	Examen 3-4_A: Bucles y Funciones		
	1º DAW - PRO		
	Departamento de Informática	16/11/2023	Página 1 de 3

## Ejercicio 1

Haz un programa que solicite un número  $n$  y dibuje un triángulo de base y altura  $n$  centrado de la forma de la figura. Un ejemplo de 5 filas ( $n=5$ ) sería:

```

      *
     * *
    * * *
   * * * *
  * * * * *
```

Para entenderlo mejor verlo en forma de tabla

				*				
			*		*			
		*		*		*		
	*		*		*		*	
*		*		*		*		*


## Ejercicio 2

Escribir un programa que lea un número entero por teclado y llame a una función que tenga como argumento dicho número y que compruebe si el número es impar. La función debe devolver un booleano que usará el programa principal para mostrar por pantalla si el número es impar o no. Ejemplo:

- Para 36 el programa mostraría  
¿36 es impar? False
- Para 36 y 3 el programa mostraría:  
¿37 es par? True

Aprovechando esta función `impar()`, utilizarla para saber si el número tiene algún divisor impar de la siguiente manera:

- Mediante un bucle que empiece en 2 y acabe antes de  $n$  y con incremento 1.
- El bucle debe estar optimizado para que realice el menor número de estimaciones posible con las condiciones iniciales propuestas (llamar a la función `impar()` no se considera una estimación).
- Dentro del bucle el programa principal deberá ir mostrando los divisores impares del número.
- En caso de que uno de los divisores sea número 13, el programa mostrará el 13 como último divisor y no mostrará más divisores impares aunque los hubiera.

	<b>Examen 3-4_A: Bucles y Funciones</b>		
	<b>1º DAW - PRO</b>		
	<b>Departamento de Informática</b>	<b>16/11/2023</b>	<b>Página 2 de 3</b>


## Ejercicio 3

Realiza un programa que calcule el producto de los  $n$  primeros números pares llamando a una función:

- A la que se le pase un parámetro entero y devuelva el valor del resultado en una variable tipo **int**. El cálculo se realizará con un método iterativo. El nombre de la función será: **productoParIterativo()**.
- A la que se le pase un parámetro entero y devuelva el valor del resultado en una variable tipo **long**. El cálculo se realizará con método recursivo. El nombre de la función será: **productoParRecursivo()**.
- Con una función sobrecargada de **productoParIterativo ()**. Esta función recibirá dos parámetros: el primero el mismo que en los casos anteriores y el segundo parámetro será un número entero. En este caso se imprimirá calculado desde la misma función:
  - Si el segundo parámetro es mayor o igual que cero, se imprimirá el producto de los  $n$  primeros números pares positivos.
  - Si el segundo parámetro es menor que cero, se imprimirá el producto de los  $n$  primeros números pares negativos.
  - Si el segundo parámetro es cero, se imprimirá: “El cero es un número par”.
- Con una función sobrecargada de **productoParRecursivo()** que sea tipo **void** y que imprima el resultado del producto desde la misma función, haciendo uso de la sobrecarga recursiva de una función que vimos en clase.

Realiza un menú que dé a elegir entre los distintos métodos: iterativo, recursivo, iterativo sobrecargado, iterativo sobrecargado recursivo.

**Pista:** un número par lo puedes poner como  $2*n$ .

	<b>Examen 3-4_A: Bucles y Funciones</b>		
	<b>1º DAW - PRO</b>		
	<b>Departamento de Informática</b>	<b>16/11/2023</b>	<b>Página 3 de 3</b>

## Consideraciones Generales

Escribir el código de los tres ejercicios planteado, usando ayudándote del IDE Netbeans 19 junto con la versión 17 de JDK, como hemos visto en clase. Usar los elementos y estructuras vistos hasta ahora en clase.

A la hora de evaluar se aplicará la rúbrica adjunta, donde además de comprobar el correcto funcionamiento de los programas, se evaluarán los siguientes puntos:

- Código eficiente: uso apropiado de los recursos. No hay código innecesario o realiza acciones superfluas.
- Realizar el menor número de operaciones posibles.
- Crear una interfaz de usuario sencilla e intuitiva.
- Código ordenado: con sangrías en los bloques, espaciados apropiados, etc.
- Nombre de variables que tienen significado.
- Establece comentarios en el código.

## Rúbrica

PRO Examen 4.1: Bucles-Funciones

Acciones Cancelar Guardar

		Excelente	10 Avanzado	7.5 Medio	5 Básico	2.5 Nulo	0
<b>1. Triángulo de *</b>	<b>20 %</b>						
Imprimir árbol de asteriscos centrado 1c 1d 1a 1b 1g 1i		El programa funciona y tal y como se pide en el enunciado y el código está impecable.	Medio + el código está mejor en eficiencia y apariencia que en la casilla "medio".	El programa funciona correctamente pero tiene fallos de indentación, sin comentarios, variables innecesarias...	El programa funciona pero el resultado no es correcto aunque se aproxima al resultado solicitado.	El programa no compila, o hace algo totalmente diferente a lo que se pide en el enunciado.	
<b>2. Booleano</b>	<b>30 %</b>						
Programas que deveuelven 1c 1d 2d 2e 2f 3b 3c 3e 3f 3g		El programa funciona y tal y como se pide en el enunciado.	El programa funciona correctamente aunque no contempla algún punto planteado en el enunciado.	El programa funciona el bucle pero no realiza la función booleana como se pedía.	El programa funciona pero sólo la primera parte de función booleana.	El programa no compila, o hace algo totalmente diferente a lo que se pide en el enunciado.	
<b>2b. Código-eficiencia</b>	<b>5 %</b>						
Código ordenado, eficiente, 3e 3f 3g		Cumple todos los criterios sobre código y eficiencia.	Falla en un criterio sobre código y eficiencia.	Falla en dos criterios sobre código y eficiencia.	Falla en tres criterios sobre código y eficiencia.	El programa falla en más de 3 criterios de código y eficiencia.	
<b>3a. Producto im/par</b>	<b>40 %</b>						
Cálculo del productor de los n 4c 2d 2e 2f 1c 1d 1e 1f 3b 3c 3e		Funcionan correctamente todos los apartados a), b), c) y d).	Funcionan correctamente tres de los apartados a), b), c) ó d).	Funcionan correctamente dos de los apartados a), b), c) ó d).	Funciona correctamente uno de los apartados a), b), c) ó d).	El programa no compila, o hace algo totalmente diferente a lo que se pide en el enunciado.	
<b>3b. Código-eficiencia</b>	<b>5 %</b>						
Código ordenado, eficiente, 3e 3f 3g		Cumple todos los criterios sobre código y eficiencia + menú.	Falla en un criterio sobre código y eficiencia o en el menú	Falla en dos criterios sobre código y eficiencia o 1 criterio y el menú.	Falla en tres criterios sobre código y eficiencia o 2 criterios y el menú.	El programa falla en más de 3 criterios de código y eficiencia y en el menú.	

Total: 100 %