

Práctica de arrays, punteros, funciones y struct

1. Escribe un programa que almacene en un vector los números pares del 1 al 10.
2. Implementa un programa que permita emitir invertido un vector. Es decir, si $v=\{1,2,3,4,5\}$, entonces se mostrará $v = \{5,4,3,2,1\}$
3. Define funciones que permitan, dado un vector de enteros V y su tamaño TAM:
 - a. Calcular la suma de sus elementos.
 - b. Calcular el máximo del vector
 - c. Calcular el mínimo del vector
 - d. Calcular el promedio de los valores del vector
 - e. los valores de aquellos que superaron ese promedio.
 - f. Indicar cuántas veces aparece el elemento k en V
 - g. los que son múltiplos del último número ingresado
 - h. el valor máximo e indica la cantidad de veces que apareció
 - i. el número de orden en que fue ingresado.
 - j. aquellos que son pares.
 - k. aquellos que estén en posición par.
 - l. Aquellos que son impares.
4. Se ingresa un vector de 10 números enteros La computadora lo muestra eliminando las repeticiones.
5. Dado un listado de valores numéricos i y x, donde i es un entero mayor que 0 y x un float, almacenar x en la posición i de un vector. El listado no se ingresa ordenado por posición. Informar la cantidad de números y el contenido del vector indicando la posición ocupada por cada número a partir de la primera posición. Utiliza el ciclo while y funciones.
6. Verifica el siguiente código. Obtené conclusiones.

```
int edad, notas[3];  
int n;  
n=3; edad=20; notas[n]=99;  
printf("Edad %d", edad);
```

7. ¿Que está mal en el siguiente fragmento?

```
int b[10];  
for (i=0; i<10; i++) b[i]=b[i+1]
```

8. ¿Qué sucede si declaras un vector de la siguiente manera: `int vec[6];` y luego se ejecutas: `vec[6]= 78;`
9. ¿Qué está mal en el siguiente fragmento?

```
int b[10];  
for (i=0; i<10; i++) B[i]=b[i+1];
```

10. Construye un programa utilizando funciones que lea una serie de enteros (como máximo 10; 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 el número de veces que aparece. Utiliza ciclo while.
11. 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 resultado en pantalla.
12. Comenta el siguiente programa:

```
int v[5];  
int *p;  
...  
p = &v[0];  
*p = 1;  
*(p+1) = 2;  
*(p+2) = 3;
```

13. Escribe un programa que imprima los elementos de un vector de enteros en orden inverso utilizando punteros (no usar subíndices []).

```
int v[10] = {1,2,3,4,5,6,7,8,9,10};  
int *p;
```

14. Escribe una función que reciba un vector de enteros y su tamaño, y devuelva la cantidad de números impares que contiene. Trabaja con notación de punteros y utiliza ciclo while.

```
int impares (int *v, int tam);
```

15. 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);
```

16. 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);

17. 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>
---	---

18. Construye una función tal que dados dos vectores de 5 elementos cada uno, los concatene para formar 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;

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

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

20. Se ingresan los N y M elementos de los arreglos unidimensionales A y B, respectivamente. La computadora muestra su unión, su diferencia y su intersección.

21. Se dispone de una lista de números enteros clasificados en orden creciente. Se desea conocer si un número dado introducido desde el teclado se encuentra en la lista. En caso afirmativo, averiguar su posición y mostrarla por pantalla. En caso negativo se desea insertarlo en la posición adecuada y posteriormente mostrar la posición por pantalla.

22. 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?
```

23. Determina la salida del programa:

```
#include <stdio.h>
#define SIZE 5
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);
    }
```

24. Considerando las siguientes declaraciones y sentencias:

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

a) ¿Cuál es el valor de x?

☐ 1 ☐ 2 ☐ 3 ☐ 4

b) ¿Cual es el valor de array[1]?

☐ 2 ☐ 4 ☐ 6 ☐ 8

25. 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?

26. 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];`

27. 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.

28. 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.

29. 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.

