

ELEMENTO MENOR

BRIGADA 6

Un **arreglo** permite almacenar múltiples valores en una sola variable en posiciones de memoria continuas, facilitando el posterior acceso a estos datos.

En el presente programa se necesita encontrar el valor mínimo entre un conjunto de datos enteros, por lo que se presentan dos opciones:

1. Los datos se obtienen del usuario quien los digita uno a uno.
2. Se crea una función que después se llama para analizar un arreglo ya compuesto por distintos valores.

ARREGLO QUE DIGITA EL USUARIO	ARREGLO DECLARADO
DATOS DE ENTRADA	
Son aquellos que el usuario va ingresando. Primero se solicita el número de datos a evaluar (extensión del arreglo) y después se solicita cada dato uno a uno.	Los datos se encuentran en el arreglo ya establecido.
El dato de salida es únicamente el elemento con el mínimo valor encontrado en el arreglo.	

Para la primera opción, se inicia declarando las variables necesarias:

arreglo [0], tam, menor, i = entero

Se solicita el número de valores que se va a comparar y dicho valor se guarda en la variable “*tam*”, después se imprime la solicitud para guardar cada uno de los valores.

Dentro del “FOR” se establece un valor inicial de **i=1**, donde siempre debe ser menor o igual al valor de la variable “**tam**” y se aumentará de uno en uno.

Se solicitan los números de acuerdo con el tamaño establecido y se guardan en el arreglo declarado en un inicio. La variable menor va cambiando de acuerdo con los valores introducidos, el primer valor se guarda en “**menor**”, si el próximo valor es menor a este, entonces este segundo valor pasa a tomar su lugar. Por lo que la variable “menor” siempre tendrá el valor mínimo del vector, y por ende es el que se imprime al final.

En la segunda opción se siguen los mismos pasos del primer código, sin embargo aquí no se le solicita al usuario ingresar un conjunto de datos, sino que ya se tiene un vector establecido y después se llama esta función llamada “**emenor**” para comparar los valores y ocupando un **for** donde el valor mínimo se guarda en la variable “**menor**” como en la primera opción se imprime el elemento con menor valor.

```

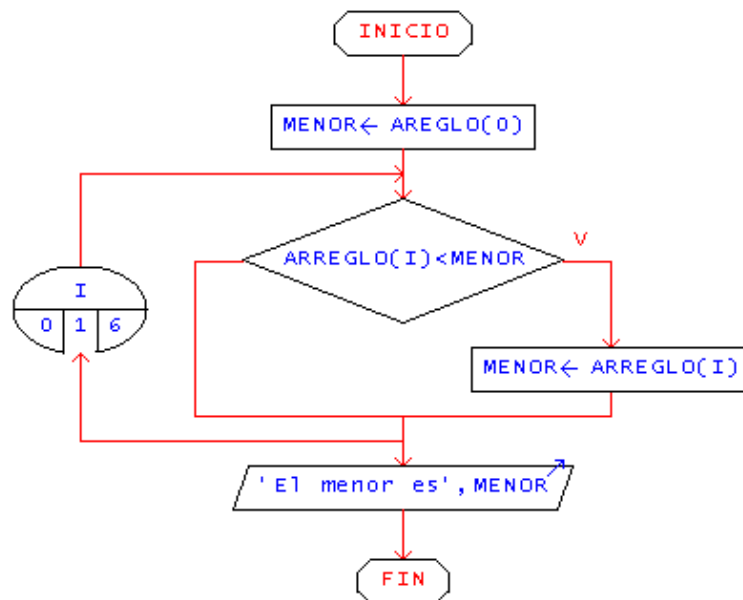
INICIO
    arreglo [0]:= entero
    tam, menor, i:= entero
    menor:=arreglo[0]

    ESCRIBIR "Numero de valores a comparar"
    LEER tam
    ESCRIBIR "Dime los numeros"

    PARA i DESDE 1 HASTA i<=tam HACER
        ESCRIBIR "Numero" i
        LEER arreglo[i]
        SI arreglo[i]<menor
            menor:=arreglo[i]
        FIN SI
    FIN PARA
    ESCRIBIR "El menor es" para
FIN

INICIO
    arreglo [0]:= entero
    tam, menor, i:= entero
    menor:=arreglo[0]

    PARA i DESDE 0 HASTA i<tam HACER
        SI arreglo[i]<menor
            menor:=arreglo[i]
        FIN SI
    FIN PARA
    ESCRIBIR "El menor es" para
FIN
    
```



