



---

**Universidad de las Fuerzas Armadas ESPE**

**Departamento:**

**Carrera: Ingeniería en Electrónica y Automatización**

**Taller académico N°: 1**

**Parcial Nro:2**

---

**1. Información General**

- **Asignatura:** Fundamentos de Programación
  - **Apellidos y nombres de los estudiantes:** Alvear Alexander, Campoverde Anthony, Velecela Mateo
  - **NRC:** 20823
  - **Fecha de realización:** 03/05/2025
- 

**2. Objetivo del Taller y Desarrollo**

**Objetivo del Taller:**

Desarrollar habilidades prácticas en el manejo de arreglos (vectores) en C y pseudocódigo en PseInt, aplicando estructuras de control (como el ciclo for) para operaciones básicas de recorrido, ordenamiento y manipulación de datos, con énfasis en la eficiencia del código.

**Desarrollo:**

Aprendimos a manejar arreglos en C con ciclos for, resolviendo ejercicios prácticos para la aplicación intuitiva de conocimientos. Al final, discutimos los resultados y reforzamos la lógica de índices y bucles.

---

**U2 T1 VECTORES Y FUNCIONES**

**Problema 2.1.3. Vector con término general dado**

Sea la sucesión:

$$v_k = k^2 + 3,$$



Desarrolle un programa que lea el número  $n$  de componentes que se quieren calcular de la sucesión y almacenarlas en un vector  $vec$ , tal que  $vec(i) = v_i$ . Se mostrará el vector por pantalla. Puede asumir que  $n$  será siempre menor o igual a 100.

Para calcular las componentes del vector se utilizará una iteración con un índice tomando valores de 1 a  $n$  en diagrama de flujo (de 0 a  $n-1$  en C). A la vez, se irá calculando la componente ( $vec(i) = i^2 + 3$ ) y mostrándola por pantalla.

- **Tabla de Objetos**

OBJETO	NOMBRE	VALOR	TIPO
M1	n	Variable	Entero
M2	vec	Vector Variable	Entero

- **Seudocódigo**

```
1  Algoritmo VectorConTerminoGeneral
2      Definir n, i Como Entero
3      Dimension vec[100]
4
5
6      Escribir "Ingrese el número de términos n (<= 100): "
7      Leer n
8
9
10     Si n ≤ 0 O n > 100 Entonces
11         Escribir "Error: n debe ser un entero positivo <= 100."
12     Sino
13
14         Para i ← 1 Hasta n Hacer
15             vec[i] ← (i * i) + 3
16         FinPara
17
18
19         Escribir "El vector es:"
20         Para i ← 1 Hasta n Hacer
21             Escribir Sin Saltar vec[i], " "
22         FinPara
23         Escribir ""
24     FinSi
25 FinAlgoritmo
```

Algoritmo VectorConTerminoGeneral

Definir n, i Como Entero

Dimension vec[100]

Escribir "Ingrese el número de términos n (<= 100): "

Leer n

Si n <= 0 O n > 100 Entonces

Escribir "Error: n debe ser un entero positivo <= 100."

