ELEMENTO MENOR

OBJETIVO:

Identificar el elemento menor de un arreglo compuesto por elementos de tipo entero

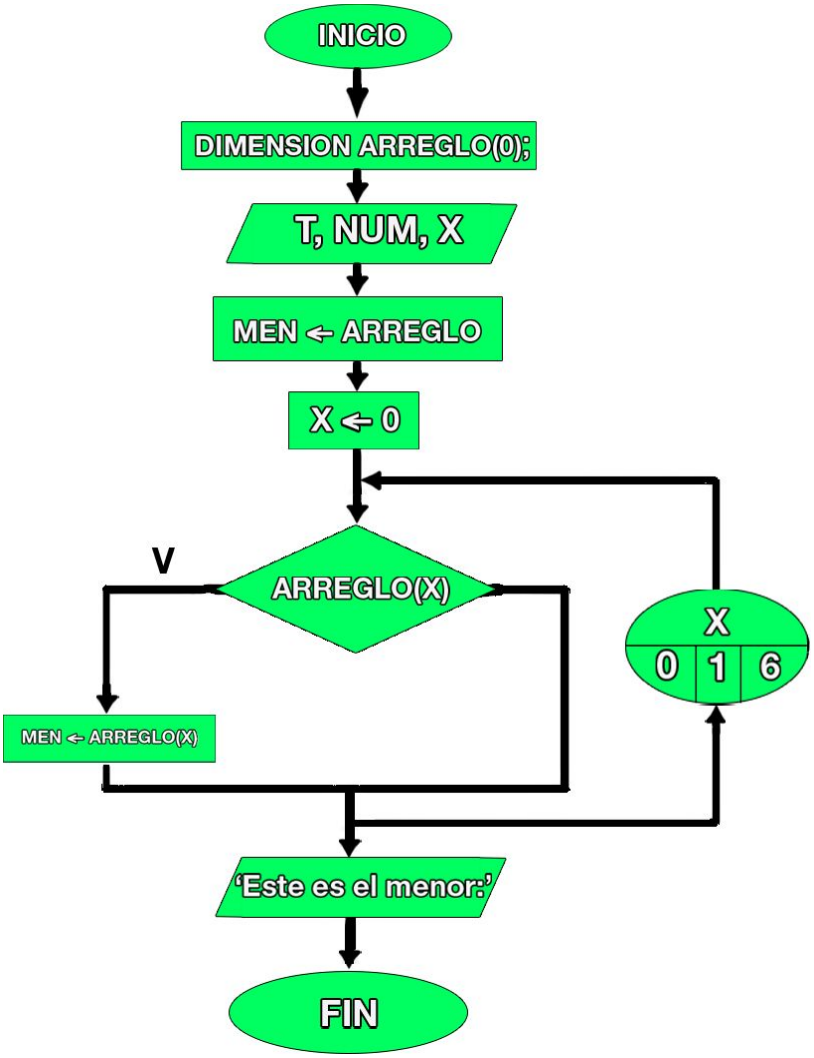
Datos de entrada: Un arreglo con n elementos de tipo entero

Datos de salida: Un número de tipo entero

ANÁLISIS

El problema a resolver es encontrar el elemento de menor valor dentro de un arreglo de enteros, por lo que se planteó un programa que analiza cada valor en función de ciclos mediante **for** e **if** que se repiten hasta que termine de registrar todos los elementos de nuestro arreglo o lista proporcionada, por lo que una vez termina de leer todo el arreglo (no importa si son 5 valores dentro del arreglo o 20), este ya habrá ubicado en base a determinadas condiciones para comparar cada uno y así clasificarlos, se procede a imprimir en pantalla el elemento de menor valor dentro de este arreglo.

DIAGRAMA DE FLUJO



PSEUDOCÓDIGO

INICIO

FUNC hallarMenorElemento(arreglo: ENTERO , t: ENTERO) RET: ENTERO
men:=arreglo[0]: ENTERO
PARA x:=0: ENTERO DESDE x<t HACER x:= x+1
SI(arreglo[x]<menor) ENTONCES
men=arreglo[x]
FIN SI
FIN PARA
ESCRIBIR “Este es el menor: “ men
RET 0
FIN FUNC
FIN

INICIO
FUNC principal (vacio) RET: vacio
arreglo:= (7,6,5,4,3,2): ENTERO
hallarMenorElemento(arreglo, 6)
FIN FUNC
FIN

PROGRAMA EN C

```c
#include <stdio.h>
int hallarMenorElemento(int arreglo [], int t)
{
int men, x;
men = arreglo[0];

for (x=0; x<t; x++)
{
if (arreglo[x]<men)
{
men=arreglo[x];
}
}
printf("Este es el menor %d\n", men);
return 0;
}
```

TEST

```c
//Programa test
#include "../menorElemento.c"
int main()
{

int arreglo2[] = {-3,0,2,44,6,2};
hallarMenorElemento(arreglo2,6);

int arreglo [] = {7,6,5,4,3,2}
hallarMenorElemento(arreglo, 6);

return 0;
}

```

PRUEBA DE ESCRITORIO DEL TEST

```
Output

/tmp/sGZm2RsaWb.o
Este es el menor: -3
Este es el menor: 2
```