

Nombre:

Práctica Test 1: Diciembre 2002

1. Programa: Vectores y Matrices

Realizar un programa (`c:\temp\examen\ex1.c`) que declare un vector con capacidad para almacenar N valores tipo `int`. El programa pedirá al usuario valores comprendidos entre -100 y 100 y los irá guardando en el vector siempre que sean distintos de cero. Cuando el usuario introduzca el valor cero, termina la lectura de datos y empiezan los cálculos:

- Construir una matriz cuadrada de números reales, cuya primera columna sean todos los elementos del vector leído, la segunda columna los mismos elementos divididos por 2, la tercera columna divididos por tres, etc. (ver ejemplo)
- Calcular la suma de todos los elementos de la matriz.
- Mostrar por pantalla la matriz y el resultado de la suma, de manera ordenada.

EJEMPLO:

```
Dame un valor entre -100 y 100, (0 para terminar) -->12
Dame un valor entre -100 y 100, (0 para terminar) -->3
Dame un valor entre -100 y 100, (0 para terminar) -->-5
Dame un valor entre -100 y 100, (0 para terminar) -->0
Fin de la lectura
La matriz es:
 12.0  6.0  4.0
  3.0  1.5  1.0
 -5.0 -2.5 -1.7
```

La suma de los elementos es: 18.333333

2. Programa: Cadenas de Caracteres

Realizar un programa (`c:\temp\examen\ex2.c`) que lea una frase desde teclado y la guarde en un vector de caracteres `char a[100]`.

A continuación construirá otro vector `b`, basado en `a`, donde los dobles espacios queden convertidos a un único espacio. Finalmente se mostrará por pantalla la frase inicial y la frase final entre los caracteres '`<`' y '`>`'. Ver ejemplo:

```
Escribe una frase
hola adios grrrrrr
Frase original: < hola adios grrrrrr>
Frase final: < hola adios grrrrrr>
```

Notas

- Lo más importante es que los programas funcionen bien. Sin embargo se evaluará el código que se escriba a mano en estas hojas.
- En el primer problema se valorará especialmente el control del número de elementos del vector, para evitar accesos fuera de la memoria (buffer overflow).
- En el segundo problema se valorará especialmente que las cadenas de caracteres estén bien terminadas con '`\0`', y que no se acceda a elementos no inicializados o fuera de la memoria del vector.
- En ambos problemas se valorará el aspecto de la salida de resultados y si se adapta al formato que se pide en el enunciado
- En ambos problemas se valorarán los comentarios de descripción de las variables, los comentarios de las distintas secciones del código y el correcto sangrado del código.