

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

Boletín de prácticas de Programación 3: Métodos y archivos

Notas:

- En ciertos ejercicios se pide simplemente una o varias funciones. Evidentemente hay que hacer un pequeño programa en el main (como quiera el alumno) para probar que dichas subrutinas funcionan, por lo que conviene hacer llamadas a las funciones con distintos datos o preparar una entrada de datos por parte del usuario.
- Se recomienda realizar todos los ejercicios del boletín antes de introducirse en las partes "opcionales" de los distintos ejercicios.
- Aunque el ejercicio no lo indique (sobre todo en los últimos) **es necesario** aplicar modularidad en la medida de lo posible evitando repetir código y minimizando el tamaño del programa principal.
- **Es obligatorio comentar** las funciones tal y como se vio en clase. No se valida la función que no esté comentada para javadoc.
- Evita buscar soluciones en Internet, no es una buena idea. Trata de pensar tú la forma de realizar los ejercicios. Ver una solución hecha debería ser el último recurso cuando ya se ha probado casi de todo. Sí es práctico para comparar o ver otras soluciones a algo que hayas resuelto.
- Los ejercicios que empiezan por doble asterisco (**) implica que tiene trabajo con archivos. Sáttelos si aún no se vio esa parte en teoría.

1. En el mismo archivo haz las siguientes funciones:

a) Codifica un método que deje en pantalla **n** líneas en blanco (es un parámetro). No devuelve nada.

b) Escribe una función denominada **par** con un único parámetro entero. **Devuelve** true si el parámetro es número par y false si no lo es. No muestra nada.

c) Escribir una función que tenga un argumento (otra forma de llamar a un parámetro) de tipo entero y que devuelva la letra 'P' (devuelve char) si el número es positivo o cero y la letra 'N' si es negativo. Intenta hacerlo con el operador ternario (ver Apéndice I del Tema 2); si no te sale hazlo con if.

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

d) En el programa principal primero pides el nombre del usuario. A continuación dejas 10 líneas en blanco mediante la 1ª función. Luego pides un número entero al usuario e indicas si es par y positivo o negativo.

****2.** Amplia el programa anterior (directamente el main) para que guarde en un archivo los siguientes datos: el nombre en una línea, el número en la siguiente y si es par/impar y positivo/negativo.

El archivo queda algo así:

Hay que hacerlo con lo aprendido en archivos, primero modifico el programa anterior para hacerlo escrito y después un archivo de lectura.

Minerva McGonagall

-5

El n.º es impar in negativo.

3. Realiza un programa que halle la superficie de un cilindro sabiendo que:

$$\text{Superficie} = 2 * \pi * \text{radio} * \text{altura}$$

Debe constar de 3 métodos:

- **pedirDato:** pide un dato real usuario comprobando que sea un valor positivo. Al final devuelve dicho dato.
- **superficieCilindro:** Se le pasa como parámetros el radio y la altura, hace el cálculo de la superficie y lo devuelve (sin **nada** de interfaz de usuario).
- **mostrarDato:** Función que tiene un real como parámetro y un String. Muestra en la misma línea el String y a continuación el real con 3 decimales. No devuelve nada.

Por tanto en el programa principal (main) solo debe llamarse a los tres métodos (y declarar variables auxiliares para recoger los datos). Dos veces al primero para pedir primero radio y después la altura, luego se llama al segundo para hacer el cálculo, y finalmente se llama al tercero para mostrar el resultado. Si lo necesitas usa variables intermedias para quedar con resultados de las funciones.

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

4. a) Función año bisiesto. Realizar una función denominada *bisiesto* a la cual se le pasa un año como parámetro y devuelve *true* si dicho año es bisiesto y *false* en caso contrario.

Un año es bisiesto cuando:

- Es múltiplo de 4 (P. ej 1984)
- Pero los múltiplos de 100 no lo son (Por ejemplo 1800)
- Salvo si a su vez son múltiplos de 400 que caso sí lo son (p. ej. 2000)

b) En el programa principal se hará un bucle que pida continuamente años al usuario hasta que introduzca el año 0 momento en el cual el programa termina.

****5.** Haz un programa que realice las siguientes tareas:

Pide un año al usuario y guarda en un archivo dicho año en la primera línea indicando si es o no bisiesto y a continuación todos los bisiestos desde ese año hasta el año actual (cada uno en una línea).

Un ejemplo de como puede quedar el archivo:

```
2015 No es bisiesto.
2016
2020
2024
```

A continuación lee el archivo y muestra en pantalla todos los años menos el primero. Debe estar separados por comas en la misma línea y ocupando 6 caracteres.

