

Enunciados

problemas de desarrollo

VersiÓn 1.1

24/01/2020

Contenido

[1. Conocimientos Teóricos 2](#_Toc21947186)

[2. Práctica 2](#_Toc21947187)

[2.1 Problema 1 2](#_Toc21947188)

[2.2 Problema 2 3](#_Toc21947189)

[2.3 Problema 3 3](#_Toc21947190)

[2.4 Problema 4 4](#_Toc21947191)

[2.5 Problema 5 5](#_Toc21947192)

[2.6 Problema 6 5](#_Toc21947193)

1. Conocimientos Teóricos

Considerando como lenguaje JavaScript, atender a las siguientes consignas. Las respuestas deben ser cortas, claras y concisas:

1. Escribir una función que permita multiplicar dos números y obtener el resultado, pero de esta forma**:  multiplicar(a)(b);**
2. ¿Cuál es la diferencia entre “==” e “===”?
3. ¿Cómo se verifica que un número es un entero?
4. ¿Cómo se hace posible que el siguiente código funcione?

var arr = [1, 2, 3, 4, 5];

var avg = arr.promedio();

console.log(avg);

1. Explicar que es una función de callback y dar un ejemplo.
2. ¿Por qué motivo la siguiente comparación arroja “false” como resultado y a que se debe?

0.1 + 0.2 == 0.3

1. ¿Cuál es el resultado del siguiente código, y por qué motivo? *Pista: No es el mismo motivo que el anterior.*

123456789101112.00 + 0.01

1. Práctica
   1. Problema 1

Dada la siguiente secuencia iterativa definida por el conjunto de enteros positivos:

n --> n/2 (si n es par)

n --> 3\*n+1 (si n es impar)

Usando la regla anterior, y empezando por ejemplo en 13, se genera la siguiente secuencia:

13, 40, 20, 10, 5, 16, 8, 4, 2, 1

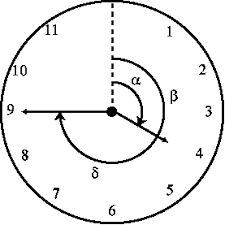
Puede verse que esta secuencia (comenzando en 13 y finalizando en 1) contiene 10 términos. Se sabe que las secuencias para todos los números terminan en "1".

La pregunta a responder es: ¿Cuál número inicial, por debajo de 1.000.000 (un millón) produce la cadena más con mayor cantidad de términos?

Se pide entregar el programa que implementa el algoritmo, en el lenguaje y tecnología que resulte apropiado.

* 1. Problema 2

Dada una hora en formato “HH:mm”, donde “HH” es un número entero de 0 a 11, y “mm” es un número entero de 0 a 59, obtener los valores para los ángulos de la aguja de las horas y minutos. Debe considerarse el ángulo con respecto a la vertical del cuadrante del reloj, como se indica en la imagen:



Se espera como resultado de la llamada al programa / método / función los valores para las variables alfa (horas) y beta (minutos).

* 1. Problema 3

En este caso se pide resolver **el problema 1 de los ejercicios obligatorios** de forma óptima considerando las variables de **tiempo de ejecución y memoria**, minimizando el costo total en su relación CT = T + 5\*M, donde T es el tiempo total de ejecución en milisegundos para encontrar la cadena y M es la cantidad de memoria máxima usada medida en KB.

**Ejemplo**: Se propone la solución A cuyo tiempo de ejecución promedio es de 1580ms y consume 16K de memoria. Luego se propone la solución B cuyo tiempo de ejecución promedio es de 1900ms y consume 8K de memoria. Las relaciones serían:

CT\_A = 1580 + 5 \* 16 = 1580 + 800 = 2380

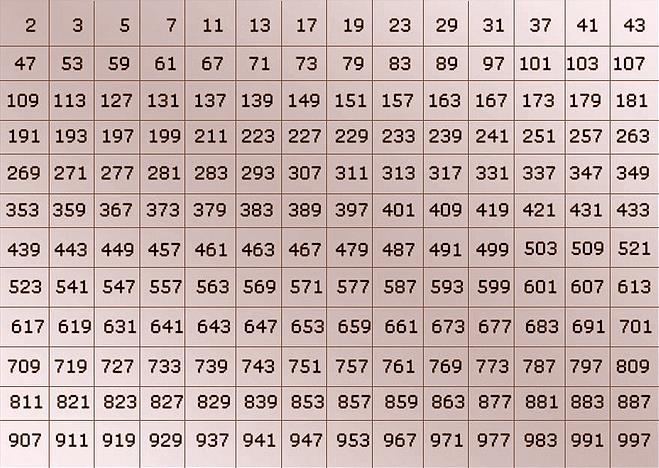
CT\_B = 1900 + 5 \* 8 = 1900 + 400 = 2300

Es superior la solución B.

* 1. Problema 4

Escribir un programa tal que, dada la matriz de la imagen y dos números que indican filas y columnas como parámetro, se arme una sub-matriz y se muestre el contenido recorriendo sus celdas en el sentido de las agujas del reloj, desde el exterior hacia el centro, comenzando por la celda [0,0].

La matriz dada puede ser considerada constante y no es parte del problema la generación de números primos. La submatriz deberá contener los valores correspondientes a la matriz dada.



**Ejemplo**

Si la entrada “filas=3, columnas=4”, se forma la sub-matriz:

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 3 | 5 | 7 |
| 47 | 53 | 59 | 61 |
| 109 | 113 | 127 | 131 |

La salida debe ser: **2, 3, 5, 7, 61, 131, 127, 113, 109, 47, 53, 59**

* 1. Problema 5

Dado un tablero de ajedrez, calcular la posición de N reinas, tal que ninguna de ellas amenace a otra. La cantidad de reinas será la misma que la cantidad de filas/columnas del tablero (N reinas en un tablero de NxN cuadros). La verificación se hará con N = 15.

Escribir un algoritmo que calcule y muestre las soluciones a un tamaño de tablero dado por el número N. La forma de mostrar cada solución deberá ser en notación PGN.

No pueden usarse los algoritmos conocidos de [F. Nauck](https://es.wikipedia.org/w/index.php?title=Franz_Nauck&action=edit&redlink=1), S. Günther, J.W.L. Glaisher ó E. Dijkstra. Plantear una alternativa innovadora, no es requisito que sea óptima.