



PRACTICA DEL MODULO 4

Ejercicios

[Breve descripción](#)

Ejercicios de programación para el módulo 4.

Versión 1.1

Martin Jerman

Martin.jerman@inspt.utn.edu.ar



Ejercicios con vectores

1. Escribe un programa que almacene, en un vector con dimensión 10, los números 10 primeros números de la Quiniela. El ingreso de los datos será indicando posición-numero, de modo que puedo ingresar la posición 5 antes que la posición 2.
2. Escribe un programa con un menú que defina las siguientes funciones y emita los resultados, la dimensión del vector será de 10 (elementos):
 - a) Cargar m elementos de un vector por teclado. Deberá considerar que no podrá superar el máximo de elementos de vector: `void cargar(int vect[], int unNum);`
 - b) 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[]);`
 - c) Dado un elemento y dado un vector de enteros, desarrolle una función que devuelva el número de apariciones del elemento en el vector. `int veces(int valor, int vect[], int unNum);`
 - d) 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 unNum);`
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. Se dispone de un conjunto de datos asociados a la velocidad de cierta reacción química bajo diferentes condiciones de temperatura y presión. Se desea establecer una representación gráfica (histograma) de esos datos de la forma siguiente:
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:
 - a) Insertar delante: Esta opción agrega un número en la primera posición del vector. Deberá desplazar, si fuese necesario, el resto de los elementos una posición a la derecha. Si el vector estuviese lleno, se perdería su último elemento.
 - b) 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.
 - c) 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.

