



*NOMBRE DE LA MATERIA:* **PROGRAMACIÓN**

*NRC:* **42555**

*HORARIO:* **MARTES Y JUEVES 9 AM – 10:55AM**

*ESTUDIANTE:* **EFRAIN ROBLES PULIDO**

*CODIGO:* **221350095**

*TEMA:* **ARREGLOS UNIDIMENSIONALES**

*FECHA:* **24 DE OCTUBRE DE 2021**

## Practica 34: Arreglos unidimensionales

### Pseudocódigo

//Efrain Robles Pulido

// Practica 34: Arreglos unidimensionales

Principal

Inicio

entero datos [10], i

desde (i ← 0; i<=9; inc i) //Llena el arreglo con elementos

inicio

imprimir ("Dame el elemento", i)

leer (datos[i])

fin

desde (i ← 9; i>=0; dec i) //Muestra los elementos del arreglo en sentido inverso

imprimir ("El elemento", i," es", datos[i])

regreso 0

Fin

### Código en lenguaje C

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>

int main() //Practica 34: Arreglos unidimensionales
{
    int datos[10], i;

    for(i=0; i<=9; i++) { //Llena el arreglo con elementos
        printf("\nDame el elemento %d\n", i);
        scanf("%d", &datos[i]);
    }

    for(i=9; i>=0; i--) //Muestra los elementos del arreglo en sentido inverso
        printf("%\tEl elemento %d es %d\n", i, datos[i]);

    return 0;
}
```

Dame el elemento 0  
90

Dame el elemento 1  
54

Dame el elemento 2  
21

Dame el elemento 3  
40

Dame el elemento 4  
72

Dame el elemento 5  
14

Dame el elemento 6  
2

Dame el elemento 7  
9

Dame el elemento 8  
54

Dame el elemento 9  
27

El elemento 9 es 27  
El elemento 8 es 54  
El elemento 7 es 9  
El elemento 6 es 2  
El elemento 5 es 14  
El elemento 4 es 72  
El elemento 3 es 40  
El elemento 2 es 21  
El elemento 1 es 54  
El elemento 0 es 90

Process returned 0 (0x0) execution time : 15.120 s  
Press any key to continue.

## Practica 35: Suma de dos arreglos unidimensionales

### Pseudocódigo

//Efrain Robles Pulido

// Practica 35: Suma de dos arreglos unidimensionales

Principal

Inicio

real A[5]  $\leftarrow$  {3.2,4.6,1.7,8.3,5.2}, B[5]  $\leftarrow$  {4.9,7.1,9.5,2.7,6.8}, C[5]

entero i

desde (i  $\leftarrow$  0; i $\leq$ 4; inc i) //Suma los arreglos que ya fueron inicializados

C[i] $\leftarrow$ A[i] + B[i]

desde (i  $\leftarrow$  0; i $\leq$ 4; inc i) //Muestra los elementos del arreglo en sentido inverso

imprimir (A[i], "+", B[i], " = ", C[i])

regreso 0

Fin

### Código en lenguaje

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>

int main() //Practica 35: Suma de dos arreglos inidimensionales
{
    float A[5]={3.2,4.6,1.7,8.3,5.2}, B[5]={4.9,7.1,9.5,2.7,6.8}, C[5];
    int i;

    for(i=0;i<=4;i++) //Suma los arreglos que ya fueron inicializados
        C[i]=A[i]+B[i];

    for(i=0;i<=4;i++) //Muestra completamente cada una de las sumas
        printf("\t%.2f + %.2f = %.2f\n",A[i],B[i],C[i]);
    return 0;
}
```

```
3.20 + 4.90 = 8.10
4.60 + 7.10 = 11.70
1.70 + 9.50 = 11.20
8.30 + 2.70 = 11.00
5.20 + 6.80 = 12.00

Process returned 0 (0x0)   execution time : 0.113 s
Press any key to continue.
```

## **Practica 36: Guardar 10 elementos de tipo real en un vector e imprimir en pantalla el "promedio" de todos sus elementos**

### **Pseudocódigo**

//Efrain Robles Pulido

//Practica 36: Guardar 10 elementos de tipo real en un vector e imprimir en pantalla el "promedio" de todos sus elementos

Principal

Inicio

real A[10], acum

entero i

desde ( $i \leftarrow 0$ ;  $i \leq 9$ ; inc i)

inicio

imprimir ("Dame calificación #", i+1, ":")

leer (A[i])

acum=acum + A[i]

fin

imprimir ("Promedio final:", acum/10)

regreso 0

Fin

## Código en lenguaje C

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>

int main()//Practica 36: Guardar 10 elementos de tipo real en
//un vector e imprimir en pantalla el "promedio" de todos sus elementos
{
    float A[10], acum;
    int i;

    for(i=0;i<=9;i++){
        printf("\nDame calificacion #%d:\n",i+1);
        scanf("%f",&A[i]);
        acum=acum+A[i];
    }

    printf("\n\tPromedio final: %.2f\n",acum/10);

