos obtener un resultado con decimales, una vez entendido esto colocamos la operación haciendo el llamado correspondiente a nuestros valores, para finalmente realizar la operación y almacenarla en “z”.

```
18     z = (OP[0]/OP[2])+OP[6]-OP[1]*OP[3];
19     printf("\n 1/3 + 7 - 2 * 4 = %f",z);
20
```

Y para terminar se coloca la función -return 0;- para asegurarnos de que todo salga bien.

```
21     return 0;
22 }
23
```