- d) Finalizar el programa.
6. Desarrolle una función que reciba un vector de 10 números enteros y muestre los valores por pantalla, 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 se declara un vector de la siguiente manera: `int vec[6];` y luego se ejecuta: `vec[6] = 78;`?
 9. ¿Qué está mal en el siguiente fragmento de código?

```
int b[10];  
for (i=0; i<=10; i++)  
    b[i]=b[i+1];
```
 10. Construye un programa, utilizando funciones, que lea como máximo 10 números enteros, el ciclo termina al introducir un cero. 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.
 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 contenido del vector por pantalla.
 12. Dado un vector de dimensión N y un valor x float, desarrollar una función que almacene 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. Se debe utilizar funciones.
 13. 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;
```
 14. 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.
 15. Escribe una función que devuelva un puntero al máximo valor de un vector de `double`. Si el vector está vacío debe retornar `NULL`.
 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. 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;
18. Ídem anterior, pero los elementos de los dos vectores deben concatenarse intercalados. Ej:
V4= 2-7-56-80-7-2-8-4-30-13;
19. Se ingresan los N y M elementos de los arreglos unidimensionales A y B, respectivamente. Desarrollar 3 funciones que devuelvan la unión, diferencia e intersección de dichos vectores.
20. Escriba un una función que devuelva un vector de 6 posiciones que deben tener los valores de la lotería generados aleatoriamente entre 1 y 54 (utilice la función rand).
21. 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.
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. 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) $a + m[c/a] + b-- * m[1] / *q + 10 + a--;$
b) $a + (b * (c - d) + a) * *p++;$
c) $m[d] - d * e + (m[9] + b) / *p;$
d) $b++ * c-- + *q * m[2] / d;$
e) $(b/a) * (m[3] * ++e);$
f) $++*p+++*q;$
g) $++*p + ++*q;$
h) $m[c/a]-*p;$
i) $q[-3] + q[2];$

24. Desarrolle una función que pueda procesar cualquier vector de enteros, y emita el promedio de los valores que estén cargados.
25. Desarrollar una función que reciba un vector, su tamaño y un número; y devuelva la posición del número que se encuentra o -1 si no se encuentra.
26. Desarrolle una función recursiva que reciba un vector de enteros y devuelva la suma de los elementos.



Ejercicios con cadenas

27. Desarrolle una función que reciba una cadena por parámetro y devuelva su tamaño. No puede usar funciones de cadenas en el desarrollo.
28. Desarrolle una función que reciba una cadena por parámetro y devuelva la cantidad de vocales que contiene. No puede usar funciones de cadenas en el desarrollo.
29. Desarrolle una función que reciba una cadena por parámetro y devuelva la cantidad de consonantes. No puede usar funciones de cadenas en el desarrollo.
30. Desarrolle una función que reciba una cadena por parámetro y un carácter, y que reemplace todas las ocurrencias de ese carácter por X.
31. Desarrolle una función que reciba dos cadenas por parámetro y devuelva: 1 si la primera cadena es mayor, 0 si ambas son iguales o -1 si la segunda es mayor. No puede usar funciones de cadenas en el desarrollo.
32. Desarrolle una función que reciba dos cadenas por parámetro y retorne 1 si la segunda cadena es subcadena de la primera y cero en caso contrario.
33. Desarrolle una función que reciba dos cadenas y devuelva 1 si la cadena 2 está dentro de la cadena 1 (si es subcadena) y 0 (cero) si no lo es.
34. Desarrolle una función que reciba una cadena y la muestre invertida por pantalla.
35. Desarrolle una función que reciba una cadena y la devuelva sin espacios (compactarla). Al hacer las pruebas, no utilizar literal cadena.
36. Desarrolle una función que reciba una cadena y devuelva un 1 si es palíndromo (palabra capicúa) o cero si no lo es.
37. Desarrolle una función que reciba una cadena y devuelva la cantidad de mayúsculas que contiene.
38. Desarrolle una función que reciba una cadena y convierta todas sus minúsculas en mayúsculas.
39. Se declara: `char s[4];` y se ejecuta la siguiente instrucción: `scanf("%s", s);` se introduce por teclado: `PEPITO GRILLO`, e inmediatamente se ejecuta la instrucción `puts(s);` ¿Cuál de los siguientes literales se visualizará en la pantalla?
 - a) `PEPI`
 - b) `PEPITO GRILLO`
 - c) `PEPITO`
 - d) `PEPIT`
 - e) Ninguno, porque hay un error de sintaxis en alguna instrucción.
40. Si un programa contiene la sentencia: `strcpy(palabra, 'a');`
 - a) El programa no tiene ningún problema
 - b) El programa dará un error de compilación
 - c) El programa dará un error de ejecución
 - d) El programa dará un aviso (warning) de compilación
 - e) El programa no dará error alguno, pero no funcionará correctamente
41. Que devuelva la ejecución de la instrucción `strcmp("2345", "23345");`?



42. Construye un programa que permita ingresar una frase (hasta 256 caracteres) y resuelve:
 - a) Contar la cantidad de letras.
 - b) Contar la cantidad de palabras.
 - c) Determinar cuál es la vocal que aparece con mayor frecuencia.
 - d) Determinar cuántas veces aparece determinada letra, leída de teclado.
 - e) Averiguar qué cantidad de letras tiene la palabra más larga.
 - f) Leídas dos letras de teclado, se pide determinar la cantidad de veces que la primera letra precede a la segunda.
43. Escriba un programa que implemente una función que reciba una frase por teclado y la muestre ordenada por pantalla en forma inversa (si la frase es "Aguante Guns N Roses", la salida sería "Roses N Guns Aguante").

Ejercicios avanzados

