	RAMA:	Informática	CICLO:	DAM		
	MÓDULO	Programación				CURSO: 1º
	PROTOCOLO:	Boletín clase	AVAL:		DATA:	
	AUTOR	Francisco Bellas Aláez (Curro)				

Boletín de prácticas de Programación 2: Iniciación y estructuras de control

Los nombres de los ejercicios tendrán la forma *BolX_EjerY* donde *X* e *Y* son los números del boletín y del ejercicio dentro del boletín. Por ejemplo el ejercicio 3 de este boletín tendrá el nombre *Bol2_Ejer3*.

Usa nombres claros para los identificadores que utilices en tus programas. Comenta el código si deseas aclarar algún punto.

- Realiza un programa que muestre los números del 1 al 20. Debes hacerlo con los 3 bucles (while, do-while y for) y un contador en cada bucle..
 - A continuación en el mismo programa y con el bucle que prefieras, haz que muestre los números pares entre 1 y 50. Estos deben quedar bien formateados en una columna de tamaño 5 caracteres justificados a la derecha mediante printf.
- Realizar un programa que pida dos números enteros al usuario y muestre el mayor de los dos o un mensaje de igualdad en caso de que sean iguales. También debe indicar si son múltiplos o no.
- Haz un programa que pida un nº al usuario. Si es positivo realiza mediante un bucle, un contador y un acumulador la suma de números desde el 1 hasta el que metió el usuario. Finalmente muestra dicha suma final. Si el número es negativo o 0 no hace nada.
- Haz un programa que pida un nº entero al usuario y luego un número real.
El entero debe mostrarlo en decimal; en octal ocupando 5 caracteres y rellenando los vacíos a la izquierda con ceros; y en hexadecimal poniendo los dígitos “letra” en mayúscula.
El real lo mostrará con 3 decimales y ocupando al menos 7 caracteres.
- Programa un conversor de temperaturas. Al usuario se le pide los grados Celsius y se le da dos opciones: a si los quiere convertir a Fahrenheit (1) o a Kelvin (2). Las opciones serán 1 y 2 para pedir tipo de conversión y 3 salir. Usa como plantilla el código que hay en el apartado de la estructura *do-while*. El programa se repetirá hasta que se elija 3. Finalmente muestra el resultado de la conversión con 2 decimales. Usa *doubles*. Las funciones de conversión son:

$$K=C+273$$

$$F=1.8*C+32$$
- Realizar un menú con las opciones (los números se piden en cada opción, es decir, en cada case):
 - Cuadrado de un número real. Se mostrará con 2 decimales de aproximación
 - Inverso de un número (1/x). Se mostrará con 4 decimales de aproximación. No se admite el 0 (vuelve a pedirlo hasta que se meta distinto de 0).
 - Raíz cuadrada de un número. Debe mostrar el resultado con 3 decimales de aproximación. No admite números negativos (debe pedir número hasta que se meta uno positivo o 0). Averigua en internet cómo se haya la raíz cuadrada en Java.
 - Operación AND a nivel de bit entre dos enteros. Resultado en hexadecimal.
 - Operación OR a nivel de bit entre dos enteros. Resultado en hexadecimal.
 - Salir. El programa no debe acabar mientras no se pulse esta opción.

COLEXIO VIVAS S.L.	RAMA:	Informática	CICLO:	DAM		
	MÓDULO	Programación				CURSO: 1º
	PROTOCOLO:	Boletín clase	AVAL:		DATA:	
	AUTOR	Francisco Bellas Aláez (Curro)				

7. a) Realizar un programa que acepte 5 números del usuario y calcule y muestre su media (la media es la suma de los números dividido entre 5). Al final debe permitir ejecutar de nuevo el programa o salir del mismo. *Nota: Si lo ves claro, puedes hacer directamente el apartado (b) y saltarte el (a)*
 b) Mejorar el apartado anterior (o realiza otro programa) de forma que admita una cantidad indefinida de números. Es decir, el usuario introduce números y acaba cuando pulse el 0. En ese momento calcula la media (el 0 no se incluirá en el cálculo de la media).

8. Realizar un programa que pida tres números al usuario en tres variables distintas (no uses bucles) y muestre el mayor de los tres (si son iguales dos de ellos, o los tres muestra cualquiera de los máximos). Trata de hacer lo la menor cantidad posible de comparaciones.

9. Realizar un programa que muestre el máximo de una serie de números enteros leídos por teclado. Primero se debe pedir la cantidad de números a leer, tras lo cual se pedirá un número tras otro y cuando se llegue a la cantidad indicada se mostrará cuál es el máximo de los introducidos.
 Ejemplo de salida:
 Cuántos números deseas introducir? 3
 Introduce número: 5
 Introduce número: 9
 Introduce número: -2
 El número máximo introducido es el 9

10. Realizar un programa que pida dos números al usuario en sendas variables y luego intercambie el valor de dichas variables. Finalmente mostrará en pantalla ambas variables.
 Nota: A este algoritmo se le denomina swapping o intercambio de variables y tiene diversas aplicaciones.