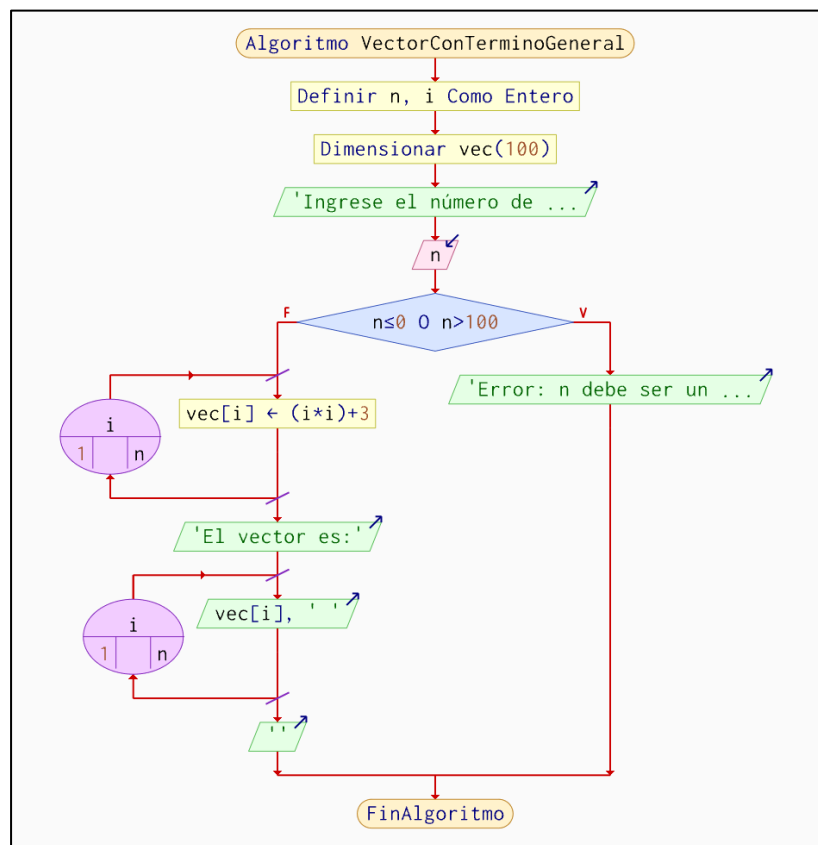


Sino

```
Para i <- 1 Hasta n Hacer  
    vec[i] <- (i * i) + 3  
FinPara
```

```
Escribir "El vector es:"  
Para i <- 1 Hasta n Hacer  
    Escribir Sin Saltar vec[i], " "  
FinPara  
Escribir ""  
FinSi  
FinAlgoritmo
```

- Diagrama de Flujo





- Código en C

```
Start here X Codigo-1-Segundo-Parcial-Deber-1-FP-Grupo-1.c X
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main () {
5      int n;
6      int vec[100];
7
8
9      printf("Ingrese el numero de terminos n (>=100): ");
10     scanf("%d", &n);
11
12     if(n<=0 || n>100){
13         printf("Error, el numero ingresado no cumple con la condicion indicada (>=100). \n");
14         return 1;
15     }
16     for (int i=1; i<=n ; i++){
17         vec[i-1]=i*i+3;
18     }
19
20     printf("El vector es:\n");
21     for (int j=0; j<n; j++){
22         printf("%d ", vec[j]);
23     }
24     printf("\n");
25
26     return 0;
27 }
28
29
```

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

int main (){
    int n;
    int vec[100];

    printf("Ingrese un numero que sea <=100: ");
    scanf("%d", &n);

    if(n<=0 || n>100){
        print("Error, no cumple la condicion (>=100) \n");
        return 1;
    }
    for (int i=1; i<n; i++){
        vec[i-1]=i*i + 3;
    }

    printf("El vector es: \n");

    for (int j=0; j<n; j++){

        printf("%d", vec[j]);
    }
}
```



```
printf("\n");  
  
    return 0;  
}
```

- **Prueba de Escritorio**

```
C:\Users\Anthony\Desktop\codigos\Codigo-1-Segundo-Parcial-Deber-1-FP-Grupo-1.exe  
Ingrese el numero de terminos n (>=100): 5  
El vector es:  
4 7 12 19 28  
  
Process returned 0 (0x0)   execution time : 4.633 s  
Press any key to continue.
```

**Problema 2.1.4. Comprobar si dos valores pertenecen a un vector**

Realice un algoritmo que lea dos números enteros por teclado y determine si ambos valores forman parte de un vector de enteros previamente definido de dimensión 15.

La solución se basa en dos variables bandera, que representan si uno de los números está en el vector. Se inicializan ambas a 0, y se recorre el vector comparando cada componente con los valores leídos por el teclado. Si alguno coincide, se cambia el valor de la bandera asociada a 1. Al finalizar, si ambas valen 1, el resultado será positivo.

- **Tabla de Objetos**

OBJETO	NOMBRE	VALOR	TIPO
M1	vec	Vector Constante	Entero
M2	size	Constante	Entero
M3	num1	Variable	Entero
M4	num2	Variable	Entero
M5	flag1	Variable	Entero
M6	flag2	Variable	Entero



- Seudocódigo