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

31. Supone que el vector b contiene los números:

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

Cual 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; }
```

32. En un centro medico, 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 mas 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. El programa puede considerar que un médico que ya se ocupo permanece en ese estado.

33. Se ingresa un vector de 20 float y la computadora los ordena

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

34. Muestra el contenido del vector siguiente después de cada pasada completa de una ordenación por burbuja común:

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

35. 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. Indique los pasos que seguirían los siguientes algoritmos:
- Búsqueda secuencial.
 - Búsqueda binaria.
36. Da un ejemplo de un vector de 100 elementos en el que, como mucho, dos elementos estén fuera de su posición definitiva, pero que requiera 99 pasadas de ordenación por burbuja para quedar completamente ordenado.
37. 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 imprimir cuantas veces se presentó el número mas grande.
38. 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 esta 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 numero 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.
39. La moda o valor modal de una lista de números se define como aquel numero que tiene el máximo valor de apariciones en la lista. Por ejemplo para el conjunto: 72,72,72,75,75,75,78,8,82,82,82,85,85,85,89, la moda es el valor: 82. Escriba un procedimiento que determine el valor modal para un vector de enteros ordenado.
40. Implementa la potenciación escribiendo una función recursiva llamada potencia que reciba una base x y un exponente n, donde x es float y n unsigned int. Tenga en cuenta que:

$$x^n = x^{n-1} \quad \text{si } x > 0$$

$$= 1 \quad \text{si } x = 0$$

41. Escribe una función recursiva para calcular la suma de los contenidos de un vector de números y pruébala.
42. Escribe un programa que muestre una tabla que contenga código de artículo y la cantidad de unidades que se vendieron en total.
43. 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 el promedio de los últimos 5 días).

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	.	.

44. Una tienda vende 5 artículos diferentes, cuyos códigos son 101, 102, 103, 104 y 105 respectivamente. Se desea obtener un listado conteniendo código de artículo y unidades vendidas por día de lunes a viernes, (ten presente que el mismo artículo se vende más de una vez, por tanto aparecerá más de una vez en el listado, con diferentes cantidades vendidas).
45. 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.
46. 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.
47. 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.*
48. 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."
49. 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.
50. 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 una tabla de 20 reales es:

Nota: Pint es un puntero a entero.

- ☐ void func (Pint, Cadena &, * float);
- ☐ void func (Pint, Cadena, float []);
- ☐ void func (int &, Cadena, float [20]);
- ☐ void func (Pint, Cadena[], float []);
- ☐ void func (int *, Cadena *, float[20]);
51. Un listado contiene nombres de clientes y cantidades que han comprado. Puede haber varias entradas para algunos clientes pero están agrupados ya que la lista es alfabética.

Alvarez 12

Álvarez 10

Barrios 25

Escribe un programa que llame a `INFORMA` para leer todos los datos y luego llame a `IMPRIMIR_REPORTER` a imprimir el nombre de cada cliente y su compra total. Para los datos anteriores de ejemplo, la primera línea de este reporte será: Álvarez \$22

.....
.....

52. 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.
53. ¿Qué diferencia hay entre el nombre de un array o arreglo y un puntero?
54. ¿Cómo sabe el compilador el tamaño de un objeto al que apunta un puntero?
55. 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 `{0,0,0,0,0,0,0,0,0,0}` y la variable entera posición tiene el valor 9, ¿cuál será el contenido de la tabla `t` tras la llamada `desplaza (t, &posicion)`?

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

56. Las funciones siguientes pueden, por lo general, utilizar vectores parcialmente, debiendo pasar como parámetro una longitud **num** ≤ **nmax**. Nótese que el interés por el cual algunos enteros se pasan por referencia (con punteros) es el de modificar su valor puesto que el vector se modifica (longitud, elementos, etc.). Los vectores serán declarados con una longitud máxima **nmax=10** (elementos).
- Leer *m* elementos de un vector por teclado. Deberá considerar que no podrá superar el máximo de elementos de vector (**num**).
 - `void leer(int vect[], int num);`
 - Leer elementos de un vector hasta encontrar el número entero *a* (secuencias con marca). Deberá retornar el número de elementos que ha leído sin contar el entero *a*. No podemos leer más de *nmax* enteros.
 - `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);`
- 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[t, 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);`

57. 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.

58. 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[]);
```

59. 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.
60. 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)
61. 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))

62. Un bucle para recorrer una tabla bidimensional de 20 por 30 enteros sería de la forma:

- ☐ for (i=0;i=20;i++) for (j=0;j=30;j++)
- ☐ for (i=0;i==20;i++) for (j=0;j==30;j++)
- ☐ Ninguna de las restantes respuestas es correcta
- ☐ for (i=0;i<=20;i++) for (j=0;j<=30;j++)
- ☐ for (i=0;i<20;i++) for (j=0;j<30;j++)

