

## ELEMENTO MENOR

### I. ANÁLISIS

Descripción del problema: Dado un arreglo de números enteros, se buscará el menor elemento.

Datos de entrada: arreglo[]

Datos de salida: menorNumero

### II. PSEUDOCÓDIGO

INICIO

```
    FUNC    encontrarElementoMenor(arreglo[]):  ENTERO,    tam:
ENTERO) RET: ENTERO
```

```
    menorNumero:= arreglo[0]: ENTERO
```

```
    contador:=0:ENTERO
```

```
    MIENTRAS contador<tam
```

```
        SI(arreglo[contador]<menorNumero) ENTONCES
```

```
            menorNumero=arreglo[contador]
```

```
        FIN SI
```

```
        contador:=contador+1
```

```
    FIN MIENTRAS
```

```
    ESCRIBIR "El numero menor del arreglo es: " menorNumero
```

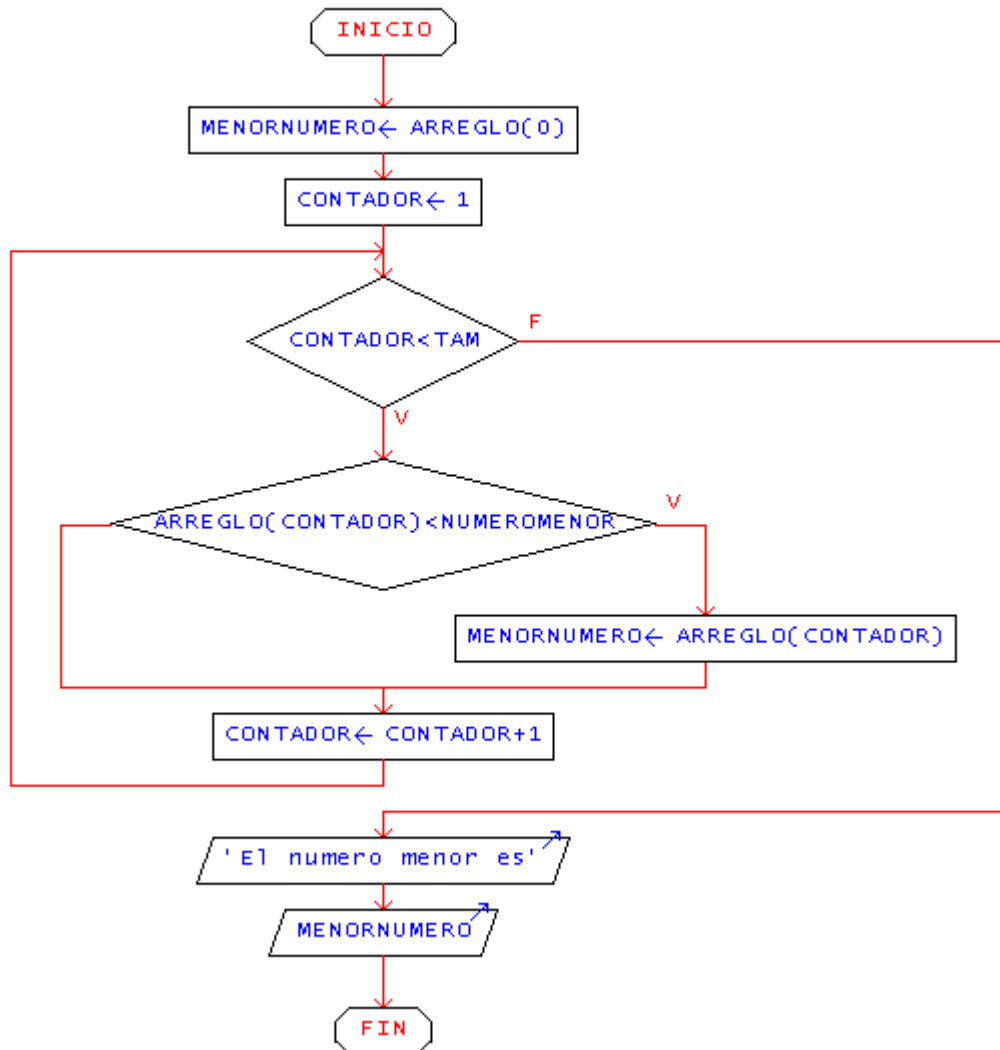
```
    RET 0
```

```
    FIN FUNC
```

FIN

### DIAGRAMA DE FLUJO

### PROCESO ENCONTRARELEMENTOMENOR



### III. PRUEBA DE ESCRITORIO

Variables	Valor de la variable							Salida
arreglo[]	{3,2,4,6,-3,8}							El numero menor es -3
menorNumero	3			2		-3		
contador	1	2	3	4	5	6	4	
tam	6							

### IV. CODIFICACIÓN

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

```

int encontrarElementoMenor(int arreglo[], int tam)
{
    int menorNumero, contador=0;
    menorNumero = arreglo[contador];
    while (contador<tam)
    {
        if (arreglo[contador]< menorNumero)
        {
            menorNumero= arreglo[contador];
        }
        contador++;
    }
    printf("El numero menor es: %d\n",menorNumero);
    return 0;
}

/*int main()
{
    int arreglo[] = {7,6,5,4,3,2};
    encontrarElementoMenor(arreglo, 6);

    int arreglo2[] = {-3,0,2,44,7,2};
    encontrarElementoMenor(arreglo2, 6);
    return 0;
}*/

```