

Ejercicios Fundamentos de la programación

□1. Realizar un programa que imprima la tabla de multiplicar por un número leído como dato.

□2. Realizar un programa para calcular el máximo común divisor de dos números enteros.

□3. Realizar un programa que escriba un rombo simétrico con asteriscos tomando como dato el número de asteriscos que tiene de lado.

□4. Realizar un programa que calcule el número e, con el dato introducido como error, mediante el desarrollo de la serie:

$$e = 1 + \sum_{i=1}^n \frac{1}{i!}$$

□5. Realizar un programa que lea la longitud de los tres lados de un triángulo y analice qué tipo de triángulo es entre:

- No es triángulo
- Equilátero
- Isósceles
- Escaleno
- Rectángulo

□6. Realizar un programa que analice un texto terminado con un punto (.) y contabilice los siguientes aspectos:

- Número total de caracteres
- Número total de vocales
- Número total de “A” o “a”
- Número total de “E” o “e”
- Número total de “I” o “í”
- Número total de “O” o “o”
- Número total de “U” o “u”

□7. Realizar un programa que dado un número N, introducido como dato, escriba todos los números entre 1 y 10000 que cumplan las dos siguientes reglas:

- La suma de sus cifras debe ser un divisor de N
- El producto de sus cifras debe ser un múltiplo de N

□8. Realizar un programa que a partir del capital (C), el tanto por ciento de interés anual (I) y los años de amortización (A) de un crédito, introducidos como

datos, calcule la anualidad fija a pagar a lo largo de los A años siguiendo la fórmula:

$$Anualidad = C \frac{\left(1 + \frac{I}{100}\right)^A \frac{I}{100}}{\left(1 + \frac{I}{100}\right)^A - 1}$$

El programa también debe calcular para todos los años la parte de la anualidad dedicada al pago de intereses y de la amortización.

- 9. Realizar una función que devuelva un valor booleano cierto o falso si el número que se le pasa como argumento es "perfecto". Para que un número sea "perfecto" es necesario que su valor sea igual a la suma de todos sus divisores incluyendo al 1 y sin incluirle a él mismo.
- 10. Realizar un procedimiento que lea una letra, compruebe que es una de las utilizadas para escribir los números romanos y devuelva su valor, según la siguiente tabla:

I	-	1	C	-	100
V	-	5	D	-	500
X	-	10	M	-	1000
L	-	50	Resto	-	0

Utilizando el procedimiento anterior, realizar otro procedimiento que lea un número romano y devuelva su valor entero correspondiente. Este procedimiento tendrá en cuenta las reglas de escritura de los números romanos. Finalmente, utilizando este último procedimiento realizar un programa que lea dos números romanos e indique cuál de ellos es mayor.

- 11. Realizar un programa que escriba todas las permutaciones que se pueden obtener de las letras A, B, C y D.
- 12. Realizar una función que devuelva el día de la semana cuando se le pasan como argumentos el día, el mes y el año de una fecha cualquiera. Utilizando la función anterior escribir un programa al que se le introducen como datos el mes y el año y devuelve como resultado la hoja del calendario de dicho mes.

□13. Realizar un programa que analice un texto terminado con un punto (.) y elabore las siguientes estadísticas:

- Número total de palabras del texto
- Número de palabras que utilizan N o más vocales diferentes
- Número de palabras que utilizan M o más consonantes diferentes

Los valores N y M se leerán como datos del programa.

□14. Realizar un programa para controlar las plazas de un aparcamiento. El aparcamiento dispone de 25 plazas de dos tamaños diferentes: 15 pequeñas y 10 grandes con la disposición que se muestra a continuación:

- Plazas Grandes (G): del 1 al 10.
- Plazas Pequeñas (P): del 1 al 15.

La asignación se realizará automáticamente según el tamaño del vehículo que se quiere aparcar con el siguiente algoritmo:

- Cada vehículo solamente ocupará una plaza.
- Un vehículo pequeño siempre ocupará una plaza pequeña, salvo que estén todas ocupadas y exista alguna grande libre.
- Un vehículo grande sólo puede aparcar en una plaza grande. Si todas están ocupadas no podrá aparcar, aunque estén todas las pequeñas libres.
- De todas las plazas libres, siempre se ocupará primero la de número menor.

El programa tendrá 3 opciones básicas:

- Entrada: es necesario indicar el tamaño de coche (P/G).
- Salida: es necesario indicar la plaza que se deja libre. Por ejemplo: P 5.
- Situación del aparcamiento: indicando las plazas libres y las ocupadas.

□15. Realizar un programa que simule un cajero automático de monedas. Los tipos de monedas que dispone el cajero son de 10, 20 y 50 céntimos de euro y 1 y 2 euros. Inicialmente el cajero tiene 100 monedas de cada tipo, que se van consumiendo para proporcionar las cantidades solicitadas. El cajero debe obtener la cantidad solicitada con los tipos de moneda que tenga en cada momento, tratando siempre de utilizar las monedas de mayor valor.

