

Práctica I - Arrays, funciones punteros, struct, ordenamientos y búsquedas

1. Escribe un programa que almacene, en un vector con dimensión=10, los números del 1 al 10.
2. Escribe un programa con un menú que defina las siguientes funciones y emita los resultados, la dimensión del vector será de 10 (elementos).
 - Leer m elementos de un vector por teclado. Deberá considerar que no podrá superar el máximo de elementos de vector.
 - `void leer(int vect[], int num);`
 - Leer elementos de un vector hasta encontrar el número entero a . Deberá retornar el número de elementos que ha leído sin contar el entero a .
 - `int leerMarca(int vect[]);`
 - Dado un elemento y dado un vector de enteros, realice una función que devuelva el número de apariciones del elemento en el vector.
 - `int veces(int valor, int vect[], int num);`
 - Invertir los elementos del vector sin utilizar otro vector. Por ejemplo, el vector formado por los enteros: 1 2 3, debe quedar 3 2 1.
 - `void invertirOrden(int vect[], int num);`
3. Continúa agregando funciones al ejercicio anterior que permitan:
 - a. Calcular y emitir la suma de sus elementos.
 - b. Calcular y emitir el mínimo del vector.
 - c. Calcular y emitir el promedio de los valores del vector
 - d. Emitir los valores de aquellos que superaron ese promedio.
 - e. Emitir los valores del vector que son múltiplos del último número ingresado en el mismo.
 - f. Emitir el valor máximo e indicar la cantidad de veces que apareció y el número de orden en que fue ingresado.
 - g. Emitir los valores que son pares.
 - h. Emitir los valores que son impares.
 - i. Emitir aquellos que estén ubicados en posición par.
4. Supongamos que disponemos de un conjunto de datos asociados a la velocidad de cierta reacción bajo diferentes condiciones de temperatura y presión. Se desea establecer una representación gráfica (histograma) de esos datos de la forma siguiente: dados los datos $V = \{4,3,1,2\}$ el gráfico será:
Cond 1: * * * *
Cond 2: * * *
Cond 3: *
Cond 4: * *
5. Escribe un programa que, a partir de un vector vacío de enteros, permita insertar nuevos números en posiciones válidas del mismo. El programa emitirá repetidamente al usuario un menú con cuatro opciones:
 - ✓ Insertar delante: Esta opción agrega un número en la primera posición del vector. Deberá desplazar, si fuese necesario, el resto de elementos una posición a la derecha. Si el vector estuviese lleno, se perdería su último elemento.
 - ✓ Insertar detrás: agrega un nuevo elemento al final del vector, es decir, en la última celda. Si el vector estuviese lleno, el elemento se inserta en la última celda del vector y se perdería el elemento que estaba allí anteriormente.
 - ✓ Insertar en una posición dada: Dado un índice del vector, se debe insertar un nuevo elemento en dicha posición, siempre que el nuevo elemento sea contiguo a los ya existentes. Por ejemplo, si en el vector hay tres elementos (que ocupan las posiciones 0, 1 y 2) no se permite agregar un elemento en la posición 7 porque no forma una secuencia continua, pero sí se permite en la posición 3. Por otra parte, si la posición dada está ocupada, los elementos se deben desplazar una posición a la derecha y, si el vector estaba lleno, se pierde el último elemento.
 - ✓ Finalizar el programa.