```
Pseudocodigo-2-Segundo-Parcial-Deber-1-FP-Grupo-1.psc* X
1  Algoritmo ComprobarValoresEnVector
2
3      Definir vec Como Entero
4      Dimension vec[15]
5      vec[1] ← 4
6      vec[2] ← 7
7      vec[3] ← 12
8      vec[4] ← 19
9      vec[5] ← 28
10     vec[6] ← 46
11     vec[7] ← 9
12     vec[8] ← 6
13     vec[9] ← 78
14     vec[10] ← 89
15     vec[11] ← 21
16     vec[12] ← 95
17     vec[13] ← 1
18     vec[14] ← 73
19     vec[15] ← 10
20
21
22     Definir num1, num2, flag1, flag2, i Como Entero
23     flag1 ← 0
24     flag2 ← 0
25
26     Escribir "Ingrese el primer número: "
27     Leer num1
28     Escribir "Ingrese el segundo número: "
29     Leer num2
30
31     Para i ← 1 Hasta 15 Hacer
32         Si vec[i] = num1 Entonces
33             flag1 ← 1
34         FinSi
35         Si vec[i] = num2 Entonces
36             flag2 ← 1
37         FinSi
38     FinPara
39
40
41     Si flag1 = 1 Y flag2 = 1 Entonces
42         Escribir "Ambos números están en el vector."
43     Sino
44         Escribir "Al menos uno de los números no está en el vector."
45     FinSi
46 FinAlgoritmo
```



### Algoritmo ComprobarValoresEnVector

Definir vec Como Entero

Dimension vec[15]

vec[1] <- 4

vec[2] <- 7

vec[3] <- 12

vec[4] <- 19

vec[5] <- 28

vec[6] <- 46

vec[7] <- 9

vec[8] <- 6

vec[9] <- 78

vec[10] <- 89

vec[11] <- 21

vec[12] <- 95

vec[13] <- 1

vec[14] <- 73

vec[15] <- 10

Definir num1, num2, flag1, flag2, i Como Entero

flag1 <- 0

flag2 <- 0

Escribir "Ingrese el primer número: "

Leer num1

Escribir "Ingrese el segundo número: "

Leer num2

Para i <- 1 Hasta 15 Hacer

Si vec[i] = num1 Entonces

flag1 <- 1

FinSi

Si vec[i] = num2 Entonces

flag2 <- 1

FinSi

FinPara

Si flag1 = 1 Y flag2 = 1 Entonces

Escribir "Ambos números están en el vector."

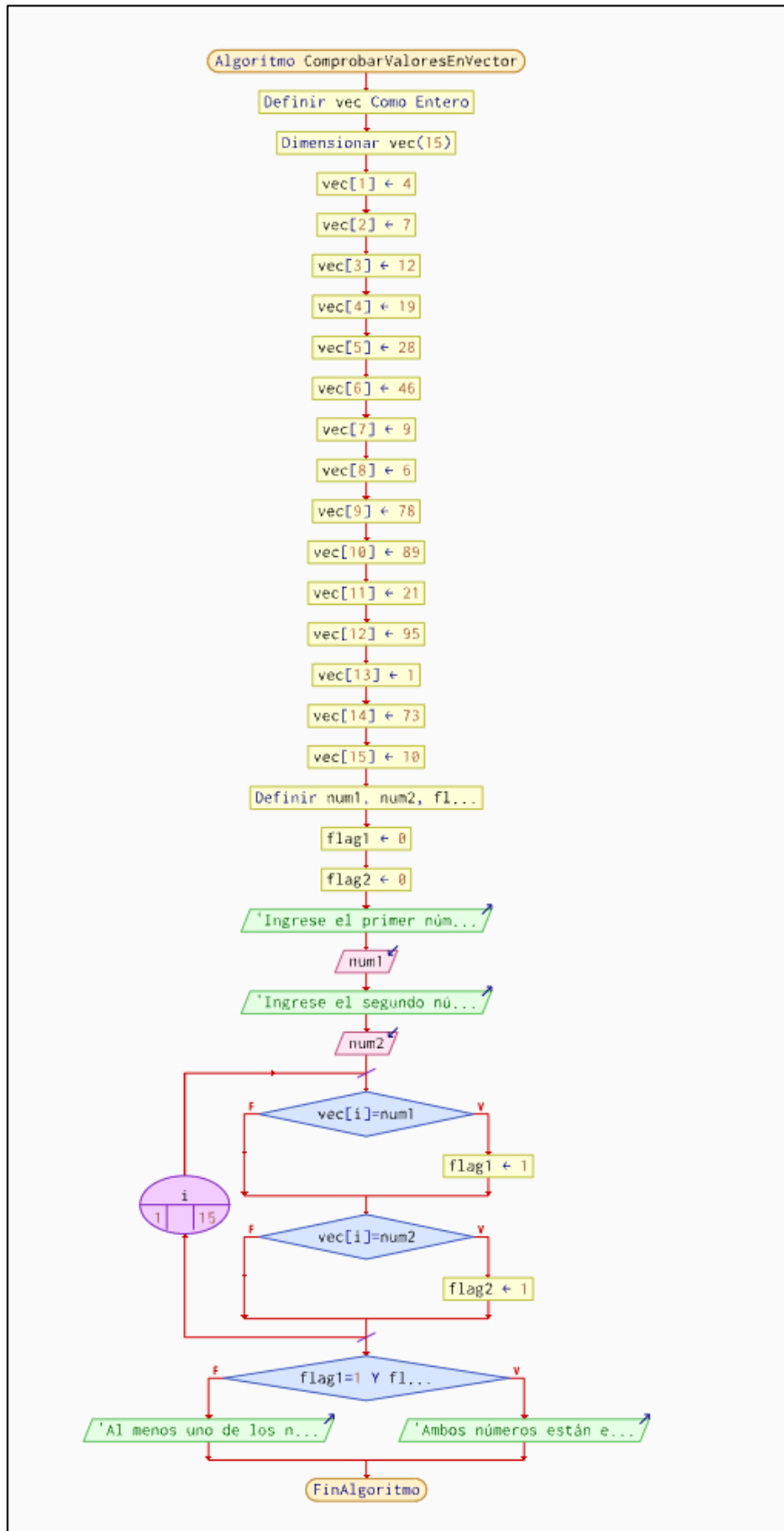
Sino

Escribir "Al menos uno de los números no está en el vector."

FinSi

FinAlgoritmo

- Diagrama de Flujo







- Código en C