Si este programa lo tienes en la misma carpeta que el de bisiesto no es necesario que copies la función, puedes acceder a ella poniendo `NombreClase.nombreFuncion`. Por ejemplo:

```
Bol3Ejer4.bisiesto(2024);
```

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

6. a) Realizar una función que halle y devuelva la potencia de un número (**No se permite usar funciones de Math**). La base puede ser real y el exponente entero puede ser negativo (recuerda que $a^{-b} = (1/a)^b$).

Parámetros del método: la base y el exponente. Valor devuelto: la potencia

b) Realiza un nuevo método (en la misma clase) que muestre en pantalla las **n** primeras potencias de un número **a** (n y a serán parámetros del método).

Por ejemplo, si los parámetros son a=2 y n=4 mostrará: 1, 2, 4, 8.

Para hacerlo deberás llamar al creado en el apartado anterior para calcular las potencias, no las puedes calcular de nuevo en este método.

c) Escribe un método que sume la progresión geométrica (x puede ser real) y devuelva el resultado.

$$1+x+x^2+x^3+x^4+\dots x^n$$

Se debe utilizar la función potencia (sin modificarla, sólo llamándola) programada anteriormente.

Por ejemplo si x=2 y n=4 mostrará 15 por ser el resultado de 1+2+4+8.


Parámetros del método: x y n. Valor devuelto: resultado de la progresión.

****7. a)** Haz una función llamada saveFile a la cual se le pasa una cadena que representa el nombre de un archivo. Dicha función debe leer un archivo completo y devolverlo como string.

Para probarla en el main usa el archivo del ejercicio 2.

b) Realiza una función denominada appendFile a la que se le pasan dos cadenas, la primera es el nombre de un archivo y la segunda es un texto que debe **añadir** al final del archivo. Para ello primero lee el archivo, concatena la nueva cadenas y lo guarda de nuevo. Utiliza la función creada en (a) para la lectura.

c) Realiza una función denominada appendFile2 que hace lo mismo que la realizada en (b) pero usando el método explicado en el Apéndice II de los apuntes.

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

8. a) Escribir una función que se le pase un número y que devuelva *true* o *false* dependiendo de si dicho número es o no es primo.

b) Para probarla haz un programa principal que pida un n.º mayor que 2 al usuario y muestra los números primos menores que dicho número.

Nota: Un número es primo si y sólo si es divisible únicamente por 1 y por él mismo. Por tanto para saber si un número es primo se debe dividir por todos los números menores que él y mayores que 1, y si alguna de esas divisiones exacta entonces el número NO es primo. Existen métodos de optimizar lo anterior, piensa a ver si se te ocurre alguno. El 1 actualmente no se considera primo.

****9.** Modifica el ejercicio anterior de los números primos (solo el main) de forma que hagas un menú clásico con estas opciones:

1. Mostrar primos menores que uno dado: Esto es lo que ya estaba del ejercicio previo.
2. Primos en archivo: Pide el nombre de un archivo que debe contener números, cada uno en una línea. Lee dicho archivo y muestra dichos números en pantalla diciendo si son o no primos.
3. Archivo con primos: Similar a 1, se le pide un número mayor que 2 al usuario y guarda en un archivo todos los primos desde 2 hasta el que meta el usuario. Estarán todos en la misma línea separados por punto y coma (;).
- 4 Salir: El programa solo finaliza si se selecciona esta opción.

10. a) Realizar un método que permita hallar el área de un rectángulo o de un triángulo rectángulo a partir de la base, la altura y un parámetro booleano (denominado bandera, flag o interruptor) para decidir si se trata de un rectángulo o un triángulo. Es decir, si se pasa como parámetro *true* hace el cálculo del área de un rectángulo, si se pasa *false* calcula el del triángulo.

Devuelve el área. Parámetros del método: base, altura y la bandera.

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

b) En el main codificar un menú clásico con las siguientes opciones:

- Area de un triángulo (base*altura/2)
- Area de un rectángulo (base*altura)
- Area de un cuadrado (lado*lado)
- Area de un círculo (Pi*radio²)
- Salir

Las tres primeras opciones deben realizarse **utilizando el método creado en el apartado anterior** (por supuesto, **sin modificarlo**). Para la cuarta debe crearse un nuevo método con parámetro radio.