63. Analiza y comenta el siguiente programa:

```
#include <stdio.h>

#define NUMALUMNOS 10
#define NUMASIGNATURAS 3

/*prototipo de la función que calcula la nota media; recibe por parámetro la fila
de la matriz que corresponde a la asignatura cuya nota media se va a calcular*/
double calculoMedia(double *);

main() {
    double
    notas[NUMASIGNATURAS][NUMALUMNOS]={ {2.5,3.5,5.5,10,6.8,3.0,6.0,7.0,8.5,9.0},
    {5.5,1.5,5.5,10,6.8,3.0,6.0,7.0,5.5,9.0}, {5.5,5.5,5.5,10,9.8,3.0,4.0,
    7.0,8.5,3.0}}; int i;

    for (i=0; i<NUMASIGNATURAS; i++)

    printf("La nota media de la signatura numero %d es %4.2f\n", i+1,
    calculoMedia(notas[i]));}

double calculoMedia(double *notasAsignatura){
    double media=0.0;
    int i;
    media+=notasAsignatura[i];
    media/=NUMALUMNOS;
    return media; }
```

64. Se ingresa una matriz de NxM componentes. La computadora indica:

- a. la cantidad de ceros que contiene
- b. el valor máximo de cada columna.
- c. el elemento mínimo de toda la matriz y las posiciones en que aparece.
- d. el promedio de la matriz
- e. el promedio de cada columna
- f. los promedios de cada fila, e indica qué filas tienen el promedio máximo.
- g. el menor y mayor elemento de esa matriz y sus posiciones.
- h. devuelva el número de elementos negativos, iguales a cero y positivos que hay en ella.

65. Se ingresan dos matrices de $N \times M$ componentes enteras. La computadora las suma y muestra la matriz resultante
66. Se ingresan dos matrices (de $N \times M$ y $M \times K$ componentes enteras). La computadora muestra su producto.
67. Se ingresan dos matrices. La computadora indica los elementos comunes a ambas y su ubicación.
68. Se ingresa una matriz de $N \times M$ componentes enteras. La computadora muestra las sumatorias de las columnas y los emite ordenados de menor a mayor.
69. Se ingresa una matriz de $N \times N$ componentes. La computadora calcula y muestra:
- La traza de la matriz (suma de los elementos de la diagonal principal)
 - La suma de los elementos de la otra diagonal
70. Se ingresa una matriz de $N \times N$ componentes enteras. La computadora muestra su contenido en el orden resultante de recorrerla:
- con trayectoria de caracol - b) en zigzag - c) en zigzag diagonal. Ejemplo: Para la matriz
- | | | | |
|---|---|---|---|
| 4 | 3 | 2 | 7 |
| 8 | 1 | 6 | 0 |
| 1 | 5 | 9 | 2 |
| 9 | 4 | 7 | 8 |
- se mostrarán:
- 4 3 2 7 0 2 8 7 4 9 1 8 1 6 9 5
 - 4 3 2 7 0 6 1 8 1 5 9 2 8 7 4 9
 - 4 3 8 1 1 2 7 6 5 9 4 9 0 2 7 8
71. Se lee una matriz de $M \times N$ y dos enteros correspondientes a posiciones de filas, y se emite la matriz con dichas filas intercambiadas
72. 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 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: * *
73. Escribe un programa que lea dos matrices de 5×5 de reales y que visualice la suma de ambas.
74. Escribe un programa que almacene en una matriz los números impares del 1 al 50 y los imprima.
75. Construir un programa que realice la multiplicación de un vector por una matriz. Los elementos del vector y de la matriz se leerán por teclado.
76. Dada una matriz de orden $N \times M$, elaborar un programa que lea cada uno de los componentes de la matriz y escribe la matriz resultante de restar a cada elemento la media aritmética de su fila y la media aritmética de su columna. Los valores N y M son arbitrarios y se leerán por teclado al principio del programa.
77. Escribe un programa que capture e imprima la siguiente tabla:

Distrito	Candidato A	Candidato B	Candidato C
1	194	48	206
2	180	20	320
3	221	90	821

78. Escribe un programa que halle si una matriz introducida es un cuadrado mágico. Un cuadrado mágico es una matriz cuadrada con un número impar de filas y columnas (e incluso de diagonales) suman el mismo valor.
79. Escribe un programa que lea un vector de N elementos enteros. A continuación, debe rellenar una matriz de $N \times N$ elementos de tal forma que la primera fila contenga el vector leído; las siguientes deben contener el vector de la fila anterior pero con los elementos rotados a la izquierda una unidad. Por último, se debe visualizar dicha matriz. Por ejemplo, supongamos que $N = 5$ y $V = [1; 2; 3; 4; 5]$, el programa visualizará:

```
1 2 3 4 5
2 3 4 5 1
3 4 5 1 2
4 5 1 2 3
5 1 2 3 4
```

