

Fecha de Entrega: **Marzo 9 de 2015 antes de las 11:50AM COT**

### Instrucciones de Entrega

Todo el código fuente y los datos se debe encontrar en un repositorio público en github con un commit final hecho antes de Marzo 9 de 2015 antes de las 11:50AM COT. El nombre del repositorio debe ser `CM20151_HW3_NombreApellido`, por ejemplo yo debo crear un repositorio llamado `CM20151_HW3_SebastianPerez`. El link al repositorio lo deben enviar a través de **sicuplplus** antes de la fecha/hora límite.

En cada parte del ejercicio se entrega 1/3 de los puntos si el código propuesto es razonable, 1/3 si se puede ejecutar y 1/3 si entrega resultados correctos.

**Bono:** Si se entrega la tarea antes de las 11:50AM COT del Viernes 6 de Marzo los puntos se calificarán sobre 25-25-30-40, es decir la nota máxima posible es 120 en ese caso.

1. 20 pt **Calculando Capicúas** <sup>1</sup> En un cuaderno de ipython llamado `cal_capicuas.ipynb`, responda las siguientes preguntas:
  - (a) 8 pt Imprima una lista de todos los capicúas de 9 dígitos. Cuántos hay?
  - (b) 5 pt Imprima una lista de todos los capicúas de 9 dígitos que también son capicúas en representación binaria. Cada línea de la lista debe ser de la forma: `cap_base10 = cap_bin`. Cuántos hay?
  - (c) 7 pt escriba un programa que verifique si una cadena de caracteres es palíndroma. Un ejemplo de texto palíndromo <sup>2</sup>:

```
Ay! Oír la sonrisa, madre: Lola iba sin ropa.
-Va, se sube, medita.
-Airado, llamé a comer seis ratones.
-Aleve su amor a aves o maleza para robar tesoro.
-Se trabó.
-¡Rara paz!
-El amo se va a Roma.
-Use vela.
-Sé notar, si es remo cae mal; lo daría a ti.
-Déme buses a vapor.
-Ni sabía lo lerda, mas irnos al río, ya.
```

---

<sup>1</sup><http://es.wikipedia.org/wiki/Capicua>

<sup>2</sup><http://www.movimientorever.blogspot.com/2013/02/obras-presentadas-al-iii-premio.html>

2. **20 pt Subiendo Escaleras** Usted tiene que subir una escalera que tiene exactamente  $N$  escalones, numerados de 1 a  $N$ . Con cada paso, usted puede subir de a uno o dos escalones, o más precisamente: Con su primer paso usted puede terminar en el escalón 1 o 2. Si usted está en el escalón  $K$  puede moverse a los escalones  $K + 1$  o  $K + 2$ . Finalmente, tiene que llegar al escalón  $N$ . Su tarea será calcular el número de maneras diferentes de escalar esta escalera.

Escriba su solución en un archivo llamado `subiendo_escaleras.py`

- (a) **8 pt** Cuántas distintas maneras existen de subir la escalera de 13 escalones? Cuántas para una de 15?
- (b) **12 pt** Escriba una función, `escaleras(A,B)` que dados dos arreglos  $A$  y  $B$ , cada uno de  $L$  enteros, devuelva un arreglo de  $L$  enteros que especifique las respuestas consecutivas, es decir, la posición  $i$  del arreglo debe contener el número de maneras distintas de escalar la escalera con  $A[i]$  escalones módulo  $2B[i]$

Por ejemplo, dado  $L = 5$  y:

```
A[0] = 4   B[0] = 3
A[1] = 4   B[1] = 2
A[2] = 5   B[2] = 4
A[3] = 5   B[3] = 3
A[4] = 1   B[4] = 1
```

La función debe resultar en la secuencia  $[5, 1, 8, 0, 1]$ .

Asuma que:

$L$  es un entero en el rango  $[1 - 10000]$ , cada elemento del arreglo  $A$  es un entero en el rango  $[1 - L]$  y cada elemento del arreglo  $B$  es un entero en el rango  $[1 - 20]$ .

3. **25 pt Manipulando Archivos con Python**

El objetivo de este punto es aprender a manipular archivos de datos.

- (a) **5 pt** En la carpeta `test` 100 archivos `.txt` llamarlos todos con un `for` y hacer una animación de las columnas 1 y 2 de estos archivos, El código debe llamarse `animate.py`.
- (b) **5 pt** El archivo `.csv` contiene datos que están separados por `,` lea estos archivos y reemplaze la `verb+,+ por +` y guarde este nuevo archivo como `.dat`. Repita esto pero ahora usando `!` para usar comandos en la terminal. Por ejemplo `! ls` me muestra los archivos en donde está el código. El código debe llamarse `csvtodat.py`.
- (c) **5 pt** En el archivo `Sainte-Beuve.txt` se encuentra el libro de Baudelaire lea las primeras `n` líneas del archivo, donde `n` entra por parámetro (es decir use la librería `sys`). El programa debe imprimir el número de vocales que hay en cada línea. El código debe llamarse `vocales.py`.
- (d) **5 pt** Realice un tutorial en un `ipython` notebook en el cual explique cómo leer y escribir un archivo binario. De ejemplos. Este notebook debe llamarse `binary-tutorial.ipynb`
- (e) **5 pt** Leer el archivo `Houston-security.xls` con la librería `Pandas`, lea cada una de las columnas y responda a qué hora del día es más seguro salir según el delito de asalto agravado. Haga un histograma para la hora del delito para todos los tipos de delitos que hay. El notebook que haga esto debe llamarse `Houston_tenemos_un_problema.ipynb`

4. 35 pt **Predicciones Financieras** Una aplicación interesante del análisis de datos, es en la predicción de eventos financieros. En este ejercicio vamos a calcular algunas predicciones sobre la fluctuación entre peso colombiano y dolar en el último año. (COP/USD). Cree un archivo llamado `prediccion.Rmd` que genere lo siguiente (Recuerde usar el paquete `ggplot2`):
- (a) 5 pt Usando el paquete `Quandl`, descargue la serie de tiempo de precios diarios de COP/USD desde el 2000 hasta 2015-02-23. Imprima los 5 primeros valores de esta serie de tiempo.
  - (b) 10 pt cree una visualización donde se muestre la evolución temporal de COP/USD en este periodo, usando `ggplot2`. El título del gráfico debe contener el rango de tiempo.
  - (c) 10 pt Cree un gráfico donde se vean estos datos año por año. Tip: Puede usar el paquete `lubridate` y el concepto de *facets*.
  - (d) 10 pt Haga una visualización de su predicción de esta serie de tiempo hasta hasta 2017-01-01. Puede utilizar desde un modelo lineal (*lm*) hasta cualquier generalización que le parezca *glm*, *gam*, etc. Describa y justifique su escogencia.