6. Se ingresa en un vector, 10 números enteros. La computadora muestra los valores, eliminando las repeticiones.
7. Dadas las siguientes declaraciones:
`int x, int_array[MAX]; x = int_array[4];`
explica qué traduce el compilador para acceder al cuarto elemento del arreglo y asignárselo a x.
8. ¿Qué sucede si declaras un vector de la siguiente manera: `int vec[6];` y luego ejecutas: `vec[6]= 78;`
9. Verifica el siguiente código. Obtén conclusiones.
- ```
int n=3;
int edad, notas[n];
edad=20; notas[n]=99;
printf("Edad %d", edad);
```
10. ¿Que está mal en el siguiente fragmento de código?
- ```
int b[10];
for (i=0; i<=10; i++) b[i]=b[i+1];
```
11. ¿Qué está mal en el siguiente fragmento de código?
- ```
int b[10];
for (i=0; i<10; i++) B[i]=b[i+1];
```
12. Construye un programa, utilizando funciones, que lea como máximo 10 números enteros, el ciclo termina al introducir un 0. A continuación lea otro entero y compruebe si ese entero está o no entre los anteriores. En caso de que se encuentre, contar y emitir el número de veces que aparece. Utiliza ciclo while.
13. Escribe un programa que llene un vector de 50 elementos con los números de Fibonacci. Los primeros números son 1, 1, 2, 3, 5 (cada número es la suma de los dos anteriores). Emite el contenido del vector por pantalla.
14. Dado un vector de dimensión N y un valor x float, almacenar x en la posición i de un vector. Los valores no se ingresan ordenados por posición, se debe ingresar la posición y el valor de x hasta completar el vector. Al finalizar, emitir el contenido del vector indicando la posición ocupada por cada número a partir de la primera posición. Utiliza el ciclo while y funciones.
15. Comenta el siguiente programa:
- ```
int v[5];
int *p;
...
p = &v[0];
*p = 1;
*(p+1) = 2;
*(p+2) = 3;
```
16. Escribe un programa que imprima los elementos de un vector de enteros en orden inverso utilizando punteros (no utilizar subíndices []).
- ```
int v[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p;
```
17. Escribe una función que reciba un vector de enteros y su tamaño, y retorne la cantidad de números impares que contiene. Trabaja con notación de punteros y utiliza ciclo while.
- ```
int impares (int *v, int tam);
```
18. Escribe una función que devuelva un puntero al máximo valor de un vector de "doubles". Si el vector está vacío debe retornar NULL.

```
double* max (double *v, int tam);
```

19. Dadas las siguientes declaraciones:

```
int v[3] = {10,20,30};  
int *p;  
p = v;
```

Explica que imprimiría el printf en cada caso de los siguientes:

- a) `(*p)++; printf ("%d", *p);`
- b) `*(p++); printf ("%d", *p);`
- c) `*p++; printf ("%d", *p);`

20. Qué emite el siguiente programa?:

<pre>#include <stdio.h> main() { int x[3], *puntero; x[0]=10; x[1]=20; x[2]=30; puntero = x; puntero = &x[0]; printf("%d\n\n",puntero[0]);</pre>	<pre>printf("%d\n\n",*puntero); printf("%X\n\n",&puntero); printf("%X\n\n",&puntero[1]); printf("%d\n\n",puntero[1]); printf("%d\n",*(puntero+1)); printf("%d\n",*(puntero+2)); system("pause");}</pre>
--	---

21. Construye una función tal que dados dos vectores de 5 elementos cada uno, los concatene en un tercer un vector de 10 elementos. Ej:

V1= 2-56-7-8-30; V2= 7-80-2-4-13; V3= 2-56-7-8-30-7-80-2-4-13;

22. Ídem anterior, pero los elementos de los dos vectores deben emitirse intercalados. Ej:

V4= 2-7-56-80-7-2-8-4-30-13;

23. Se ingresan los N y M elementos de los arreglos unidimensionales A y B, respectivamente. La computadora emite la unión de ambos, su diferencia y su intersección.

24. Escribe un programa para simular 10 sorteos de una lotería. Se deben elegir 6 números distintos, del 1 al 54 al azar. Una salida en pantalla típica podría ser: "los seis números son: 47, 3, 18, 24, 10, 51."

25. Para probar un congelador, la fábrica registra en un listado la temperatura en el interior durante todos los días del mes de junio. Escriba una función que reciba un vector con todas estas temperaturas (generalmente, negativas) y devuelva la mínima temperatura. Luego escriba una segunda función que diga en qué día del mes se produjo la temperatura mínima.

26. Aritmética de punteros:

```
int B[] = {3,4,1,2,7,12,-4};  
float f = 4.234, *ptf;  
*(B+3) = *B + 15;  
ptf = &f;  
*B = (int) (*ptf);  
f = *ptf + 20;  
*(B + 5) = (int) (*ptf); // que emite por pantalla B[], f, ptf?
```

27. Determina la salida del programa:

<pre>#include <stdio.h></pre>	<pre>#define SIZE 5</pre>
-------------------------------------	---------------------------

```

void mystery (int *, int *);

main () {
    int i;
    int x[SIZE] = {2,4,6,8,10};
    int y[SIZE] = {1,3,5,7,9};
    int *xPtr = NULL;
    int *yPtr = NULL;
    mystery (x, y);
    for (i=0; i<SIZE; i++) {
        printf ("%d\t", x[i]); }
    printf ("\n");
}