```
Start here X Codigo-2-Segundo-Parcial-Deber-1-FP-Grupo-1.c X
1 //Universidad de las fuerzas Armadas "ESPE"
2 //GRUPO 1: ALEXANDER ALVEAR, ANTHONY CAMPOVERDE, MATEO VELECELA
3 //CODIGO 1; U2 T1
4 //VECTOR CON TERMINO GENERAL
5
6
7 #include <stdio.h>
8 #include <stdlib.h>
9
10 int main(){
11     int vec[] = {4,7,12,19,28,46,9,6,78,89,21,95,1,73,10};
12     int size = 15;
13
14     int num1,num2,flag1=0,flag2=0;
15
16     printf("Ingrese el primer numero: ");
17     scanf("%d", &num1);
18
19     printf("Ingrese el segundo numero: ");
20     scanf("%d", &num2);
21
22     for(int i=0; i<size; i++){
23         if (vec[i]== num1){
24             flag1=1;
25         }
26         if (vec[i]==num2){
27             flag2=1;
28         }
29     }
30     if(flag1 && flag2){
31         printf("Ambos numeros estan en el vector. \n");
32     } else{
33         printf("Al menos uno de los numeros no esta en el vector. \n");
34     }
35     return 0;
36 }
37
38
```

```
//Universidad de las fuerzas Armadas "ESPE"
//GRUPO 1: ALEXANDER ALVEAR, ANTHONY CAMPOVERDE, MATEO VELECELA
//CODIGO 1; U2 T1
//VECTOR CON TERMINO GENERAL

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

int main(){
    int vec[] = {4,7,12,19,28,46,9,6,78,89,21,95,1,73,10};
    int size = 15;

    int num1,num2,flag1=0,flag2=0;

    printf("Ingrese el primer numero: ");
    scanf("%d", &num1);

    printf("Ingrese el segundo numero: ");
```



```
scanf("%d", &num2);

for(int i=0; i<size; i++){
    if (vec[i]== num1){
        flag1=1;
    }
    if (vec[i]==num2){
        flag2=1;
    }
}
if(flag1 && flag2){
    printf("Ambos numeros estan en el vector. \n");
} else{
    printf("Al menos uno de los numeros no esta en el vector. \n");
}
return 0;
}
```

- **Prueba de Escritorio**

```
C:\Users\Anthony\Desktop\codigos\Codigo-2-Segundo-Parcial-Deber-1-FP-Grupo-1.exe
Ingrese el primer numero: 12
Ingrese el segundo numero: 28
Ambos numeros estan en el vector.

Process returned 0 (0x0)   execution time : 10.649 s
Press any key to continue.
_
```

### **Problema 2.1.5 Vector de factoriales.**

Dado un vector, Vec, que contiene los primeros 15 números naturales, calcule un vector fact con sus factoriales y mostrarlo por pantalla.

Nota: ¿Por qué a partir del número 12 no funciona correctamente el cálculo del factorial? ¿Qué se podría hacer para evitarlo?

El algoritmo consiste en dos bucles anidados, uno externo, que da valores a la variable i entre 1 y 15, y otro interno, que calcula la factorial de la componente i-ésima. El algoritmo deja de funcionar para el valor 12 porque en la codificación en C se han empleado variables de tipo entero. El valor de 13 excede la capacidad de



almacenamiento de una variable entera, y el cálculo se corrompe. Una posible solución es emplear variables de tipo unsigned long int (entero largo sin signo), que incrementan la capacidad de almacenamiento.

- **Tabla de Objetos**

OBJETO	NOMBRE	VALOR	TIPO
M1	vec	Vector Constante	Entero
M2	fact	Vector Variable	Entero de 64 bits sin signo

- **Seudocódigo**

