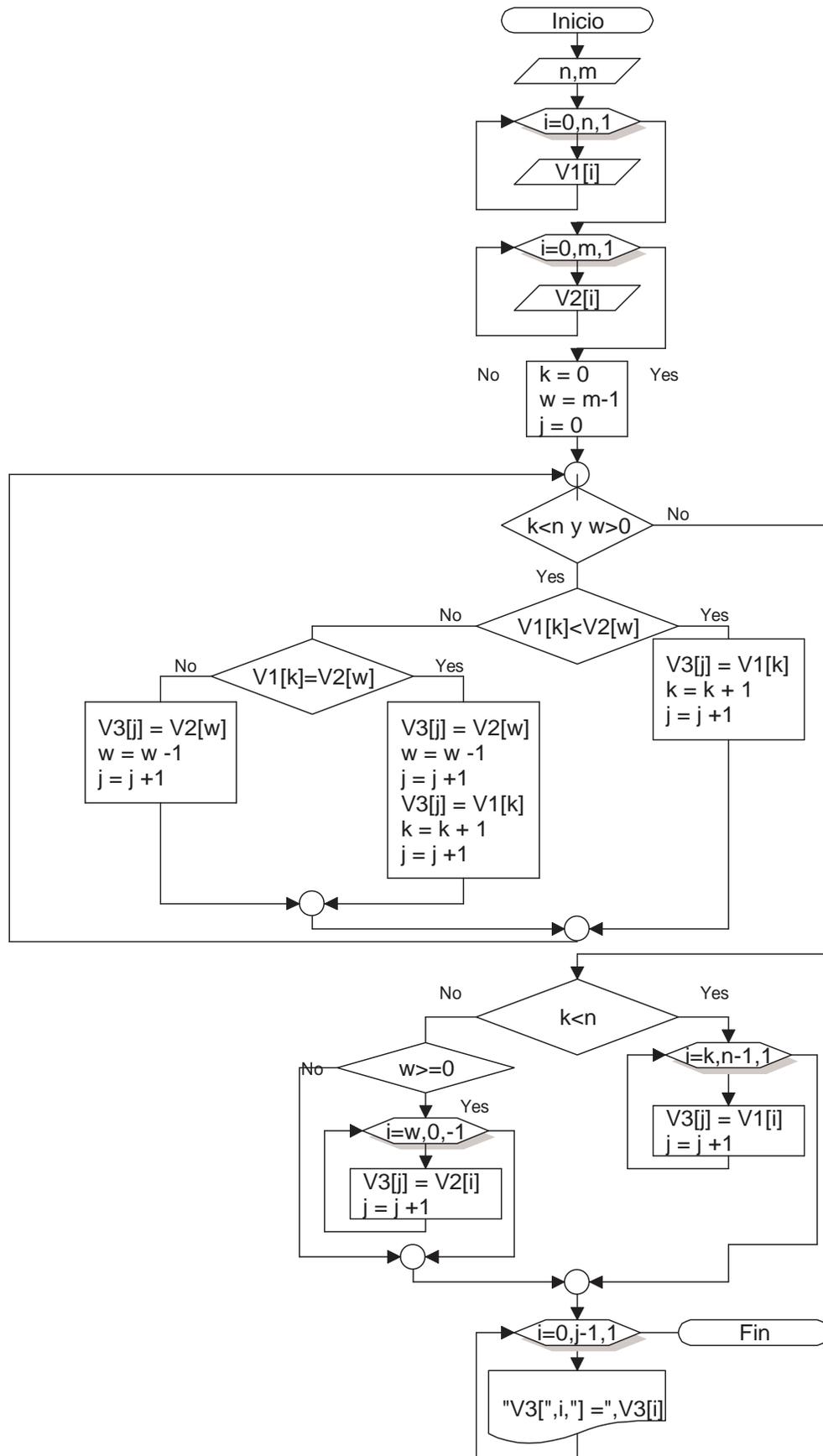


## Ejercicio 1

Formulación del Problema	Código:		Nombre		
	Enunciado				
	Elabore un programa para leer un arreglo cuyos datos se proporcionan en orden ascendente, leer un segundo arreglo cuyos datos se proporcionan en orden descendente, mezclarlos en un tercer arreglo cuyos datos aparezcan en orden ascendente, y por último, imprimir el arreglo resultante.				
Etapas de Resolución del Problema					
Fase de Resolución	Análisis del Problema				
	Descripción:	<ul style="list-style-type: none"> <li>• Leer el número de elementos del vector ordenado ascendentemente (n) y el número de elementos del vector ordenado descendientemente (m)</li> <li>• Leer los elementos de cada uno de los vectores de entrada.</li> <li>• En un ciclo repetitivo asignar los elementos al nuevo vector de forma ascendente, es decir, recorriendo el vector ascendente de arriba - abajo y el descendente de abajo - arriba, se comparan los elementos si el elemento del vector descendente es menor se asigna al nuevo vector, si son iguales se, asignan los elementos de los vectores al nuevo vector y si el elemento menor es el del vector descendente se asigna este al nuevo vector. Luego de asignar un elemento al nuevo vector se deben cambiar los índices tanto del nuevo vector, como del vector del cual se asignó el elemento.</li> <li>• Sí al final del ciclo anterior los índices de los vectores ascendentes y descendentes no indican que se han asignado todos los elementos, se deben asignar los elementos restantes que hagan falta de alguno de los dos vectores</li> <li>• Mostrar el vector resultante</li> </ul>			
	Datos de Entrada	n(# elementos vector ascendente), n(# elementos vector descendente), V1(Vector Ascendente), V2(Vector Descendente)			
	Datos de Salida	V3(Vector Resultante)			
	Datos Auxiliares	i, j(índice del vector resultante), k(índice del vector ascendente), w(índice del vector descendente)			
	Diseño y Verificación del Algoritmo				
	Formalización(Diagramas de Flujo, pseudocódigo)				



Implementación	<pre> #include&lt;stdio.h&gt; #include&lt;stdlib.h&gt; /* Declaración de constante que se utiliza para    establecer el tamaño máximo de los arreglos.