```

```

for (i=0; i<SIZE; i++) {
    printf ("%d\t", y[i]); }
printf ("\n");
return 0;
}

void mystery (int *n1, int *n2)
{
    int i;
    for (i=0; i<SIZE; i++) *(n1+i)
        = 2 * *(n2+i);
}

```

28. Considerando las siguientes declaraciones y sentencias:

```

int array[]={1,2,3,4,5,6};
int *puntero, x;
puntero = array;
puntero++;
*puntero=*puntero+6;
puntero=puntero+3;
puntero=puntero-puntero[-2];
x=puntero-array;

```

- a) ¿Cuál es el valor de x?
- a. 1, 2, 3 ó 4?
- b) ¿Cual es el valor de array[1]?
- a. 2, 4, 6 ú 8?

29. Determina el tamaño del dato double. Ahora: Si un array empieza en la posición 200 de memoria, dado el siguiente código:

```

double valores[5] = {2.0, 4.0, 6.0, 8.0, 10.0};
double *valoresPtr = valores;

```

Haz un cuadro en forma de tabla de cómo estará la memoria en cuanto a contenidos y direcciones.

a. ¿Que es valoresPtr?. b. ¿Qué contiene *valoresPtr?

30. Dadas las siguientes variables:

```

int a = 10, b = 100, c = 30, d = 1, e = 54;
int m[10] = {10,20,30,40,50,60,70,80,90,100};
int *p = &m[3], *q = &m[6];

```

Evalúa, sin usar un compilador, las siguientes expresiones. Considera que los resultados de cada una de las expresiones no influyen en las siguientes:

- $a + m[c/a] + b-- * m[1] / *q + 10 + a--;$
- $a + (b * (c - d) + a) * *p++;$
- $m[d] - d * e + (m[9] + b) / *p;$
- $b++ * c-- + *q * m[2] / d;$
- $(b/a) * (m[3] * ++e);$
- $++*p+++*q;$
- $++*p + ++*q;$
- $m[c/a]-*p;$
- $q[-3] + q[2];$

31. Construye un programa que a partir de dos vectores de enteros ordenados crecientemente, cree un tercer vector, también con los datos ordenados en forma creciente. Los dos vectores que se pretenden fusionar no tendrán elementos repetidos en sí mismos, pero entre ellos pueden tener elementos comunes. En este caso, no debe haber repeticiones en el vector que resulte de su fusión.
32. Se dispone de un vector de números enteros de tamaño N, ordenados en forma creciente. Se desea conocer si un número dado introducido desde el teclado se encuentra en la lista. En caso afirmativo, averiguar su posición y emitirla por pantalla. En caso negativo se desea insertarlo en la posición adecuada y posteriormente mostrar la posición por pantalla. Utilizar funciones y un menú.
33. ¿Qué diferencia hay entre el nombre de un array o arreglo y un puntero?
34. ¿Cómo sabe el compilador el tamaño de un objeto al que apunta un puntero?
35. Escribe funciones recursivas para calcular:
 - a) la suma de los contenidos de un vector de números y emite el vector y el resultado.
 - b) El producto de los elementos del vector
 - c) Que emita el número menor
 - d) Que emita el número mayor
36. El prototipo de una función que tiene como parámetro de entrada/salida un entero, como parámetro de salida una cadena de caracteres y como parámetro de entrada/salida un vector de 20 reales es:

Nota: Pint es un puntero a entero.

- o `void func (Pint, Cadena &, * float);`
- o `void func (Pint, Cadena, float []);`
- o `void func (int &, Cadena, float [20]);`
- o `void func (Pint, Cadena[], float []);`
- o `void func (int *, Cadena *, float[20]);`

37. Sea el siguiente código:

```
void desplaza (int t[], int *pos)
{
    t[*pos]=0;
    if (*pos < 9)
        *pos++;
    else
        *pos=0;
    t[*pos]=1;}

```

Si la tabla t contiene los valores {0,0,0,0,0,0,0,0,0} y la variable entera posición tiene el valor 9, ¿cuál será el contenido de la tabla t tras la llamada desplaza (t, &posicion); ?