80. El dueño de una tienda ha permitido a sus 20 mejores clientes que le deban 10 artículos cada uno y que le paguen en el momento que a ellos les interese. Si almacena en una matriz de 20 por 10 los precios de los artículos, indica a) cómo debe calcular el dinero que le deben en total los 20 clientes; b) cómo debe calcular el promedio de endeudamiento por cliente; c) cómo puede saber qué cliente tiene una deuda mayor.
81. El dueño de una tienda anterior se ha dado cuenta de que no todos los clientes le deben 10 artículos, sino que pueden ser menos. Para poder contemplar esto, decide ampliar una columna más en la matriz para almacenar el número de artículos que cada cliente tiene pendientes de pago. Responde a las tres preguntas anteriores teniendo en cuenta esta nueva restricción.
82. No satisfecho aún con su aplicación, el tendero se queja de tener que apuntarse en un papel qué número de cliente corresponde con cada cliente para comprobar sus datos. Por este motivo decide hacer una nueva ampliación de su sistema y mantener un vector de registros que hagan esta correspondencia. Almacenará el DNI del cliente, su nombre y apellidos y el número de cliente que se le asigna en la matriz. Indica cómo se deben dar de alta a los 20 clientes y cómo se puede encontrar la deuda total de un cliente conociendo únicamente su DNI.
83. Disponemos de un conjunto de ciudades C , y una matriz D , donde cada $D[i][j]$ almacena la distancia de la carretera que conecta las ciudades $C[i]$, $C[j]$. Un valor $D[i][j] = 0$ indica que no existe camino directo entre $C[i]$ y $C[j]$. Diseña e implemente programas que permitan responder las siguientes preguntas:
- ¿Cuáles son las ciudades que se encuentran más cercanas?
 - ¿Ídem las más lejanas?
 - ¿Cuál es la longitud promedio de las carreteras?
 - ¿Cuál es la ciudad mejor conectada?
 - ¿Y la peor?
84. Escribe un programa que utilice la generación de números aleatorios para crear oraciones. El programa debe utilizar cuatro arreglos de apuntadores a char llamados artículo, sustantivo, verbo y preposición. El programa

debe crear una oración mediante la selección de una palabra al azar de cada arreglo en el siguiente orden: artículo, sustantivo, verbo, preposición, artículo y sustantivo. Al elegir cada palabra ésta se debe concatenar a las palabras previas en un arreglo lo suficientemente grande para almacenar una oración completa. Las palabras deben separarse con espacios. Cuando se imprime la oración final ésta debe comenzar con una letra mayúscula y terminar con un punto. El programa debe generar 20 oraciones. Los arreglos deben rellenarse de la siguiente manera:

- a. Artículo: el, la un, algún, cualquiera.
- b. Sustantivo: niño, niña, perro, pueblo, carro
- c. Verbo: condujo, brinco, corrio, camino, salto.
- d. Preposición: hacia, desde, sobre, bajo, en.

Cuando tu programa funcione, realiza las modificaciones necesarias para producir una historia con varias oraciones.

85. Mediante el uso de técnicas similares al programa anterior escribe un programa que genere rimas al azar.
86. Se utiliza una matriz para cargar la información referida a las ventas realizadas por un grupo de N (numerados de 1 a N) vendedores en cada una de las M zonas (numeradas de 1 a M). Para ello, se lee un número no determinado de veces una terna formada por número de vendedor, zona, venta. Cada vendedor puede realizar ventas en varias zonas y cada zona puede corresponder a varios vendedores. Se desea emitir el listado de zonas con su total por venta, ordenado en forma decreciente por venta.
87. Si se define `int tem [10][20][30]`; ¿Cuál de las siguientes expresiones devuelve la dirección de tem?

- ☐ Ninguna de las restantes respuestas es correcta
- ☐ `tem [0][0]`
- ☐ `tem [0]`
- ☐ `tem [0][0][0]`
- ☐ Hay más de una solución correcta

88. ¿Cuál de las siguientes expresiones localiza la información del tercer elemento de la cuarta fila de la tabla m?

- ☐ Ninguna de las restantes respuestas es correcta
- ☐ `m(3,4)`
- ☐ `m[3][4]`
- ☐ `m[3;4]`
- ☐ `m[3,4]`

89. Qué doble bucle for imprime los elementos de una matriz nxn que se encuentren por debajo de la diagonal principal en orden de columnas. Es decir dada: `int t[3][3]={1,2,3,4,5,6,7,8,9}` imprimiría 4 7 8.