□16. Realizar una función que, a partir de dos puntos del plano pasados como argumentos, devuelva cierto cuando el primero esté más alejado del origen de coordenadas que el segundo y falso en caso contrario. Utilizando la función anterior realizar un programa que ordene en un vector hasta 10 puntos según su distancia al origen.

□17. Realizar un programa que analice un texto terminado con un punto (.) y extraiga del mismo las siguientes palabras:

- Palabra más larga.
- Palabra más corta.
- Palabra con más vocales.
- Palabra con más consonantes.

□18. Realizar un tipo abstracto de datos (TAD) para manejar datos enteros en forma de lista con los valores ordenados de menor a mayor y que disponga de las siguientes operaciones básicas:

- Iniciar la lista vacía.
- Comprobar si la lista está vacía.
- Retirar el primer número de la lista.
- Insertar un número en la lista ordenada.
- Conocer el número de elementos de la lista.

□19. Realizar un tipo abstracto de datos para el juego de las TRES EN RAYA. El tablero de 3x3 y cualquier cuadricula podrá estar vacía u ocupada sólo por una ficha blanca o negra. Las operaciones son:

- Poner una ficha en una cuadricula comprobando que no está ocupada.
- Quitar una ficha de una cuadricula.
- Comprobar si se produce tres en raya e indicar si es de blancas o negras.

□20. Realizar un tipo abstracto de datos (TAD) capaz de contener los siguientes datos de 50 personas: Nombre, Apellidos, DNI y Edad. El TAD dispondrá solamente de los siguientes procedimientos:

- ImprimirMayoresDe. Que recibe como entrada un numero natural, e imprime por pantalla los datos de aquellas personas cuya edad sea mayor o igual que la solicitada.
- BuscarApellido. Que recibe como entrada una cadena de caracteres, e imprime por pantalla los datos de aquellas personas cuyo primer apellido es igual a la cadena solicitada.

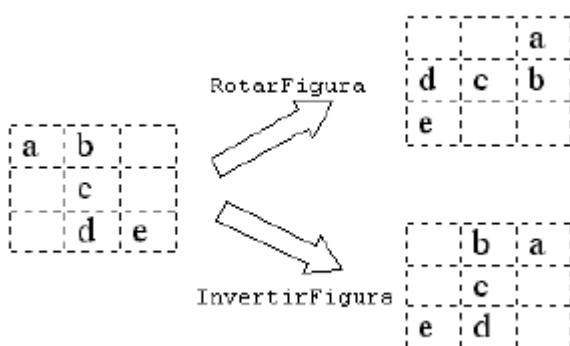
□21. Las fichas de dominó disponen de dos partes, cuyos valores posibles están comprendidos entre 0 y 6. Realizar un tipo abstracto de datos que permita almacenar las 28 fichas de dominó. Este tipo abstracto incluirá las operaciones CrearFichas, que almacena en una variable las 28 fichas y Repartir4 la cual devolverá 4 fichas elegidas de forma aleatoria de entre las 28.

□22. Realizar un tipo abstracto e datos (TAD) de una ecuación de 2º GRADO de la forma ($ax^2 + bx + c = 0$) con los procedimientos de entrada_coeficientes y cálculo de raíces.

□23. Realizar un tipo abstracto de datos llamado **matriz** con las operaciones leer, escribir en pantalla y obtener la matriz traspuesta de una matriz cuyas dimensiones puedan ser hasta 10x10. Nota: el elemento ij de la matriz traspuesta es el elemento ji de la matriz original.

□24. Escribir el tipo abstracto de dato abstracto ALUMNO (nombre, apellido, DNI y 15 notas) con las operaciones de dar de alta, introducir las notas y obtener la media.

□25. Construir un TAD (Tipo Abstracto de Datos) capaz de contener una figura como la del ejemplo en una cuadrícula de dimensión 3x3. El TAD dispondrá únicamente de los procedimientos RotarFigura e InvertirFigura, que transformarán la figura que reciban de entrada según se describe en el siguiente ejemplo:



□26. Imprimir en pantalla los datos del alumno incluidos en la cabecera. El código fuente de todas las prácticas debe incluir una cabecera con el siguiente formato:

- Debe ser un comentario delimitado por /* y */.
- Debe incluir 5 campos obligatorios: NOMBRE, PRIMER APELLIDO, SEGUNDO APELLIDO, DNI y EMAIL.
- El valor de cada campo se delimita por el símbolo #.
- El resultado de la ejecución debe imprimir cada dato en una línea distinta.

□27. Realizar un programa que imprima por pantalla rombos concéntricos dibujados con los caracteres '@', 'o' y ''.

El programa solicitará, como dato de entrada, la longitud del lado del rombo más externo (es decir, su número de caracteres). De fuera hacia dentro, el primer rombo estará formado por caracteres '@', el segundo por caracteres '', el tercero por caracteres 'o', el cuarto nuevamente por caracteres '', el quinto por caracteres '@', el sexto también por caracteres '', el séptimo por caracteres 'o' y así sucesivamente.