```
Pseudocodigo-3-Segundo-Parcial-Deber-1-FP-Grupo-1.psc X
1  Algoritmo VectorFactorialesGrandes
2    // definicion de variables, unas como entero otra como real (por lo grande que es)
3    Definir Vec, j Como Entero
4    Definir fact Como Real           // real para numero grandes
5    Dimension Vec[15], fact[15]      // indice del vector de 1 al 15
6
7    // llenar los numeros del 1 al 15 al vector
8    Para i <- 1 Hasta 15 Con Paso 1 Hacer
9      Vec[i] <- i
10   FinPara
11
12   // calculo de factoriales
13   Para i <- 1 Hasta 15 Con Paso 1 Hacer
14     fact[i] <- 1 // iniciamos factorial en 1
15
16     Para j <- 1 Hasta Vec[i] Con Paso 1 Hacer
17       fact[i] <- fact[i] * j // calculo acumulativo de factores
18     FinPara
19   FinPara
20
21   // resultados
22   Escribir "Vector de factoriales:"
23   Para i <- 1 Hasta 15 Con Paso 1 Hacer
24     Escribir Vec[i], "! = ", fact[i]
25   FinPara
26 FinAlgoritmo
```

**Algoritmo VectorFactorialesGrandes**

// definicion de variables, unas como entero otra como real (por lo grande que es)

Definir Vec, j Como Entero

Definir fact Como Real // real para numero grandes

Dimension Vec[15], fact[15] // indice del vector de 1 al 15

// llenar los numeros del 1 al 15 al vector

Para i <- 1 Hasta 15 Con Paso 1 Hacer

Vec[i] <- i

FinPara

// calculo de factoriales

Para i <- 1 Hasta 15 Con Paso 1 Hacer

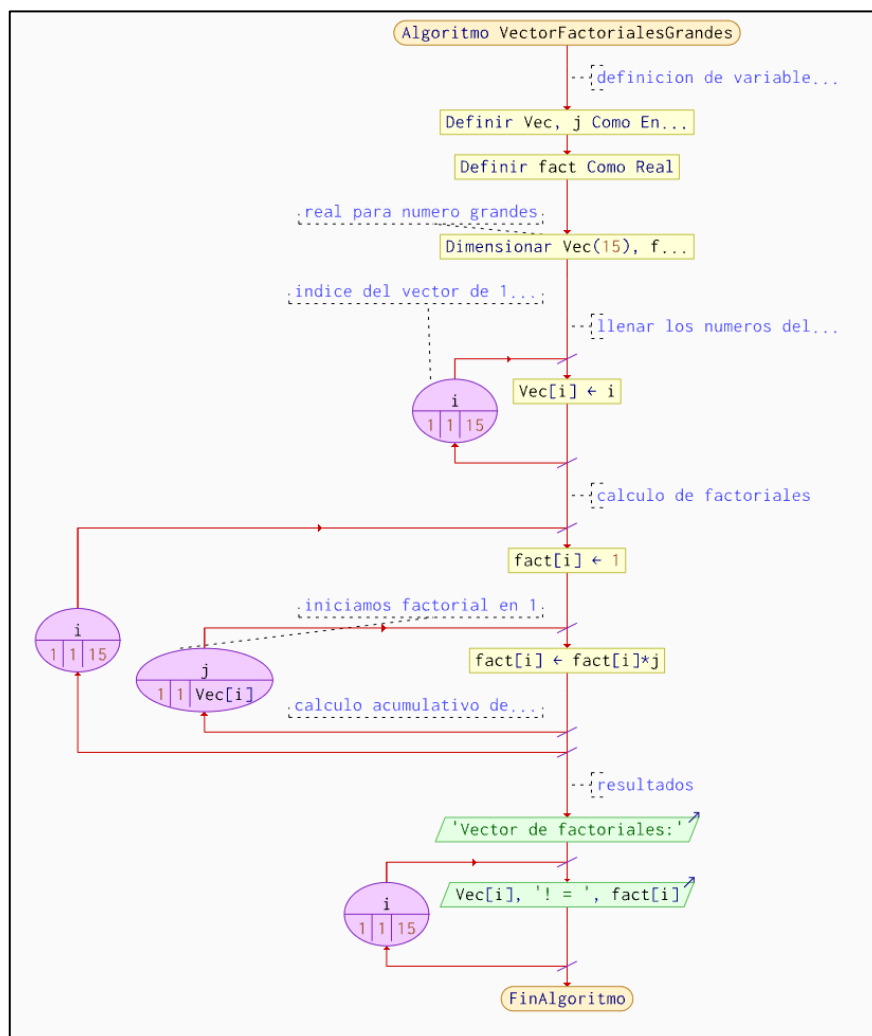
fact[i] <- 1 // iniciamos factorial en 1

