

- (3.5 puntos)** Se desea implementar una clase para representar figuras geométricas de tipo cuadrado. Un cuadrado viene definido por la coordenada de la esquina inferior izquierda (x, y) y la longitud del lado.
 - Defina los atributos de la clase y el constructor o constructores que estime oportunos.
 - Implemente un método para el cálculo del área.
 - Implemente un método para el cálculo del perímetro.
 - Implemente un método que determine si un cuadrado tiene mayor área que otro.
 - Implemente un método que determine si un cuadrado contiene a otro. Un cuadrado C_1 determinado por (x_1, y_1) y l_1 contiene a otro cuadrado C_2 dado por (x_2, y_2) y l_2 si se cumple que $x_2 \geq x_1$ y $x_2 + l_2 \leq x_1 + l_1$ y $y_2 \geq y_1$ y $y_2 + l_2 \leq y_1 + l_1$.
 - Implemente un módulo para que dado un cuadrado y un vector de cuadrados, devuelva cuántos de éstos están incluidos en el primero.
- (2.5 puntos)** Se está diseñando un sistema web que recolecta datos personales de un usuario y, en un momento dado, debe sugerirle un nombre de usuario (login). Dicho login estará basado en el nombre y los apellidos, en concreto estará formado por los N primeros caracteres de cada nombre y apellido (en minúsculas, unidos y sin espacios en blanco). Por ejemplo, si el nombre es "Antonio Francisco Molina Ortega" y $N=2$, el nombre de usuario sugerido será "anfrmoor".
 Debe tener en cuenta que el número de palabras que forman el nombre y los apellidos puede ser cualquiera. Además, si N es mayor que alguna de las palabras que aparecen en el nombre, se incluirá la palabra completa. Por ejemplo, si el nombre es "Ana CAMPOS de la Blanca" y $N=4$, entonces la sugerencia será "anacampdelablan" (observe que se pueden utilizar varios espacios en blanco para separar palabras).
 Implemente un módulo que reciba una cadena de caracteres formada por nombre y apellidos (separados por uno o más espacios en blanco) y un valor entero N y que devuelva otra cadena con la sugerencia de login.
- (2.5 puntos)** Para gestionar un campeonato de n equipos se utiliza una matriz de tamaño $n \times n$. En cada posición de esta matriz se pueden almacenar tres posibles valores ('1', 'X', '2'). La fila f y columna c contendrá un valor correspondiente al partido que enfrenta al equipo f con el c , de forma que si vale '1' indica que ha ganado f , si vale 'X' han empatado y si vale '2' ha ganado c .

	0	1	2	3	4
0		1	1	1	2
1	2		X	X	1
2	1	X		1	2
3	2	X	1		X
4	1	2	X	2	

Liga de 5 equipos:
 1: Gana fila
 2: Gana columna
 X: Empate

→ Partido 2 contra 4: Gana 4
 → Partido 3 contra 4: Empate
 → Partido 4 contra 2: Empate
 → Diagonal principal no se usa

Observe que dados dos equipos (m, n) habrá dos partidos: uno en el que se enfrentan m y n y el recíproco, de n con m . Además, el valor de la diagonal no se usa, ya que no existe el partido n contra n .

Implemente un módulo para obtener un vector con los resultados finales de la liga. En este vector se contabilizan, para cada equipo, los puntos obtenidos teniendo en cuenta que una victoria implica 3 puntos, un empate 1 punto, y una derrota 0 puntos.

- (1.5 puntos)** Cree una función **RECURSIVA** que para dos valores enteros positivos n y k calcule cuántos divisores **propios positivos** tiene n que sean estrictamente menores que k . Un número d es divisor propio de m si d divide a m y d es distinto de m (el 1 es divisor propio de cualquier número distinto de 1). La cabecera de la función ha de ser la siguiente:

```
int divisoresMenores(int n, int k)
```

Por ejemplo, `divisoresMenores(10, 6)` debe devolver 3 ya que 1, 2 y 5 son los dos divisores propios positivos de 10 estrictamente menores que 6. Observe que para cualquier número p mayor que 1 el resultado devuelto por `divisoresMenores(p, p)` debe ser 1 si p es primo.



DECSAI

Departamento de Ciencias
de la Computación e I.A.

Fundamentos de Programación. Doble Grado

Convocatoria de Febrero. Curso 2012/2013

11 de Febrero de 2013

5. (0.5 puntos. Pregunta extra para Matrícula de Honor) Sobre la clase `MiVectorCaracteres`, añadid un método que determine si dicha secuencia de caracteres $C1$ contiene a otra secuencia $C2$ en el mismo orden (no tienen que estar consecutivos) y de forma *cíclica*. Para que se cumpla este criterio se deben satisfacer que todos los caracteres de $C2$ deben estar en $C1$, deben estar en el mismo orden aunque no de forma consecutiva y finalmente, si durante la búsqueda se ha llegado al final de la secuencia $C1$, se debe proseguir la búsqueda por el inicio de $C1$, pero sin sobrepasar la posición en la que hubo la primera concordancia. Por ejemplo, las siguientes cadenas contienen a $C2 = abcd$:

`xzayobnmcpwqdfg` `ftkcpqxqdhjzaxqoblki` `tzsbluyclpygdmngrafvc`

Hay que destacar que la primera letra de $C2$ a buscar en $C1$ podría estar en cualquier sitio. Por ejemplo, si $C1 = \underline{b}ghc\underline{j}a\underline{d}x\underline{a}k$ y $C2 = abcd$, podemos ver que a partir de la primera a de $C1$ no podemos encontrar $C2$ de forma cíclica, aunque sí lo podemos hacer a partir de la segunda a de $C1$. También puede darse el caso de que $C1$ no contenga a $C2$ de forma cíclica aunque incluya todas sus letras, como por ejemplo, con $C1 = \text{tzsbluyclpcaygdmxngrfvc}$ y $C2 = abcd$