11. Escribir un programa que encuentre el primer valor de N para el cuál se cumple que la suma $1+2+3+\dots+N > 10000$. *(Nota: la solución es 141)*

12. Realizar un programa que saque los siguientes datos por pantalla (usa bucles for y haz en un único programa todos los apartados).
 a) Los números del 1 al 50.
 b) En orden inversa del 50 al 1
 c) Los números pares menores que 20 por orden creciente (2, 4, 6, 8, ... 16, 18)
 d) Los números impares entre lo 90 y el 130 por orden creciente
 e) Los múltiplos de 5 entre el 70 y el 25 por orden decreciente (70, 65, ... 30,25).

13. Hacer con bucles for (en un único programa todos los apartados):
 a) Programa que dé el resultado de la suma de números pares entre 0 y N (N lo decide el usuario).
 b) Programa que dé el resultado de la suma de números impares entre 0 y N (N lo decide el usuario).
 c) Programa que dé el resultado de la suma de números pares por una parte e impares por otro entre 0 y N (N lo decide el usuario). **Debe realizarse con un único bucle.**

<div>COLEXIO</div> <div>VIVAS</div> <div>S.L.</div>	RAMA:	Informática	CICLO:	DAM				
	MÓDULO	Programación					CURSO:	1º
	PROTOCOLO:	Boletín clase	AVAL:		DATA:			
	AUTOR		Francisco Bellas Aláez (Curro)					

14. Contar la cantidad de números negativos introducidos por el usuario. Se termina cuando el usuario introduce el cero. Se debe informar también del total de números introducidos (no se cuenta el cero).

Ejemplo de salida:

Introduce número: -10

Introduce número: -2

Introduce número: 3

Introduce número: 0

Se introdujeron 2 números negativos de un total de 3.

15. Realizar un conversor de euros a libras y de libras a euros. No debe admitir valores negativos: si el usuario mete negativo, informa del error y lo vuelve a pedir hasta que meta positivo. Debe ejecutarse de forma continua hasta que el usuario lo desee (plantea un menú con dos opciones de conversión y salir). Debe presentar los datos con 2 decimales.

16. Realizar el juego *Adivina un número entre 1 y 100* para dos personas. El Computador pide un número entre 1 y 100 al jugador 1, luego borra la pantalla y el jugador 2 debe adivinar el número en, como máximo, 5 intentos. Se debe indicar en cada intento si el número a acertar es mayor o menor que el indicado por el jugador 2. Además se informará de los intentos que quedan. Al terminar el juego se debe preguntar si se desea jugar otra partida o terminar el programa. Tanto en la petición al jugador 1 como al jugador 2 si el numero no está en el rango 1 a 100 debe volver a pedirlo y no contar intento.

17. a) Escribir un programa que muestre por pantalla los múltiplos de 5 entre 1 y 100 y la suma de los mismos. *Nota: Si lo ves claro, puedes hacer directamente el apartado (b) y saltarte el (a).*
b) A continuación debe pedir un número entre 1 y 50 y mostrará los múltiplos entre 1 y 100 de dicho número y la suma de los mismos. Permitirá la repetición del programa (solo del apartado b).

Ejemplo:

Introduzca nº entre 1 y 50: 25

25, 50, 75, 100

La suma de los múltiplos es: 250

Desea probar otro número (S/N)? N

18. Realizar un programa que calcule el factorial de un número (usar tipo long o real) introducido por el usuario (haz sólo un programa que realice 3 veces el cálculo):

- Con un bucle for.
- Con un bucle do-while
- Con un bucle while

*Nota: El factorial de un número es el número multiplicado por los anteriores incluso el 1. Como por ejemplo, factorial de 7 es $7*6*5*4*3*2*1$. La excepción es el 0 cuyo factorial es 1 ($0!=1$)*

19. a) Realizar un programa que dibuje una ristra de N asteriscos en pantalla (N es un número introducido polo usuario).

Ejemplo:

Cuantos asteriscos deseas dibujar? 10

- b) Repetirlo pero dibujando la ristra en diagonal. El usuario escogerá se la dibuja de izquierda la derecha o de derecha la izquierda.