```

Para j <- 1 Hasta Vec[i] Con Paso 1 Hacer
    fact[i] <- fact[i] * j // calculo acumulativo de factores
FinPara
FinPara

// resultados
Escribir "Vector de factoriales:"
Para i <- 1 Hasta 15 Con Paso 1 Hacer
    Escribir Vec[i], "!" = ", fact[i]"
FinPara
FinAlgoritmo
  
```

- Diagrama de Flujo





- Código en C

```
Start here X *Codigo-3-Segundo-Parcial-Deber-1-FP-Grupo-1.c X
1 //Universidad de las fuerzas Armadas "ESPE"
2 //GRUPO 1: ALEXANDER ALVEAR, ANTHONY CAMPOVERDE, MATEO VELECELA
3 //CODIGO 3; U2 T1
4 //FACTORIAL DE NUMEROS NATURALES
5
6 #include <stdio.h>
7 #include <stdlib.h>
8
9 int main () {
10
11     int vec[15]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
12     unsigned long long int fact[15];
13
14     for(int i=0; i<15; i++) {
15         fact[i]=1;
16         for (int j=1; j<=vec[i]; j++){
17             fact[i]= fact[i]*j ;
18         }
19     }
20
21     printf("Vector en factoriales: \n");
22     for(int i=0; i<15; i++){
23         printf("%d! = %llu\n", vec[i], fact[i]);
24     }
25     return 0;
26 }
27
```

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

int main (){

    int vec[15]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
    unsigned long int fact[15];

    for(int i=0; i<15; i++){
        fact[i]=1;
        for (int j=1; j<=vec[i]; j++){
            fact[i]= fact[i]*j ;
        }
    }

    printf("Vector en factoriales: \n");
    for(int i=0; i<15; i++){
        printf("%d! = %llu\n", vec[i], fact[i]);
    }
    return 0;
}
```



- Prueba de Escritorio

```
C:\Users\Anthony\Desktop\codigos\Codigo-3-Segundo-Parcial-Deber-1-FP-Grupo-1.exe
Vector en factoriales:
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
13! = 6227020800
14! = 87178291200
15! = 1307674368000

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

### Problema 2.1.6 Ordenación de un vector.

Desarrolle un programa que ordene un vector de 10 componentes de mayor a menor valor. Asuma que el vector está ya leído y almacenado en memoria.

Existen varios métodos de ordenación. Se expondrán dos:

- Ordenación iterativa. Esta solución emplea dos bucles anidados para comparar cada elemento del vector con los que le siguen, intercambiando las parejas de elementos fuera de orden. Por ejemplo, para ordenar un vector de  $n$  elementos numerados de 0 a  $n-1$  (en DF de 1 a  $n$ ), se comienza comparando el elemento 0 con los que le siguen, es decir, las parejas de elementos (0-1), (0-2), (0-3), ..., (0- $n-1$ ) (ojo, en DF se numeran los elementos desde 1). Si alguna de estas parejas presenta un orden inverso al deseado, se intercambian las posiciones de sus elementos en el vector. Tras comparar la última pareja, se tiene la garantía de que el elemento de mayor valor está en la posición 0. El procedimiento se repite a continuación con el elemento 1, esto es, se estudian las parejas (1-2), (1-3), ..., (1- $n-1$ ), intercambiando aquellas fuera de orden. El procedimiento se repite hasta llegar al elemento  $n-2$ , que debe ser comparado con el último, es decir, ( $n-2-n-1$ ). La solución aparece detallada en el diagrama de la figura 2.6, junto con su tabla de objetos, así como en la correspondiente codificación en C en el código 2.1.6.



- Tabla de Objetos

OBJETO	NOMBRE	VALOR	TIPO
M1	vector	Vector Constante	Entero
M2	i	Variable	Entero
M3	j	Variable	Entero
M4	temp	Variable	Entero
M5	n	Constante	Entero

- Seudocódigo