11. a) Realizar una función que devuelva el factorial de un número según se definió en el boletín anterior (recuerda que 0!=1). Hazlo con parámetro int pero que devuelva double o long.

b) El cálculo del coseno de un ángulo se puede aproximar por el siguiente cálculo (aproximación de Taylor):

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

Realizar un método que realice y devuelva dicho cálculo (hazlo al menos hasta el termino de potencia 4. Si quiere mete más). Debes usar la función factorial hecha en el apartado anterior y la de potencia hecha en un ejercicio previo.

c) Realizar un programa que muestre por pantalla los cosenos de los ángulos de 0.1, 0.2, y hasta 1 radian. Además en cada línea mostrará el resultado del coseno según tu función y el error absoluto obtenido de restarlo de la función Math.cos(). Usa 5 decimales de aproximación para todo y que los valores ocupen 8 posiciones.

Opcional: Realiza la función factorial aplicando recursividad (ver apéndice de apuntes, evita buscar en internet).

12. a) Codificar un programa que genere una quiniela aleatoria. Es decir, se deben dar 14 resultados aleatorios como 1, X ó 2 indicando delante el número de partido y los resultados alineados en una columna (no uses \t, si no ajuste con printf). Se debe realizar al menos una función que devuelva un 1 una X o un 2 (char o String) aleatorio.

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

b) Realizar la quiniela ponderada, es decir, haz otra función que devuelva 1, X o 2 pero que la probabilidad de sacar 1 sea del 60%, la de sacar X sea 25% y la de sacar un 2 sea un 15%.

En el main debes dar a elegir quiniela normal o ponderada.

Pista: Sacar un número aleatorio entre 1 y 100. Si el resultado es menor o igual que 60, se asigna un 1 al resultado, si es entre 61 y 85 (60+25) se le asigna una X y si es entre 86 y 100 se le asigna un 2.

****13. a) Datos d20.** Realizar un programa que pida al usuario su nombre y 3 números distintos al usuario entre 1 y 20 (usar 3 variables) y luego que el ordenador "tira" dos dados de 20 caras. Si el segundo dado sale repetido, debe volver a tirarse hasta que salga distinto.

Debe indicar los aciertos que ha tenido el usuario. Evita repetir código haciendo funciones. El usuario debe poder repetir el juego al finalizar.

b) Modifica el programa anterior para que el usuario pueda elegir la cantidad de caras de los dados antes de jugar.

Además en un archivo de récords debe añadirse en una nueva línea el nombre de usuario, el número de caras del dado ocupando 4 posiciones y los aciertos ocupando 4 posiciones (debes guardar usando printf todo en una nueva línea).

Cuando el usuario decida finalizar, antes de terminar el programa se mostrará el archivo de récords entero.

Opcional: Crear un interfaz más agradable. Se le mostrará un tablero con 20 números (haz una tabla de 4 filas y 5 columnas) y aparecerán en color verde los aciertos, en rojo sacados por el ordenador y no acertados y en azul los seleccionados no acertados.

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

14. Juego Craps: Se desea simular este juego de dados. Para ello:

- Se simula que el jugador tira dos dados de seis caras y se calcula la suma de ambos.
- Si la suma es 7 u 11 en la primera tirada el jugador gana.
- Si la suma es 2, 3 o 12 en la primera tirada (se denomina craps) el jugador pierde (gana la CPU).
- Si la suma es un nº entre 4 y 10 salvo el 7, dicha suma son los puntos del jugador.
- Luego tira la CPU con las mismas reglas. Si al final ambos sacan puntuación gana la de mayor valor o empate en caso de igualdad.

Deben existir al menos las siguientes funciones:

- **tirada:** Tira dos dados, muestra sus valores en pantalla y devuelve la suma.
- **comprobacion:** Se el pasa un valor y devuelve -1 si pierde, 0 si gana o la puntuación en otro caso.

Como siempre haz otras para organizar código y evitar repetir código.

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

Ejercicios Avanzados (Opcionales)

15. West Bank: Eres el valeroso sheriff de un perdido pueblo del viejo oeste y te has enterado que una horda de bandidos ha decidido atracar el banco. Dicho banco tiene tres puertas de entrada numeradas como 1 y 2. En el juego llegan bandidos o clientes y debes actuar en consecuencia. Para ello la secuencia de juego será de esta manera:

El sheriff (usuario) elige puerta (1 o 2) y si dispara o no.

Alguien abre una puerta aleatoria (1 o 2). Puede ser un bandido o un cliente.