- o Se produce un error de compilación
- o {1,0,0,0,0,0,0,0,0}
- o {1,0,0,0,0,0,0,0,1}
- o {0,0,0,0,0,0,0,0,1}
- o {0,0,0,0,0,0,0,0,0}

38. Escribe un programa en el que se defina un arreglo de 10 punteros a float, se lean diez números en las ubicaciones en las que hacen referencia los punteros. Se sumen todos los números y se almacene el resultado en una dirección a la que haga referencia un puntero. El programa deberá mostrar el contenido de todas las variables, tanto los punteros como los números de tipo float.

Arrays, funciones, punteros y struct

39. Un listado contiene nombres de clientes, cantidades de un producto que han comprado y precio unitario. Ejemplo:

```
Álvarez 12 30
Barrios 25 23
```

Escribe un programa que cargue los datos y los emita. Para los datos anteriores de ejemplo, la primera línea de este reporte será: Álvarez 12

40. Escribe un programa que muestre un listado que contenga código de artículo y la cantidad de unidades que se vendieron de ese artículo. Calcula la cantidad total vendida.
41. Un listado contiene nombre y nota de un grupo de hasta 30 alumnos (pueden ser menos). El profesor usa un método extraño para ajustar las calificaciones: si hubo menos de 5 alumnos con 9 o más, a cada alumno se le suma 1 punto a su nota; en caso contrario, a cada alumno se le suman 0,5 puntos. Ten en cuenta que ningún podrá tener una calificación superior a 10 puntos. Diseña un programa que muestre un listado conteniendo: nombre del alumno, nota original y nota ajustada.
42. Escribe un programa de sueldos para una pequeña compañía que tiene seis empleados. Para cada empleado/a, el programa recibe desde el teclado el nombre, sexo, horas trabajadas y sueldo por hora. Por cada empleado ingresado, el programa deberá mostrar en pantalla el nombre y el sueldo de cada empleado. Posteriormente, deberá informar el total de pagos para cada sexo, y el sueldo promedio de hombres y de mujeres.
43. Calcular la media aritmética de las notas pertenecientes a **10** materias de un curso de 8 alumnos. Para cada curso deberá indicar la media obtenida.

```
void leerNotasCurso(float notas[]);

float Media_Materias_Curso(float notas[]);
```

44. Escribe un programa que permita calificar a un grupo de diez alumnos de la escuela secundaria. Por teclado se ingresan el nombre y las tres calificaciones de cada alumno, y con esos datos el programa deberá calcular e informar el promedio de cada alumno y decir si está aprobado o no; para aprobar se requiere un promedio de seis o más y haber obtenido al menos seis en la última de las tres calificaciones. Posteriormente el programa deberá informar cuántos alumnos aprobaron y cuántos obtuvieron un promedio de al menos 8 puntos.
45. Una organización de investigación del consumidor tiene un listado de datos que contiene los nombres de las tiendas y los precios que cobran por una grabadora de DVD en particular. Escribe un programa para determinar el precio más bajo que se cobra por una grabadora y luego imprime dos listas:
- *Lista 1: Los nombres de todas las tiendas que cobren el precio más bajo.*
 - *Lista 2: Los nombres y precios de todas las tiendas cuyo precio no exceda al mínimo en más de \$15.*
46. En un centro médico, cada ficha de papel contiene el nombre de un doctor, especialidad (ej. Cardiología o pediatría) y estado de disponibilidad (0 para disponible, 1 para ocupado). Escriba un programa que luego de cargar por teclado los datos de las tarjetas, asigne un médico a un paciente como se indica:
- ✓ Se introducen el nombre de paciente y su preferencia de especialidad
 - ✓ Primero se intenta asignar un medico con la especialidad que se pidió. Si hay más de un médico de la especialidad pedida, se asigna solo uno al azar (no necesariamente el primer médico disponible de la lista). De lo contrario, se asigna un médico al azar elegido entre los disponibles.

- ✓ Considerar que se ingresa más de un paciente. Cuando se asigna un médico a un paciente, este médico deja de estar disponible.

DIA	VENTAS ESE DÍA	TOTAL A LA FECHA	PROMEDIO ACTUAL DE 5 DÍAS
1	10	10	
2	60	70	
3	21	90	
4	20	110	
5	35	145	29.0
6	40	185	35.0
.	.	.	.
15	23	.	.

47. Un comercio posee un listado con los totales diarios vendidos, en un período de 15 días. Escribe un programa que genere una tabla -similar a la que se presenta en el ejemplo- (la columna promedio, será cargada con el promedio de los últimos 5 días).