```
Pseudocodigo-4-Segundo-Parcial-Deber-1-FP-Grupo-1.psc X
1  //Universidad de las fuerzas Armadas "ESPE"
2  //GRUPO 1: ALEXANDER ALVEAR, ANTHONY CAMPOVERDE, MATEO VELECELA
3  //CODIGO 4; U2 T1
4  //Orden Iterativo para vector de mayor a menor
5
6  Algoritmo OrdenacionVectorDescendente
7      // definición de variables
8      Definir vector Como Entero
9      Definir i, j, temp, n Como Entero
10     Dimension vector[10]
11     n ← 10
12
13     // asignacion de valores al vector (puede modificarse)
14     vector[1] ← 23
15     vector[2] ← 5
16     vector[3] ← 78
17     vector[4] ← 12
18     vector[5] ← 9
19     vector[6] ← 34
20     vector[7] ← 67
21     vector[8] ← 42
22     vector[9] ← 3
23     vector[10] ← 19
24
25     // mostrar vector original
26     Escribir "Vector original:"
27     Para i ← 1 Hasta n Con Paso 1 Hacer
28         Escribir Sin Saltar vector[i], " "
29     FinPara
30     Escribir ""
31
32     // algoritmo para orden descendente por ordenación
33     Para i ← 1 Hasta n-1 Con Paso 1 Hacer
34         Para j ← i+1 Hasta n Con Paso 1 Hacer
35             Si vector[i] < vector[j] Entonces
36                 // Intercambiar elementos
37                 temp ← vector[i]
38                 vector[i] ← vector[j]
39                 vector[j] ← temp
40             FinSi
41         FinPara
42     FinPara
43
44     // mostrar resultado
45     Escribir "Vector ordenado (mayor a menor):"
46     Para i ← 1 Hasta n Con Paso 1 Hacer
47         Escribir Sin Saltar vector[i], " "
48     FinPara
49     Escribir ""
50 FinAlgoritmo
```



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA



#### Algoritmo OrdenacionVectorDescendente

```
// definición de variables
Definir vector Como Entero
Definir i, j, temp, n Como Entero
Dimension vector[10]
n <- 10

// asignacion de valores al vector (puede modificarse)
vector[1] <- 23
vector[2] <- 5
vector[3] <- 78
vector[4] <- 12
vector[5] <- 9
vector[6] <- 34
vector[7] <- 67
vector[8] <- 42
vector[9] <- 3
vector[10] <- 19

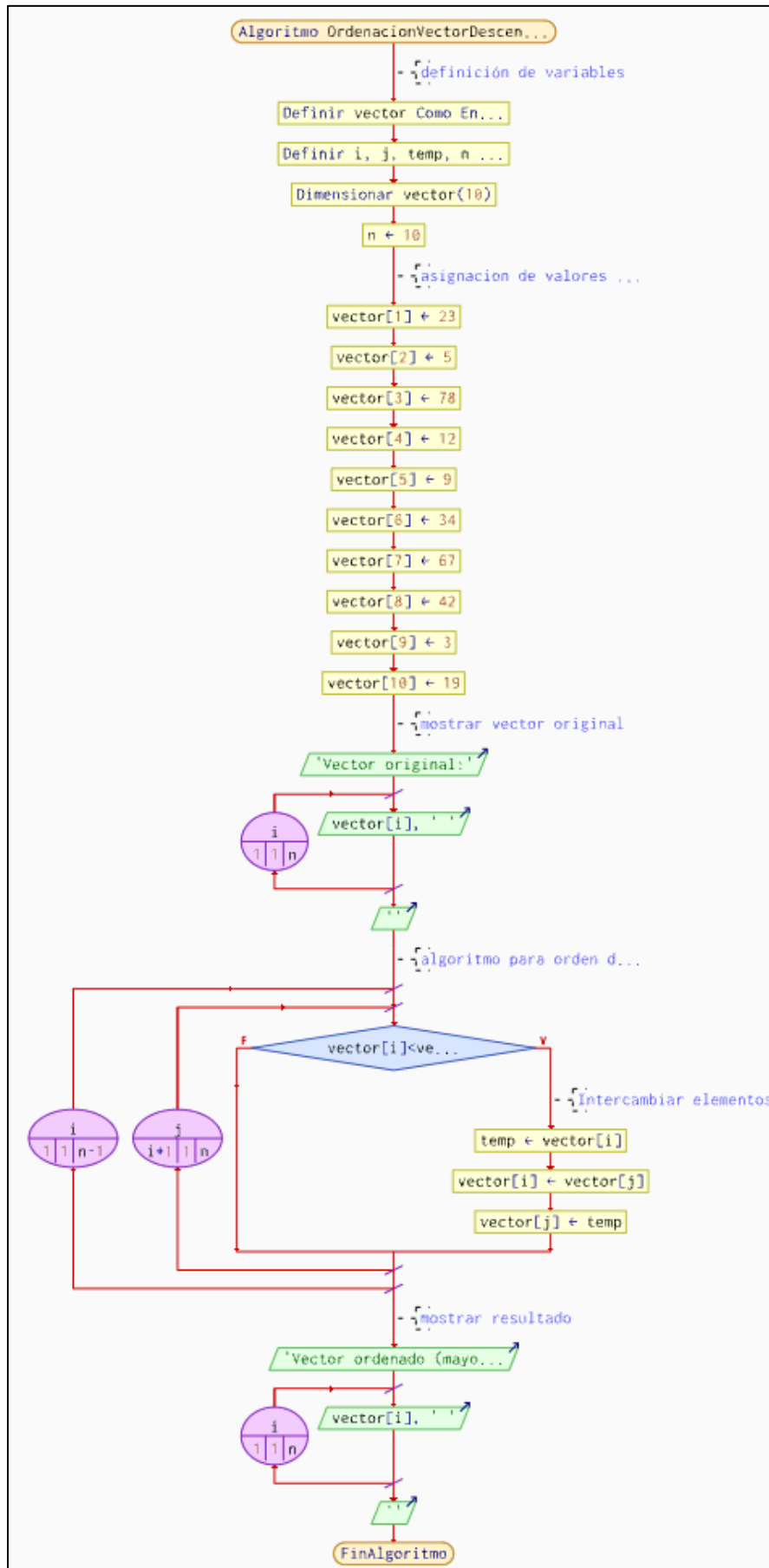
// mostrar vector original
Escribir "Vector original:"
Para i <- 1 Hasta n Con Paso 1 Hacer
    Escribir Sin Saltar vector[i], " "
FinPara
Escribir ""

