

Problema

Realizar un algoritmo que determine el subconjunto de elementos que dentro de un arreglo, al ser sumados consecutivamente den como resultado 0.

Algoritmo propuesto

Data: Un arreglo **A**

Result: Un subconjunto del arreglo original cuyos elementos sumados dan como resultado 0

```
negativos = [];  
positivos = [];  
combinacionesPositivos = [];  
combinacionesNegativos = [];  
respuesta = [];  
separarPositivosNegativos(A, positivos, negativos);  
respuesta = compararIdenticos(positivos, negativos);  
if respuesta != [] then  
|   return respuesta  
end  
combinacionesPositivos = combinaciones(positivos);  
combinacionesNegativos = combinaciones(Negativos);  
respuesta = compararIdenticos(combinacionesPositivos, combinacionesNegativos);  
return respuesta
```

Algorithm 1: Laboratorio 5

¿Qué hace el método «separarPositivosNegativos»?

Esta función recibe como parámetro dos arreglos, uno donde colocará los números positivos, y otro donde pondrá los números negativos. De manera iterativa recorre el arreglo, destinando los elementos según si son mayores o menores que cero.

Data: Tres arreglos **A B C**

Result: Los elementos de un conjunto clasificados según su procedencia o anterioridad al cero.

```
for i = 0 to A.lenght do  
|   if A[i] > 0 then  
|   |   B.append(A[i])  
|   else  
|   |   C.append(A[i])  
|   end  
end
```

Algorithm 2: Laboratorio 5

¿Qué hace el método «CompararIdenticos»?

El método realiza una comparación entre elementos idénticos entre dos listas, si se cumple esta condición rompe el ciclo y retorna un subarreglo con los elementos reversos.

Data: Dos arreglos **A B**

Result: Un arreglo con elementos reversos

```
for itemA in A do
  for itemB in B do
    if itemA == -(itemB) then
      | return [itemA, itemB]
    end
  end
end
```

Algorithm 3: Laboratorio 5

¿Qué hace el método «Combinaciones»?

El método de combinaciones obtiene todas las posibles combinaciones entre elementos de un conjunto.