

Departamento de Ingeniería de Sistemas Trabajo Independiente

## Programación de Computadores

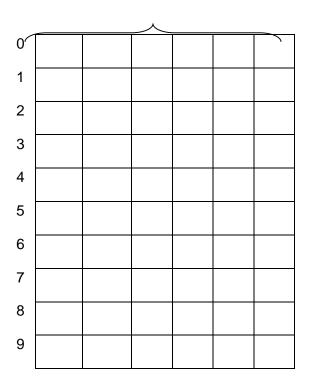
Apuntadores Memoria Dinámica Matrices-Vectores

## **Ordenamiento por Cubetas**

El ordenamiento en cubetas inicia con un arreglo unidimensional de longitud N que contiene los enteros positivos a ordenar y una matriz de diez filas por N columnas. Cada fila de la matriz es considerada una cubeta.

	9	103	27	9	17	100
--	---	-----	----	---	----	-----

N Elementos



El funcionamiento del algoritmo es muy sencillo, primero tome cada elemento del vector y basado en el número de unidades del valor colóquelo en la fila de la matriz indicada por las unidades del número. Por ejemplo el valor 97 se colocará en la fila 7 de la matriz, 3 se colocará en la fila 3 y 100 se colocará en la fila O.

Cada número que se deba ubicar en la misma fila lo hará en la columna inmediatamente disponible, por ejemplo 97 y 17 son valores que deben ubicarse en la fila 7 en la primera y segunda columna respectivamente.

Ahora debe recorrer toda la matriz por filas e ir ubicando los elementos en el arreglo unidimensional.

Se debe repetir este proceso por las decenas, centenas, millares, etc.

Ejemplo:

Ordenar el Vector de longitud N=6.

9	103	27	9	17	100

Paso	Descripción			E	jemplo	)		
1	Se debe construir una matriz de 10 filas por N columnas dinámicamente e inicialmente ubicar cada uno de los elementos del vector en la fila indicada por sus unidades.	0 1 2 3 4 5 6 7 8	100	17				
2	Ahora se debe recorrer la matriz por filas y trasladar estos valores al vector.		100 1	03 27	17	9	9	

3	Nuevamente se repite el proceso ubicando los valore							
	del vector en la matriz en la fila indicada por las	_					1	
	decenas de cada número	0	100	103	9	9		
		1	17					
		2	27					
		3						
		4						
		5						
		6						
		7						
		8						
		9						
4	Ahora se debe recorrer la matriz por filas y trasladar							
	l							
	estos valores al vector.	4.0	0 1 400					07
	estos valores al vector.	10	00 103	9	9	1	7	27
5	En este ejemplo se repite por última vez el proceso,	10	00 103	9	9	1	7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila						7	27
5	En este ejemplo se repite por última vez el proceso,	0	9	9	17	27	7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila						7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0	9	9			7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0	9	9			7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0 1 2	9	9			7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0 1 2 3	9	9			7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0 1 2 3 4	9	9			7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0 1 2 3 4 5	9	9			7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0 1 2 3 4 5	9	9			7	27
5	En este ejemplo se repite por última vez el proceso, ubicando los valores del vector en la matriz en la fila	0 1 2 3 4 5 6 7	9	9			7	27

6	Finalmente se debe recorrer la matriz por filas y trasladar estos valores al vector para obtener el vector							
	ordenado.	9	9	17	27	100	103	

Elabore un programa en C++ utilizando apuntadores, memoria dinámica para la creación de los vectores y las matrices en cada uno de los pasos, el cual debe generar el vector ordenado. El vector inicial debe ser creado dinámicamente, solicitándole inicialmente al usuario el número máximo de cifras para los elementos del vector y generarlos aleatoriamente. Por ejemplo: si el usuario digita 3 los números estarán entre 0 y 999. Debe usar librerías (.h, .cpp).

Parámetro a Evaluar		Valor (puntos)
Definición de variables	Usa nombres cortos que indican	0.5
	que va a contener cada variable y	
	ubicación dentro del algoritmo, al	
	inicio del programa.	
Captura de datos	0.5	
	para llenar el vector inicial y lo	
	hace dinámicamente	
Memoria Dinámica	Se crea dinámicamente la matriz	1
	y el vector por cada ciclo que pasa	
	de acuerdo al número de cifras en	
	los números	
Ordenamiento por Cubetas	El programa <b>hace el</b>	2
	ordenamiento correctamente	
	usando de acuerdo al enunciado	
Uso de Librerías	Uso los archivos.cpp y .h	0.5
Salida - Presentación	Muestra el vector ordenado	
	correctamente al final	0.5
	recorriéndolo y usando	
	apuntadores.	
	-	
	El programa es claro para su	
	lectura, hace la correspondiente	
	indentación.	