// algoritmo para orden descendente por ordenación
Para i <- 1 Hasta n-1 Con Paso 1 Hacer
    Para j <- i+1 Hasta n Con Paso 1 Hacer
        Si vector[i] < vector[j] Entonces
            // Intercambiar elementos
            temp <- vector[i]
            vector[i] <- vector[j]
            vector[j] <- temp
        FinSi
    FinPara
FinPara

// mostrar resultado
Escribir "Vector ordenado (mayor a menor):"
Para i <- 1 Hasta n Con Paso 1 Hacer
    Escribir Sin Saltar vector[i], " "
FinPara
Escribir ""
FinAlgoritmo
```



- Diagrama de Flujo





- Código en C

```
Start here X Codigo-4-Segundo-Parcial-Deber-1-FP-Grupo-1.c X
1 //Universidad de las fuerzas Armadas "ESPE"
2 //GRUPO 1: ALEXANDER ALVEAR, ANTHONY CAMPOVERDE, MATEO VELECELA
3 //CODIGO 4; U2 T1
4 //ORDEN ITERATIVO (DE MAYOR A MENOR)
5
6 #include <stdio.h>
7 #include <stdlib.h>
8
9 int main(){
10     int vector[10]={23,5,78,12,9,34,67,42,3,19};
11     int i, j, temp, n=10;
12     printf("Vector original:\n");
13     for(int i=1; i<n; i++){
14         printf("%d ", vector[i]);
15     }
16     printf("\n");
17
18     for(i=0; i< n-1; i++){
19         for(j= i+1; j<n; j++){
20             if (vector[i]< vector[j]){
21                 temp=vector[i];
22                 vector[i]=vector[j];
23                 vector[j]=temp;
24             }
25         }
26     }
27
28     printf("Vector Ordenado (mayor a menor): \n");
29     for(i=0;i<n;i++){
30         printf("%d ", vector[i]);
31     }
32
33     return 0;
34 }
35
```

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

int main(){
    int vector[10]={23,5,78,12,9,34,67,42,3,19};
    int i, j, temp, n=10;
    printf("Vector original:\n");
    for(int i=1; i<n; i++){
        printf("%d ", vector[i]);
    }
    printf("\n");

    for(i=0; i< n-1; i++){
```



```
for(j= i+1; j<n; j++){  
    if (vector[i]< vector[j]){  
        temp=vector[i];  
        vector[i]=vector[j];  
        vector[j]=temp;  
    }  
}  
}  
  
printf("Vector Ordenado (mayor a menor): \n");  
for(i=0;i<n;i++){  
    printf("%d ", vector[i]);  
}  
  
return 0;  
}
```

- Prueba de Escritorio

```
C:\Users\Anthony\Desktop\codigos\Codigo-4-Segundo-Parcial-Deber-1-FP-Grupo-1.exe  
Vector original:  
5 78 12 9 34 67 42 3 19  
Vector Ordenado (mayor a menor):  
78 67 42 34 23 19 12 9 5 3  
Process returned 0 (0x0)   execution time : 0.043 s  
Press any key to continue.
```