



EJERCICIOS 1

- 1) Diseñar un algoritmo que dé como salida todos los números enteros menores que 1000 tales que cumplen las dos condiciones siguientes:
- que no terminen en cero,
 - que tengan la propiedad de que si borramos su último dígito (el más a la derecha), el número resultante es divisor del original.

Por ejemplo, el 39 sería uno de tales números ya que no termina en cero y si borramos el dígito más a la derecha (9), el entero resultante (3), es divisor de 39.

- 2) Diseñar un algoritmo que lea de un dispositivo de entrada un número entero $c \geq 0$, sume tantos números impares como sea necesario para que su suma S , sea menor o igual que c , pero sumando el siguiente impar la suma resultante exceda c . El programa debe dar como salida 3 datos:
- El número de impares sumados,
 - El primer impar no sumado,
 - La suma S .

Por ejemplo, si el dato de entrada c vale 12, tenemos que $1+3+5 = 9$ y $1+3+5+7 = 16$, por tanto, el número de impares sumados es 3, el primer impar no sumado es 7 y la suma S es 9.

- 3) Realizar el diseño de un algoritmo que calcule el siguiente sumatorio

$$y = (1+1!) + (2+2!) + \dots + (N+N!)$$

siendo N un número entero introducido por teclado y denotándose por $x!$, el factorial de x .

- 4) Diseñar un algoritmo que cambie cualquier suma de dinero hasta 99 € usando las monedas de 1 €, 5€, 10€ y 20€.

5) Diseñar un algoritmo que construya el cuadrado latino de orden n , que ilustra el gráfico adjunto:

1	2	3	4	5	$n=5$
5	1	2	3	4	
4	5	1	2	3	
3	4	5	1	2	
2	3	4	5	1	