



PROPUESTA DE CÓMO ENFOCAR LA SOLUCIÓN DE LAS PREGUNTAS DEL EXAMEN

Tenemos esta pregunta:

PREGUNTA 3, problema de programación en JavaScript: (3,50 puntos)

La serie de Fibonacci es una serie de números enteros positivos: 0, 1, 1, 2, 3, 5, 8, 13, 21, ... que comienza por los términos 0 y 1 y tiene la propiedad de que cada término posterior es la suma de sus 2 predecesores en la sucesión ($2 = 1 + 1$, $3 = 2 + 1$, ...).

Desarrolla un script que solicite un número entero N entre 1 y 8 y después produzca un cuadrado de números que son los miembros de la serie de Fibonacci, de longitud de lado N unidades, como el formato siguiente, donde cada asterisco representa un término de la serie de Fibonacci.

* *

* *

* *

Cada número o término debe aparecer formateado como se indica en los ejemplos. Se recomienda realizar una función `imprimirTermino` para que cada término o número de la serie se imprima:

- ocupando tantos caracteres como ocupe el mayor de los términos
- cada elemento que no sea un dígito (0 a la izquierda) se imprima como un punto "."

Cada término va separado del anterior por un espacio.

Por ejemplo: (las entradas aparecen en fondo gris, las salidas calculadas en fondo azul)

(Enviar y Reset representan botones html)

Introduzca un número comprendido entre 1 y 8: 4

Enviar Reset

```
..1 ..1 ..2 ..3
..5          ..8
.13         .21
.34 .55 .89 144
```

Introduzca un número comprendido entre 1 y 8: 3

Enviar Reset

```
.1 .1 .2
.3 .5
.8 13 21
```

Una forma muy recomendable de enfocar estas preguntas es tirando de trabajo hecho en boletines (hay código ejemplo de propuestas de solución de casi todos los boletines en la carpeta de código de ejemplo):

Observaciones:

1) PIEZA 1: en el [DWECC-Core-JS-Boletín 02-bucles parte 1.odt](#), ejercicio 9, se trabajó el problema de crear el cuadrado de asteriscos vacío

2) PIEZA 2: en [DWECC-Core-JS-Boletín 04-funciones.odt](#), ejercicio 4, se trabajó crear una función que genere los términos (números) de la serie de Fibonacci:

- llamémosla: `fibonacci` (`termino`)

3) PIEZA 3: podemos crear un par de funciones sencillas:

- `numeroDeTerminos` (`ladoCuadrado`): calcula el número total de términos a imprimir
 - en el primer ejemplo del enunciado número = 4 --> nos da 12 términos
 - en el segundo ejemplo del enunciado número = 3 --> nos da 8 términos



- `longitudUltimoTerminoFibo (termino)`: indica cuántos dígitos (ancho de campo) tendrá el último término de la serie de fibonacci a imprimir (es decir, el `anchoDeCampo`)
 - en el primer ejemplo del enunciado número = 4 --> nos devuelve: 3
 - en el segundo ejemplo del enunciado número = 3 --> nos devuelve: 2
- `imprimirFiboConFormato (fibonacci (termino), anchocampo)`: por ejemplo
 - `imprimirFiboConFormato (fibonacci (2), 2)` imprime: `.2`
 - `imprimirFiboConFormato (fibonacci (11), 3)` imprime: `.89`

Por lo tanto: si en la PIEZA 1 sustituimos cada línea de código que imprime

- un asterisco por:
`imprimirFiboConFormato (fibonacci (termino), anchoDeCampo)`
- un espacio por:
`imprimirEspacios (anchoDeCampo)`

Tenemos el problema resuelto (nota: `anchoDeCampo` lo puedes calcular con `longitudUltimoTerminoFibo (termino)`)

Resumiendo:

1) Leer el enunciado del problema detenidamente

2) Descomponer el problema en piezas o partes

3) Ver que piezas o partes ya tenemos hechas.

En el ejemplo anterior:

- dibujar un cuadrado vacío por dentro
- calcular un término de la sucesión de Fibonacci

4) Elaborar las nuevas.

En el ejemplo anterior:

- `numeroDeTerminos`
- `longitudUltimoTerminoFibo`
- `imprimirFiboConFormato`

5) Acoplar todas las piezas

En el ejemplo anterior, por ejemplo:

1. Copiar el código de solución de DWECC-Core-JS-Boletín 02-bucles parte 1.odt, ejercicio 9 que tengas
2. Añadirle al anterior, la función de DWECC-Core-JS-Boletín 04-funciones.odt que tengas
3. Añadirle las funciones del paso 4)

6) Probar para varias entradas

7) Depurar errores (si hace falta) (volver al paso 6))

Es decir, no hay que empezar desde cero para resolver estos problemas.

- al empezar el examen, prepara tu repositorio de código de referencia: descárgate e incorpora al entorno Visual Studio Code el código ya trabajado y/o el que estimes oportuno