    return 0;
}
```

```
Dame calificacion #1:
98

Dame calificacion #2:
56.3

Dame calificacion #3:
86.5

Dame calificacion #4:
76.8

Dame calificacion #5:
94.87

Dame calificacion #6:
56.48

Dame calificacion #7:
85.67

Dame calificacion #8:
65.98

Dame calificacion #9:
45.98

Dame calificacion #10:
95.47

Promedio final: 76.20

Process returned 0 (0x0)   execution time : 32.513 s
Press any key to continue.
```

## Practica 37: Encontrar el elemento "mayor" de un arreglo de 15 elementos

### Pseudocódigo

//Efrain Robles Pulido	inicio
//Practica 37: Encontrar el elemento "mayor" de un arreglo de 15 elementos	//se ira actualizando el valor mayor
Principal	//confome vaya avanzando
Inicio	si (A[i]>mayor)
entero A [15], i, mayor ← 0	inicio
desde (i ← 0; i<=14; inc i)	mayor ← A[i]
inicio	fin
imprimir ("Dame un numero:")	fin
leer (A[i])	imprimir ("El mayor es:", mayor)
fin	regreso 0
desde (i ← 0; i<=14; inc i)	Fin

### Código en lenguaje C

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>

int main()//Practica 37: Encontrar el elemento "mayor" de un arreglo de 15 elementos
{
    int A[15], i, mayor=0;

    for(i=0;i<=14;i++){
        printf("\nDame un numero:\n");
        scanf("%d",&A[i]);
    }
    for(i=0;i<=14;i++){
        if(A[i]>mayor){
            mayor=A[i]; //se ira actualizando el valor mayor
        } //confome vaya avanzando
    }
    printf("\n\tEl mayor es %d\n",mayor);
    return 0;
}
```

Dame un numero:

20

Dame un numero:

41

Dame un numero:

31

Dame un numero:

10

Dame un numero:

14

Dame un numero:

18

Dame un numero:

46

Dame un numero:

78

Dame un numero:

21

Dame un numero:

95

Dame un numero:

64

Dame un numero:

25

Dame un numero:

94

Dame un numero:

58

Dame un numero:

11

El mayor es 95

Process returned 0 (0x0) execution time : 31.216 s

Press any key to continue.



**Practica 38: Llenar dos arreglos de tamaño 10 con elementos de tipo real y dejar en un tercer arreglo la multiplicación de los dos anteriores. Mostrar en pantalla la operación completa de cada una de las multiplicaciones**

**Pseudocódigo**

//Efrain Robles Pulido

// Practica 38: Llenar dos arreglos de tamaño 10 con elementos de tipo real y dejar en un tercer arreglo la multiplicación de los dos anteriores. Mostrar en pantalla la operación completa de cada una de las multiplicaciones

Principal

Inicio

real A [10], B[10], C[10]

entero i

desde (i  $\leftarrow$  0; i<=9; inc i)

inicio

imprimir ("Dame un numero:")

leer (A[i])

fin

desde (i  $\leftarrow$  0; i<=9; inc i)

inicio

imprimir ("Dame un numero:")

leer (B[i])

fin

desde (i  $\leftarrow$  0; i<=9; inc i)

inicio

C[i]  $\leftarrow$  A[i] \* B[i]

fin

desde (i  $\leftarrow$  0; i<=9; inc i)

inicio

imprimir (A[i], " x ", B[i], " = ", C[i])

fin

regreso 0

Fin

## Código en lenguaje C

```
//Efrain Robles Pulido
#include <stdio.h>
#include <stdlib.h>

int main(){
/*Practica 38: Llenar dos arreglos de tamaño 10 con elementos de tipo real
y dejar en un tercer arreglo la multiplicacion de los dos anteriores. Mostrar
en pantalla la operacion completa de cada una de las multiplicaciones*/
    float A[10], B[10], C[10];
    int i;

    for(i=0;i<=9;i++){
        printf("\nDame un numero para A:\n");
        scanf("%d",&A[i]);
    }
    for(i=0;i<=9;i++){
        printf("\nDame un numero para B:\n");
        scanf("%d",&B[i]);
    }
    for(i=0;i<=9;i++){
        C[i]=A[i]*B[i];
    }
    for (i=0;i<=9;i++)
        printf("\n\t%d x %d = %d\n",A[i],B[i],C[i]);
    return 0;
}
```

Dame un numero para A:  
3.5

Dame un numero para A:  
7.34

Dame un numero para A:  
3.5

Dame un numero para A:  
2.7

Dame un numero para A:  
0.9

Dame un numero para A:  
2.4

Dame un numero para A:  
8.4

Dame un numero para A:  
2.7

Dame un numero para A:  
4.5

Dame un numero para A:  
1.2

Dame un numero para B:  
8.2

Dame un numero para B:  
6.4

Dame un numero para B:  
1.2

Dame un numero para B:  
9.5

Dame un numero para B:  
3.1

Dame un numero para B:  
7.2

Dame un numero para B:  
8.4

Dame un numero para B:  
3.1

Dame un numero para B:  
4.8

Dame un numero para B:  
3.2

3.50 x 8.20 = 28.70

7.34 x 6.40 = 46.98

3.50 x 1.20 = 4.20

2.70 x 9.50 = 25.65

0.90 x 3.10 = 2.79

2.40 x 7.20 = 17.28

8.40 x 8.40 = 70.56

2.70 x 3.10 = 8.37

4.50 x 4.80 = 21.60

1.20 x 3.20 = 3.84

Process returned 0 (0x0) execution time : 54.667 s  
Press any key to continue.