- ☐ `for(i=0;i<n;i++) for(j=0;j<i;j++) printf ("%d ",t[i][j]);`
- ☐ `for(i=0;i<n;i++) for(j=0;j<i+1;j++) printf ("%d ",t[i][j]);`
- ☐ `for(i=0;i<n;i++) for(j=i;j<n;j++) printf ("%d ",t[i][j]);`
- ☐ `for(i=0;i<n;i++) for(j=i+1;j<n;j++) printf ("%d ",t[i][j]);`
- ☐ `for(i=0;i<n;i++) for(j=0;j<=i;j++) printf ("%d ",t[i][j]);`

90. ¿Cuál de las siguientes definiciones es correcta?

- ☐ `char tab[3][2]={"CA", "SE", "CO"};`
- ☐ Ninguna es correcta
- ☐ `char tab[3][2]={'C','A','S','E','C','O'};`
- ☐ Hay más de una correcta
- ☐ `char tab[3][2]={ {'C','A'},{'S','E'},{'C','O'}};`

91. Se utiliza una matriz para cargar la información referida a las ventas realizadas por un grupo de N (numerados de 1 a N) vendedores en cada una de las M zonas (numeradas de 1 a M). Para ello, se lee un número no determinado de veces una terna formada por número de vendedor, zona, venta. Cada vendedor puede realizar ventas en varias zonas y cada zona puede corresponder a varios vendedores. Se desea emitir el listado de zonas con su total por venta, ordenado en forma decreciente por venta.

92. Una agencia tiene 10 vendedores. Cada uno se identifica con un número entero entre 1 y 120. Cada vendedor vende en distintas zonas. Las zonas son 'a', 'b', 'c'. Cada uno realiza una o más ventas. Se quiere emitir el listado de las ventas máximas por vendedor, es decir el listado de los vendedores con el mayor monto que hayan alcanzado en una operación de venta. En esta lista no deben figurar quienes no hayan vendido nada. También se quiere los totales por zona. Para ello, se ingresa un número no determinado de veces una terna: número de vendedor, número de zona, monto de la operación. Los datos no están ordenados de manera alguna. El fin de datos se indica con número de vendedor -1 y no se lee para el ni zona ni venta. Escribir el programa que resuelva el problema.

93. Declara una matriz de número reales de tamaño DIMxDIM (DIM constante con valor 25). Escribir las siguientes funciones:

a. Función Traspuesta: Dada una matriz devolver su traspuesta.

`void traspuesta(float mat[], float matras[]);`

b. Función Simetrica: Dada una matriz indica si la matriz es simétrica (1) o si no lo es (0).

`int simetrica(float mat[]);`

c. Funcion Suma: Dadas dos matrices, calcular la suma de éstas.

`void sumaMat(float mat1[],float mat2[], res[]);`

d. Funcion Resta: Dadas dos matrices, calcular la resta de éstas. Piensa esta respuesta utilizando el procedimiento de suma de matrices, definido anteriormente.

e. Función Multiplica: Dadas dos matrices, devuelve el producto de ellas.

`void multiMat(float mat1[],float mat2[], res[]);`

94. La "Ferretería Francesa" ha dispuesto sus artículos en un depósito que posee 50 pasillos, donde en cada uno de ellos los estantes se numeran del 1 al 30. Por cada artículo, se tiene su código, su ubicación (pasillo-estante), su descripción (ej. martillo) y su stock (ej. 3 unidades). Se pide realizar un programa que

disponga de un menú con las siguientes cuatro opciones:

- ✓ a) Dado un artículo que ingresa al depósito, conociendo el código del mismo el programa deberá determinar si ya existe allí; de ser así , actualizará el stock, y en caso contrario, le asignará una nueva ubicación (pasillo-estante) además de almacenar los datos necesarios.
- ✓ b) Dado un número de pasillo y estante, mostrar en pantalla el código, descripción y stock del artículo que allí se encuentra; si el estante estuviese vacío, indicarlo con un mensaje.
- ✓ c) Indicar cuántos estantes vacíos hay en todo el depósito.
- ✓ d) Salir del programa.

95. ¿Qué estructura de datos usarías en el problema anterior?dados los siguientes supuestos:

- a) Al ampliarse la ferretería, las nuevas ubicaciones deben comprender no sólo el número de pasillo y estante, sino también el número de piso del depósito.
- b) Si se agregan 9 nuevos depósitos al “deposito central”.