48. Escribe un programa para manipular un vector de registros. Cada registro debe tener campos para: legajo, apellido, nota. Los ingresos deben realizarse de tal modo que el ingreso de un nuevo registro debe ser ordenado, es decir, insertando el registro en lugar que corresponda y realizando, si es necesario, un corrimiento de los otros registros. Análogamente, el borrado se realizará corriendo los registros posteriores al registro a borrar.
49. Se ingresa un vector de registros; cada uno de ellos contiene nombre, categoría, y un campo con los tres últimos sueldos del empleado. Se quiere saber el nombre y la categoría de quien haya obtenido el mayor promedio de sueldos (considerar emitir a) solo el que aparezca por primera vez; b) todos los nombres y sueldos en caso de que coincidan los promedios)

Arrays unidimensionales, funciones, struct, ordenamientos y búsquedas

50. Define una función que reciba un vector de enteros V y su tamaño TAM y devuelva si V está ordenado o no.

51. Supone que el vector b contiene los números:

14	26	18	25	40	32	12	6	49
----	----	----	----	----	----	----	---	----

Cuál será el contenido de ese vector luego de la ejecución del siguiente ciclo for?

```
for (i=0;i<7;i++)
    if ( b[i] > b[i+1] )
    { temp=b[i+2];
      b[i+2]=b[i+1];
      b[i+1]=temp; }
```

52. Se ingresa un vector de 20 float y la computadora los ordena (utiliza un menú)

- por selección
- por burbujeo.
- por burbujeo con bandera
- por inserción

53. Muestra el contenido del vector siguiente después de cada pasada completa de un ordenamiento por burbuja común:

Pasada					
	1	2	3	4	5
A[0]=4					
A[1]=3					
A[2]=2					
A[3]=5					
A[4]=1					

54. Un procedimiento tiene que devolver ordenada una tabla *t* de registros de tipo *Treg* por el campo *c3*. La sentencia de comparación entre dos elementos de la tabla dentro del procedimiento es:
- `if (t[i]->c3 < t[j]->c3)`
 - `if (t[i].c3 < t[j].c3)`
 - `if (strcmp(t[i].c3,t[j].c3)<0)`
 - `if ((*t[i]).c3 < (*t[j]).c3)`
 - `if (*(t[i].c3) < *(t[j].c3))`
55. Se desea comparar el funcionamiento de los diferentes métodos de búsqueda. Para ello se pretende buscar el número 27 en el siguiente vector: 1 4 5 12 25 27 31 42 43 56 73 76 78 80 99. Indica los pasos que seguirían los siguientes algoritmos:
- a) Búsqueda secuencial.
 - b) Búsqueda binaria.
56. Escribe un programa para ordenar un vector de 1000 números. Además de imprimir la lista ordenada, el programa debe determinar si hubo repetición en el número máximo. Si lo hubo, deberá emitir cuántas veces se presentó el número más grande.
57. La mediana de un vector ordenado se puede definir así:
- ✓ Para un número impar de elementos (ej. vector de 5 posiciones), es el valor del elemento que está en la casilla central.
 - ✓ Para un número par, es el promedio de los que se hallan en las casillas adyacentes a la mitad.
- Así, la mediana de un vector ordenado de 25 números es el número que se encuentra en la casilla 13, mientras que para un vector de 24 casillas, es el promedio de los valores situados en las casillas 12 y escribe una función para encontrar la mediana de un vector (ya ordenado) y pruébala.
58. La moda o valor modal de una lista de números se define como aquel número que tiene el máximo valor de apariciones en la lista. Por ejemplo para el conjunto: 72,72,72,75,75,75,78,82,82,82,82,85,85,85,89, la moda es el valor: 82. Escribe una función que determine el valor modal para un vector de enteros ordenado.
59. Buscar el elemento máximo de un vector. Además se desea saber la posición que éste elemento ocupa en el vector. En caso de que exista más de un máximo, quedarse con el último encontrado. ¿Qué modificaciones tendría que hacer para quedarse con el primero?.
- `int buscar(int *pos, int vect[], int num);`
 - Insertar un elemento considerando que el vector está ordenado ascendentemente. (Inserción en orden).
 - `int insertarOrden(int elem, int vec[], int *num, int *pos);`
 - Insertar elementos en un vector desordenado.
 - `int insertarDesorden(int elem, int vect[], int *num);`
 - Eliminar un elemento dado del vector. (Buscar, luego eliminar).
 - `void eliminar(int* elem, int vect[], int *num);`