Si la puerta del sheriff coincide con la de la persona que entra y este es un bandido sucede lo siguiente:

- Si el sheriff había decidido disparar, mata al bandido y gana 1 punto.
- Si el sheriff había decidido no disparar tiene un 50% de probabilidad de perder una vida (el juego comienza con 3 vidas).

Si la puerta del sheriff coincide con la de la persona que entra y este es un cliente sucede lo siguiente:

- Si el sheriff había decidido disparar, tiene un 50% de posibilidades de matar al cliente. En caso de matarlo pierde una vida.
- Si el sheriff había decidido no disparar el cliente puede entrar sin problema. Por cada 3 clientes que entran se le da al sheriff una vida extra.

Busca modificaciones en probabilidades para hacer el juego lo más entretenido posible.

Nota: La idea de este juego está basada de forma muy básica en el West Bank de Dinamic o Bank Panic! de Sega

<https://www.youtube.com/watch?v=hZNaBgTVbls>

<https://www.youtube.com/watch?v=VcWAd-MttfQ>

<https://www.file-hunter.com/MSX/index.php?id=westbank>

<https://www.playretrogames.com/5503-bank-panic>

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

16. Calendario Perpetuo: Escribir un programa mediante subrutinas que permita ver el mes de cualquier año de la forma

```

L      M      X      J      V      S      D
      1      2      3      4      5
6      7      8      9     10     11     12
...
27     28     29     30

```

El usuario indicará únicamente mes y año que desea visualizar. Para saber en qué día de la semana correspondiente a una fecha dada se debe buscar en la web ya que hay varios válidos. Una vez encontrado el algoritmo se usará para obtener el primer día del mes y el resto se colocan a partir de este.

Da la posibilidad de guardar el mes en un archivo.

17. Realiza un juego conversacional sencillo en consola. Puedes tomar como referencia **Zork** (dejo abajo varios enlaces). Haz, eso sí, un mapa más pequeño y una historia más breve. Modularízalo correctamente. Puedes usar archivos para guardar textos o incluso elementos que describan una estancia. De esta forma cada estancia puede estar definida en un archivo que a su vez indica a que otras estancias (archivos) puede ir, de esta forma la función de lectura y presentación podría llegar a ser la misma.


Juego en español:

<http://iplayif.com/?story=http%3A//media.textadventures.co.uk/games/GQLrNEIS8k26ddJ2nco6ag/Zork.gblorb>

Juego original: http://textadventures.co.uk/games/play/5zyoqrsugeopel3ffhz_vq

Mapa: [http://www.caad.es/sites/default/files/descargas/Juegos/Glulx/Map%20\(Eng\).jpg](http://www.caad.es/sites/default/files/descargas/Juegos/Glulx/Map%20(Eng).jpg)

Video demostrativo: <https://www.youtube.com/watch?v=xzUagi41Wo0>

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

18. Realiza un programa que represente algunas funciones matemáticas como seno, coseno, tangente, una parábola, etc... permitiendo modificar alguno de los parámetros. Evidentemente se dibujaran en una resolución muy baja usando | para el eje vertical, - para el horizontal y * para la función.

19. Realizar el juego de los cañones (Versión antigua del Angry Birds o Worms). Para dos jugadores o un jugador contra la CPU. Aparecen un cañon a cada lado de la pantalla apuntando al otro. Ambos están en un terreno con distinta elevación (que será aleatoria en cada partida) y dispararán de forma alterna dando indicaciones de angulo y fuerza. Puede haber también fuerza del viento u otros factores que afecten a los disparos.

Nota: este último opcional es complicado pues incluye un manejo de los códigos ANSI más fuerte así como la creación de la física del juego (un disparo parabólico).

20. Programar un juego "mini-hundir-la-flota" contra el ordenador. Se hará con un barco de 2 casillas en un tablero de 10x10 posiciones. Las columnas con una letra de la A a la J y la fila con un número del 1 al 10.

La CPU sacará coordenadas aleatorias y dirección aleatoria (horizontal/vertical). Se debe sacar una coordenada aleatoria, y una dirección aleatoria (horizontal o vertical). A partir de ahí se crean las 2 coordenadas del barco.

Si el ordenador acierta una casilla no sigue tirando aleatoriamente si no que lo intenta en las coordenadas de alrededor en horizontal o vertical (no diagonal).

Si sabes algo de arrays, puedes hacerlo más completo con estas estructuras.

También puedes usar archivos para marcar tiradas.

21. Mejorar alguno de los opcionales del boletín anterior reestructurándolo en funciones.