COLEXIO VIVAS S.L.	RAMA:	Informática	CICLO:	DAM		
	MÓDULO	Programación				CURSO: 1º
	PROTOCOLO:	Boletín clase	AVAL:		DATA:	
	AUTOR	Francisco Bellas Aláez (Curro)				

20. a) Visualizar en pantalla una figura similar a la siguiente:

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

siendo variable (decidido por el usuario) el número de líneas a mostrar.

b) A continuación, y con la misma cantidad de líneas que en el caso anterior, dibuja una pirámide como esta:

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

Ejercicios Avanzados

Estos ejercicios son más complicados y buscan que aquellos que terminen el boletín se enfrenten a un desafío mayor. Se recomienda al alumno que lea todos los ejercicios y realice el que le resulta más atractivo o que se vea capaz en un tiempo razonable. También puede inventarse uno.

*La idea no es ajustarse sólo a la especificación del ejercicio sino además hacerlo claro y amigable para el usuario, **fuerte** frente a errores que pueda cometer a la hora de introducir datos y aplicando **creatividad**.*


El alumno no debe preocuparse de terminarlos en el tiempo que hay para este boletín pues los ejercicios opcionales pueden ser entregados en cualquier momento de la evaluación enviándolos al correo del profesor.

21. Carreras de “caballos”. Se pide al usuario que seleccione un caballo entre el 1 y el 4. Una vez seleccionado empieza la carrera: Aleatoriamente se selecciona un caballo entre 1 y 4 y se mueven entre una y 3 posiciones. Se debe ir viendo como avanzan todos los caballos. Ver las notas del final.

22. Realizar un sistema de tarificación telefónica. Se tienen los siguientes datos:

- El establecimiento de llamada y primer minuto cuestan 0,30€.
- Tarifa normal: De lunes a viernes de 8:00-13:59 y de 16:00-19:59 el precio es de 0,20€ el minuto adicional.
- Tarifa reducida: De lunes a viernes de 14:00-15:59 y de 20:00-21:59 el precio es de 0,15€ el minuto adicional.
- Tarifa superreducida: de lunes a viernes de 22:00-7:59 y sábados y domingos todo el día el precio es de 0,10€ por minuto adicional.
- Los precios anteriores son para llamadas locales. Si la llamada es nacional, el establecimiento de la llamada es el mismo pero se incrementan el resto de los precios en un 10%, y si es internacional en un 40%
- Se debe tener en cuenta que si una llamada comienza en un horario y acaba en otro hay que tener en cuenta distintas tarifas. *NOTA: Este apartado dejadlo para el final. Es recomendable que se compruebe minuto a minuto en qué horario cae y aplicar la tarifa correspondiente.*

Por tanto se le debe pedir al usuario como dato: Día de la semana, hora y minuto de inicio de la llamada, duración y tipo de llamada (local, nacional o internacional). Finalmente se le dará al usuario el coste total de la llamada.

	RAMA:	Informática	CICLO:	DAM		
	MÓDULO	Programación				CURSO: 1º
	PROTOCOLO:	Boletín clase	AVAL:		DATA:	
	AUTOR	Francisco Bellas Aláez (Curro)				

23. Juego “Atrapa al gusano”. La parte principal del juego constará de una tabla de 10x10 elementos. La tabla se irá dibujando poco a poco con una letra aleatoria minúscula excepto en cierta coordenada (también ha de ser aleatoria). El efecto debe ser: Muestra la letra (o el gusano) y tras una breve pausa tapa dicha letra o gusano con el símbolo @ (u otro cualquiera). Cuando se acabe de dibujar la tabla, se le dará al usuario 3 intentos para decir dónde se encuentra el gusano. Si falla acaba la partida y pasa al menú principal para empezar otra o terminar. Si acierta pasa al nivel siguiente y aumenta su puntuación dependiendo de los intentos gastados y del nivel en que se encuentre. A medida que se aumenta el nivel se disminuirá el tiempo en que se muestran las letras/gusano. También se puede jugar con disminuir el número de intentos al aumentar el nivel o incluso aumentar el tamaño del tablero.

Algunas cosas a tener en cuenta para realizar los juegos:

- Para hacer una pausa se usará el comando `Thread.sleep(ms)`; donde ms son los milisegundos de pausa deseados. Para poder usar dicho comando es necesario realizar el import de `java.lang.InterruptedException`; y la definición del main realizarla de la siguiente forma:

```
public static void main (String[] args) throws InterruptedException{
```
- Para sacar números aleatorios se utiliza la función `Math.random()`. Pero esta da un número aleatorio real entre 0 y 1. Piensa como harías para obtener un entero entre 1 y 10 o una letra minúscula mediante operaciones y conversiones. Si no se te ocurre busca en internet o, finalmente, pregunta al profesor.
- Para escribir un carácter en pantalla y luego volver para atrás y escribir otro usa el carácter de escape del backspace: `\b`.
- Pídele al profesor la clase `Gusano.class` que tiene una demo de lo que podría llegar a hacer el programa.