

## Examen M4

06/06/2022

Nombre: \_\_\_\_\_

### Instrucciones

Tendrás un tiempo de 110 minutos para resolver este examen. Antes de resolver cada problema, procura entender los ejemplos de ejecución y los requisitos. Si tienes dudas al respecto, consulta al ayudante presente en la sala.

**En este examen estará permitido consultar el libro guía del curso, en formato digital o impreso.** Para entregar tu examen, haz un archivo `zip` con nombre cuyo formato debe ser `RUT_sin_verificador.zip`, que contenga el programa desarrollado. En cada enunciado de pregunta se especifica cuál debe ser el nombre de archivo. Entrega tu archivo `zip` en el buzón disponible en Canvas.

### Conjetura de Collatz (collatz.py; 2 puntos)

La Conjetura de Collatz es uno de los más famosos problemas en matemática que aún no ha sido resuelto<sup>1</sup>. La conjetura dice que para cualquier entero positivo, si se aplican sucesivamente las operaciones descritas a continuación, se llega en algún momento al valor 1.

$$f(n) = \begin{cases} \frac{n}{2} & \text{si } n \equiv 0 \pmod{2} \\ 3n + 1 & \text{si } n \equiv 1 \pmod{2} \end{cases}$$

Con lo anterior, se genera una sucesión, tal que si se comienza con un valor  $n_1$ , si  $n_1$  es par,  $n_2 = \frac{n_1}{2}$ . Por otro lado, si  $n_1$  es impar, entonces  $n_2 = 3n_1 + 1$ , y así se continúan generando valores de la sucesión hasta que eventualmente se llega a 1. Cuando se llega a 1, la sucesión continúa siguiendo un mismo patrón  $4-2-1$  infinitas veces. Por ejemplo, para el  $n_1 = 20$ , la sucesión es:

20 10 5 16 8 4 2 1 4 2 1 4 2 1 ...

Escribe un programa que (1) pida un entero positivo al usuario, y luego (2) muestre toda la sucesión desarrollada paso a paso hasta que se llega al valor 1 *por primera vez*. Para ello, el programa debe definir la **función recursiva** `collatz(n, step=1)`. El primer parámetro de la función es el valor actual de la sucesión ( $n_i$ ), y el segundo parámetro es el número del término actual de la sucesión a partir de uno.

Si el usuario ingresa el valor 20, y la función es llamada con `collatz(20)` (step vale 1 si no se especifica), el resultado desplegado en consola por la función debe ser el siguiente:

```
1 20
2 10
3 5
4 16
5 8
6 4
7 2
8 1
```

<sup>1</sup>Mantén la calma, no te vamos a pedir que demuestres la conjetura.

Es decir, se muestra el número del término junto con el valor respectivo.

## Evaluación

- [0.5 punto] Correcta implementación del caso base de la recursión en la función `collatz(n, step=1)`.
- [1.5 punto] Correcta implementación de la(s) llamada(s) recursiva(s) a la función `collatz(n, step=1)`.

## Inversiones en Criptoactivos (cripto.py; 4 puntos)

Las criptomonedas son activos financieros muy riesgosos, pues su valorización es principalmente especulativa. En esta parte del examen, procesarás dos conjuntos de datos correspondientes a las dos criptomonedas más transadas, Bitcoin (BTC) y Ethereum (ETH), para generar una serie de reportes:

- [1.0 punto] Abre los archivos `BTC.csv` y `ETH.csv` y carga en memoria todo su contenido, en una estructura de datos que sea funcional a los requisitos que siguen a continuación. Se te recomienda crear una función que sirva para cargar en memoria los datos de cualquiera de los archivos, pues, ambos tienen el mismo formato. Considera que las fechas en los archivos de origen podrían tener saltos, vale decir, podría haber días sin información. Para esto, es recomendable que guardes para BTC y ETH listas que contengan todos los días (formato 'AAAA-MM-DD') para los cuales existen datos, ordenados.
- Despliega un menú<sup>2</sup> con opciones numeradas según el siguiente listado:
  1. [1.5 punto] Gráfico de moneda en un rango de fechas: Se pide al usuario una fecha de inicio, un número de días a contar de la fecha de inicio (para los cuales existen datos), y una moneda (BTC o ETH). Luego se despliega un gráfico de línea con el valor alto (*high*) diario de la moneda para el número de días deseado.
  2. [1.5 punto] Gráfico de días al alza y días a la baja: Para una fecha inicial, y una cantidad de días a contar de ella - ambos datos ingresados por el usuario - genera un despliegue de dos gráficos de barras que contengan para BTC y ETH el número de días que cerraron al alza y el número de días que cerraron a la baja, siempre comparando con el día anterior, y considerando el valor *high*.

---

<sup>2</sup>Se te entrega un código base que implementa el menú.

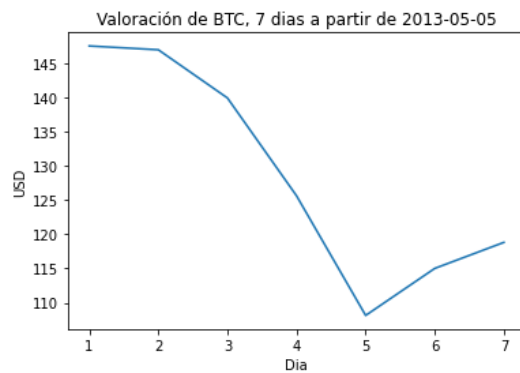
## Ejemplos de Ejecución

```
*****  
***** CRYPTO MARKET *****  
*****
```

1. Gráfico de valorización
2. Gráfico de días al alza y días a la baja
3. Salir

```
*****
```

```
[opcion]>1  
[fecha inicio (AAAA-MM-DD)]>2013-04-29  
[dias]>7  
[moneda (BTC/ETH)]>BTC
```

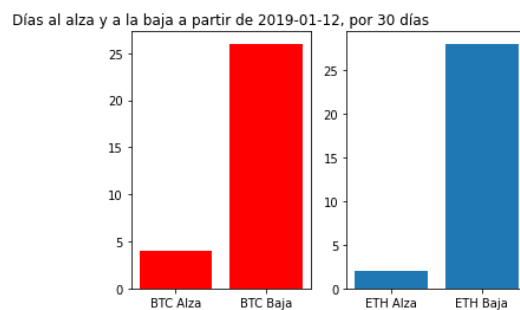


```
*****  
***** CRYPTO MARKET *****  
*****
```

1. Gráfico de valorización
2. Gráfico de días al alza y días a la baja
3. Salir

```
*****
```

```
[opcion]>2  
[fecha inicio (AAAA-MM-DD)]>2019-01-12  
[dias]>30
```



## Sistema de Evaluación del Examen

Cada ítem es evaluado con una escala de 1-5 (1: no presenta solución, 2: presenta esbozo de solución incompleto o con errores graves, 3: solución incompleta o imprecisa, 4: error(es) mínimo(s), 5: excelente) y luego un ponderador (1:0, 2:0,25, 3:0,5, 4:0,75, 5:1,0) es aplicado al puntaje del ítem. El puntaje otorgado a la solución se calcula como la suma de los puntajes parciales obtenidos, aplicándose los ponderadores. La nota final es el puntaje calculado más el punto base.