





OBJETIVOS:

El objetivo fundamental de esta práctica es el manejo de vectores, haciendo hincapié en operaciones como:

- Insertar un elemento
- •Eliminar un elemento
- Ordenar una lista de elementos
- •Insertar un elemento en una lista ordenada
- •Buscar un elemento en una lista ordenada
- •Borrar un elemento de una lista ordenada, etc.

CONTENIDOS:

- 1. Definición de la estructura de un vector o lista
- 2. Operaciones con listas desordenadas
- 3. Operaciones con listas ordenadas



1.- Definición de la estructura de vector

- Definición:
 - Vector DB 'abcdefghi'; inicializamos a valores diferentes cada byte
- Acceso:
 - MOV BX, offset vector; BX contiene la dirección del primer elemento del vector
 - MOV AX, [BX]; pero como AX es de 16 bits va a mover 2 bytes, para que tome únicamente el primer byte, el registro debe ser de 8 bits:
 - MOV AL, [BX]
- Direccionamiento:

INC BX MOV AH, [BX]



1.- Definición de la estructura de vector

Direccionamiento

XOR DI, DI

INC DI

MOV AH, [BX][DI]; esto es equivalente a: [BX+DI]

Otra Opción:

; Definir el vector de Word

Vector1 DW 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

; Acceso por palabra:

MOV BX, offset vector1

MOV AX, [BX];



2.- Operaciones con listas desordenadas AÑADIR ELEMENTO:

MOV BX, offset LISTA ADD BX, TAMAÑO_LISTA_EN_BYTES MOV [BX], NUEVO_ELEMENTO

BUSCAR UN ELEMENTO

MOV BX, OFFSET LISTA

MOV AL, ELEMENTO_BUSCADO

OTRO: CMP AL,[BX]

JE FIN

INC BX

JMP OTRO



2.- Operaciones con listas desordenadas BORRAR UN ELEMENTO:

Buscar y Borrar

MOV BX, OFFSET LISTA

MOV AL, ELEMENTO_ELIMINAR

XOR DI, DI; utilizamos DI para recorrer la lista

OTRO: CMP AL, [BX+DI]

JE ELIMINAR

INC DI

JMP OTRO

ELIMINAR: INC DI

MOV AL,[BX+DI]

MOV [BX+DI-1], AL

CMP DI, TAMAÑO_LISTA

JNZ ELIMINAR



3.- Operaciones con listas ordenadas ORDENAR ELEMENTO: → Método de la burbuja

Elementos	49	90	12	45	30 -	10	9	0	
				-	_				-
1ª pasada	49	90							
	49	12	90						
	49	12	45	90					
	49	12	45	30	90				
	49	12	45	30	10	90			
	49	12	45	30	10	9	90		
Fin 1 ^a	49	12	45	30	10	9	0	90	90 Ordenado
pasada									



2ª pasada	12	nes con	IIStas	Juden	daas	rî	193		
		45	49						
			30	49					
				10	49				
					9	49			
Fin 2 ^a pasada	12	45	30	10	9	0	49	90	49 Ordenado
3ª pasada	12	45							
	12	30	45						
	12	30	10	45					
	12	30	10	9	45				
Fin 3 ^a	12	30	10	9	0	45	49	90	45 ordenado



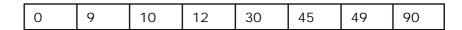
3.- Operaciones con listas ordenadas

4ª pasada	12	30					-00		
	12	10	30						
	12	10	9	30					
Fin 4ª pasada	12	10	9	0	30	45	49	90	30 ordenado
5ª pasada	10	12							
	10	9	12						
Fin 5ª pasada	10	9	0	12	30	45	49	90	12 ordenado
6ª pasada	9	10							
Fin 6 ^a pasada	9	0	10	12	30	45	49	90	10 ordenado
Ultima pasada	0	9	10	12	30	45	49	90	9 ordenado

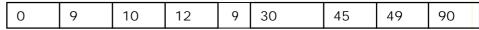


3.- Operaciones con listas ordenadas BUSCAR

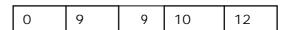
Por ejemplo: Buscar el elemento 9 en la lista anterior:



1.Se divide la lista



- 1.Se compara con el elemento, en este caso con 30: ¿9>30?. NO
- 2.Se toma la primera parte de la lista



- 1.Se compara con 10. ¿9>10?. NO
- 2.Se toma la primera parte de la mitad de lista

1 ' 1 '

6. Se compara con 9 y termina.



3.- Operaciones con listas ordenadas

INSERTAR / ELIMINAR ELEMENT

- •Determinar el lugar dentro del vector donde debe incluirse el elemento.
- •Dejar esa posición libre moviendo hacia posiciones mas altas de memoria todos los elementos de valor más alto (se parte de que la lista está ordenada de menor a mayor)
- •Se inserta el elemento en la posición que ha quedado vacía.
- •Se añade 1 al número de elementos