44. Programar el juego TA-TE-TI. Para ello se requiere modelar un tablero de 3×3 en el que el jugador 1 ingresará una coordenada en la que guardará una 'x'. Luego el jugador 2 ingresara otra coordenada en la que se guardara una 'o'. Si la posición ya esta ocupada, el programa deberá pedir el reingreso de otra coordenada. Una vez completados todos los casilleros (el programa de darse cuenta cuando esta condición se dé), el programa debe decir qué jugador gano la partida o si hubo empate (ningún jugador ganó).
45. Mc Donald necesita registrar los ingresos de sus 5 sucursales de Mar Del Plata por semana. Para ello necesita un programa que pueda cargar, en forma aleatoria, los ingresos por día de cada sucursal. Una vez cargados los datos, se pide:
 - a) Calcular qué sucursal tuvo más ingresos en un día y cual fue ese día
 - b) Calcular qué sucursal tuvo más ingresos en la semana
 - c) Calcular el promedio de ventas por día en Mar Del Plata
 - d) Calcular el promedio de ventas de Mc Donald en toda la semana laboral
46. La zapatería de barrio Alto Calzado necesita un programa para administrar su local. Para ello el programa debe cumplir las siguientes funciones:
 - a) cargar los datos de sus 5 vendedores (DNI, nombre, apellido, genero, paresVendidos).
 - b) cargar los 10 tipos de calzado que venden (idCalzado, descripción, su stock remanente).
 - c) Al vender un par de zapatillas

Armar un programa con las estructuras necesarias para contabilizar las ventas hechas por los vendedores (ingresando DNI y idCalzado de la venta) y mantenga actualizado el stock remanente de pares de zapatillas.

47. Un mayorista de bebidas necesita un programa para administrar el stock de sus productos. Los productos que vende se categorizan en: bebidas alcohólicas, bebidas gaseosas y aguas saborizadas. Dado que cada categoría tiene especificaciones técnicas, es necesario tener alguna estructura que contenga dichos datos (como descripción del tipo de gaseosa, concentración de azúcar, concentración alcohólica y lista de bebidas). Lo que se pide es que el programa pueda:
 - a) Cargar categorías de bebidas
 - b) Cargar bebidas: descripción, marca, stock (por ejemplo, Pepsi, Pepsi CO, 15000 unidades).
 - c) Venta: por cada venta se registra la cantidad de unidades vendidas y las que queden
 - d) Mostrar por pantalla las ventas realizadas y stock remanente por categoría de bebida y bebida.



48. La batalla naval es un juego que se basa en un tablero de 10 por 10 casilleros en el que se ubican 5 barcos: un portaviones (5 casilleros), 2 destructores (3 casilleros) y dos patrulleros (2 casilleros). Se pide modelar un programa en el:
- El jugador A ingrese la posición de sus 5 barcos ingresando los casilleros que ocupa cada uno. Se asumirá que el jugador será honesto e ingresará las posiciones contiguas que ocupa cada barco.
 - Luego el jugador B ingresará coordenadas con la intención de hundir los barcos. Tras cada ingreso, el juego deberá indicar si fue "agua" (marcando con un carácter 'o' en el tablero) o "golpe" (marcado con una 'x' en el tablero). En cada movimiento deberá mostrarse el tablero con todos los disparos realizados, pero obviamente sin mostrar los barcos.
 - Al hundir todos los barcos, el juego deberá contabilizar la cantidad total de disparos y la cantidad de disparos fallidos. Con la intención de que luego al invertir los roles, el jugador que menos disparos hizo es el ganador.
49. El ejército de EE. UU. necesita desarrollar un programa para modelar el avance de sus tropas en el frente de batalla. Para ello se requiere que el programa modele un mapa de 10 x 10 km. En dicho mapa se deberán modelar la cantidad de tropas. Cada tropa puede tener una cantidad de soldados, otra cantidad de humvees y otra cantidad de tanques. Con este programa, el comandante de la ofensiva puede armar tropas y ubicarlas en el mapa para un ataque efectivo. Además, cada tropa debe poder identificarse como "aliado" o "enemigo", de manera que el comandante pueda evaluar la cantidad de fuerzas aliadas y enemigas con el mismo programa.
50. Al ejercicio anterior, agregarle una lista de unidades (soldados, humvees y tanques) disponibles, de manera que el comandante disponga de un límite y pueda evaluar:
- Qué cantidad de unidades esta utilizando en el ataque
 - Qué cantidad de unidades aun tiene disponible para atacar

Ejercicios con estructuras

51. Definir un tipo de dato llamado `Fecha` compuesto por 3 campos llamados `día`, `mes` y `año`. Además defina y desarrolle sus operaciones `getDia()`, `getMes()`, `getAnio()` y `isBisiesto()`.
52. Dada la estructura `struct producto {int id; float precio};` desarrollar una función que reciba dicha estructura como puntero y aplique un descuento del 20% al precio.
53. Dada la estructura del ejercicio 2, desarrolle una función que reciba dos estructuras `producto` y devuelva aquella que tiene el precio más caro.
54. Dada la estructura del ejercicio 2, desarrolle la función `int getMenor(struct producto p[], int dim);` que devuelva la posición del producto que tenga menor precio.
55. Desarrollar una función que reciba únicamente una estructura por puntero y le cargue los valores que corresponda. La estructura es libre, pero debe cargar mínimo 2 enteros y un flotante.
56. Desarrolle una función que reciba 3 parámetros (de tipo a elección) y devuelva una estructura con esos valores cargados en sus campos.
57. Desarrollar un programa que lea registros de alumnos y procese sus notas. Se ingresarán 10 alumnos de los cuales se ingresa su legajo y 3 notas. Se pide calcular el promedio de las 3 notas y guardar el promedio de cada alumno para listarlo al final en la forma "legajo:promedio" y además mostrar el mayor promedio al final de ese listado resaltado.



Defina una estructura alumno que tenga un campo `int legajo` y otro `enum aprobado` que deberá considerar los valores "DESAPROBADO", "APROBADO PARCIAL" y "APROBADO". Desarrolle un programa que pida 5 legajos por teclado y cargue en la estructura el estado de aprobación.