*/ #define max 100  int main() {     int i,j,k,n,m,w,V1[max],V2[max],V3[max];     printf("Digite el numero de elementos del vector"            " ordenado ascendentemente: ");     /*Este ciclo válida que el tamaño del arreglo ascendente     se encuentre en el rango [1..100]*/     do{         printf("\nEl rango debe estar entre 1 y 100:");         scanf("%d",&amp;n);     }while(n&lt;1    n&gt;max);     for(i=0;i&lt;n;i++)     {         printf("\nDigite V1[%d]: ",i);         scanf("%d",&amp;V1[i]);     }     printf("\n Digite el numero de elementos del vector"            " ordenado descendientemente: ");     /*Este ciclo válida que el tamaño del arreglo descendente     se encuentre en el rango [1..100]*/     do{         printf("\nEl rango debe estar entre 1 y 100:");         scanf("%d",&amp;m);     }while(m&lt;1    m&gt;100);     for(i=0;i&lt;m;i++)     {         printf("\nDigite V2[%d]: ",i);         scanf("%d",&amp;V2[i]);     }     k=0;/*k es el índice para recorrer el vector ascendente*/     w=m-1;/*w es el índice para recorrer el vector descendente,     se inicializa en el índice del último elemento de vector,     porque se va a recorrer de abajo hacia arriba*/     j=0;/*k es el índice para el nuevo vector ascendente*/     while(k&lt;n &amp;&amp; w&gt;=0)     {         if(V1[k]&lt;V2[w])         {             V3[j] = V1[k];             j = j+1;             k = k+1;         }         else         {             /*Si los elementos de los dos vectores son iguales, se             deben asignar ambos al nuevo vector.*/             if(V1[k]==V2[w])             {                 V3[j] = V2[w];                 j = j+1;                 w = w-1;                 V3[j] = V1[k];                 j = j+1;                 k = k+1;             }             else </pre>
----------------	---

```
        {
            V3[j] = V2[w];
            j = j+1;
            w = w-1;
        }
    }
}
/*Si el valor de k al final del ciclo anterior es menor
que n, entonces no se han pasado todos los elementos del
vector ascendente al nuevo vector.*/
if(k<n)
    /*Este ciclo va desde el valor con el que quedó k y
    hasta el último elemento del vector ascendente.*/
    for(i=k;i<n;i++)
    {
        V3[j] = V1[i];
        j = j+1;
    }
else
    /*Si el valor de w al final del ciclo es mayor o igual
    que 0, entonces no se han pasado todos los elementos del
    vector descendente al nuevo vector*/
    if(w>=0)
        /*Este ciclo va desde el valor con el que quedó w y
        hasta el primer elemento del vector descendente.*/
        for(i=w;i>=0;i--)
        {
            V3[j] = V2[i];
            j = j+1;
        }
    /*Este ciclo va hasta j-1 porque ese es el número de elementos
    del nuevo vector*/
    for(i=0;i<j;i++)
        printf("\nV3[%d] = %d",i,V3[i]);
    printf("\n");
    system("pause");
    return EXIT_SUCCESS;
}
```