

PARTE 1

El objetivo de esta primera parte es completar la mayor cantidad de los problemas propuestos en el periodo de 60 minutos (de 2:15pm a 3:15pm) y enviarlos de la siguiente manera:

- Cada problema resuelto en un archivo individual respetando los nombres indicados.
- Todos los archivos son programas PARA LA LÍNEA DE COMANDOS, no son programas para un servidor web, el browser ó aplicación de escritorio.
- Si el programa acepta parámetros por la línea de comandos y la invocación es incorrecta debe mostrar un mensaje de ayuda indicando la manera correcta de invocar el programa.
- El programa debe ejecutarse directamente al ser invocado como un script independiente. Ninguna rutina debería ejecutarse si el script es importado desde otro módulo o desde el intérprete interactivo.
- El código fuente de cada script debe estar codificado utilizando el estándar UTF-8.
- Una vez concluído el período de 60 minutos enviar todos los programas disponibles en un archivo .zip a la dirección de correo electrónico aprendices.de.hacker@gmail.com indicando como asunto "Problemas - Aprendiz de Hacker".
- Todos los programas deberán estar escritos en un mismo lenguaje de programación. Ud. podrá escoger entre Java, C#, Javascript, PHP, Python ó Ruby.

PROBLEMA 1

Nombre del script: RandomStations.(java|cs|js|php|py|rb)

Escribir una programa que implemente una función llamada `random_stations` que recibe dos parámetros: el número de frecuencias distintas a generar aleatoriamente y el tipo de banda representado por las cadenas "AM" y "FM" respectivamente. Tener en consideración que las emisoras de radio AM ocupan un ancho de banda de 1060 kHz entre las frecuencias 540 kHz y 1600 kHz y que solamente pueden existir 106 frecuencias distintas distribuidas de manera uniforme entre ambos valores. En el caso de las emisoras de radio FM el ancho de banda es de 20 MHz y se distribuye uniformemente entre las frecuencias 88.1 MHz y 108.1 MHz pudiendo ser un máximo de 100 frecuencias distintas. Al ejecutarse, el programa debe utilizar dicha función para imprimir las frecuencias generadas en pantalla considerando cero decimales en el caso de las frecuencias de AM y un decimal en el caso de las frecuencias de FM.

PROBLEMA 2

Nombre del script: LeapAccumulator.(java|cs|js|php|py|rb)

Escribir un programa que imprima en pantalla la suma acumulada de los años bisiestos entre el año cero (0) y el año en curso (obtenido programáticamente) teniendo en cuenta que un año es bisiesto si es divisible por 4 y al mismo tiempo NO es divisible por 100 exceptuando los años

que son divisibles por 400 que sí son bisiestos.

PROBLEMA 3

Nombre del script: Mod10.(java|cs|js|php|py|rb)

Escribir un programa que reciba en la línea de comandos los números de una tarjeta de crédito e imprima en la salida la cadena "SI CUMPLE" o "NO CUMPLE" en caso de que el número de la tarjeta cumpla con el algoritmo "Mod 10" descrito a continuación:

Para verificar si el código cumple o no con el algoritmo "Mod 10" seguir los siguientes pasos:

- 1.- Multiplicar por dos el valor de los dígitos en cada posición par del código.
- 2.- Acumular la suma de cada dígito de los valores obtenidos en el paso anterior y sumarla al acumulado de los valores de los dígitos que ocupan posiciones impares.
- 3.- Dividir el valor de la suma entre 10.

Si el número resultante es divisible por 10 entonces la tarjeta SI cumple con el algoritmo.

Ejemplo:

Paso 1:

4	9	9	2	7	3	9	8	7	1	6
	x2		x2		x2		x2		x2	

	18		4		6		16		2	

Paso 2: $4 + (1 + 8) + 9 + (4) + 7 + (6) + 9 + (1 + 6) + 7 + (2) + 6$

Paso 3: Suma = 70 : El número de tarjeta cumple con el algoritmo Mod 10.

PROBLEMA 4

Nombre del script: RecursiveReverse.(java|cs|js|php|py|rb)

Escribir un programa que acepte una cadena de texto como único parámetro obligatorio e implementa una función recursiva llamada `invertir_cadena` devuelve una nueva cadena

compuesta por los mismos caracteres de la cadena original pero presentes en un secuencia con el orden inverso, utilizando dicha función para imprimir en la salida estándar la cadena invertida.

Tener en consideración lo siguiente:

- a) Toda cadena puede ser expresada como dos partes: cabeza (solo el primer caracter) y cola (a partir del segundo caracter hasta el final de la cadena)
- b) La cabeza corresponde al primer carácter de una cadena o una cadena en blanco en caso de una cadena vacía.
- c) La cola corresponde es una subcadena compuesta por la secuencia de caracteres comprendida desde el segundo carácter de la cadena hasta el último o una cadena vacía en el caso de una cadena compuesta por un único carácter o una cadena vacía.
- d) La cadena inversa de una cadena vacía es otra cadena vacía
- e) La cadena inversa de cualquier otra cadena es la concatenación de la cadena inversa de la cola de la cadena original seguida de la cabeza de la misma.