CÓDIGO

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

int main()
{
    int arreglo[0];
    int tam, menor, i;
    menor= arreglo[0];

    printf("Numero de valores a comparar\n");
    scanf("%d", &tam);
    printf ("Dime los numeros\n");
    for (int i=1; i<=tam; i++)
    {
        printf ("Numero %d\n",i);
        scanf ("%d", &arreglo[i]);

        if (arreglo[i]<menor)
        {
            menor=arreglo[i];
        }
    }
    printf ("El menor es %d\n", menor);
    return 0;
}
```

ARREGLO PREDETERMINADO

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

int emenor(int arreglo[],int tam)
{
    int menor, i;
    menor=arreglo[0];

    for (int i=0; i<tam; i++)
    {
        if (arreglo[i]<menor)
        {
            menor=arreglo[i];
        }
    }
    printf ("El menor es %d\n", menor);
    return 0;
}
```

PRUEBA DE ESCRITORIO

Se analiza el siguiente arreglo: {17,7,23,166,74,34}

tam=6

i=1, y debe ser menor a 6, entonces:

Para **i**=1, hasta **i**<=tam

i=1 Entonces se imprime "*numero 1*", y se guarda en **arreglo[1]** hasta que se llega al 6, que es lo que establece la estructura "*para*"

El segundo valor es 7, entonces

arreglo[2]=7<**menor** (cuyo valor es 17) **SI**, entonces "menor" ahora vale 7

arreglo[3] es 23, entonces se compara de nuevo **23**<7 **NO**, entonces "menor" aun vale 7

166<7 **NO**

menor=7

74<7 **NO**

menor=7

34<7 **NO**

Entonces se imprime el valor de menor, es decir "7".

```
C:\Users\astro\OneDrive\Documentos\Programacion\intentos\elementomenor.exe
Numero de valores a comparar
6
Dime los numeros
Numero 1
17
Numero 2
7
Numero 3
23
Numero 4
166
Numero 5
74
Numero 6
34
El menor es 7

-----
Process exited after 18.98 seconds with return value 3221225477
Presione una tecla para continuar . . .
```

```
Emenor.cpp  elementomenor.cpp  TEST.cpp
1  #include "Emenor.cpp"
2
3  int main()
4  {
5      int arreglo[] = {17,7,23,166,74,34};
6      emenor(arreglo, 6);
7
8
9
C:\Users\astro\OneDrive\Documentos\Programacion\intentos\TEST.exe
El menor es 7
-----
Process exited after 0.697 seconds with return value 0
Presione una tecla para continuar . . .
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main()
6  {
7      int arreglo[0];
8      int tam, menor, i;
9      menor= arreglo[0];
10
11      printf("Numero de valores a comparar\n");
12      scanf("%d", &tam);
13      printf ("Dime los numeros\n");
14      for (int i=1; i<=tam; i++)
15      {
16          printf ("Numero %d\n",i);
17          scanf ("%d", &arreglo[i]);
18
19          if (arreglo[i]<menor)
20          {
21              menor=arreglo[i];
22          }
23      }
24      printf ("El menor es %d\n", menor);
25      return 0;
26  }
```

```
1
2 #include <stdio.h>
3 #include <stdio.h>
4
5 int emenor(int arreglo[],int tam)
6 {
7     int menor, i;
8     menor=arreglo[0];
9
10    for (int i=0; i<tam; i++)
11    {
12        if (arreglo[i]<menor)
13        {
14            menor=arreglo[i];
15        }
16    }
17    printf ("El menor es %d\n", menor);
18    return 0;
19 }
20
```