**INFORME DE LABORATORIO #11**

**ARREGLOS MULTIDIMENSIONALES Y FUNCIONES EN PYTHON**

**OBJETIVO**

Comprender la importancia del uso de arreglos multidimensionales y la aplicación del a técnica *Dividir y Conquistar* para desarrollar soluciones a problemas cotidianos, mediante el lenguaje de programación Python, de tal forma que la suma de los objetivos intermedios permita dar solución al objetivo general planteado para la solución.

**INSTRUCCIONES**

* El informe de laboratorio se debe realizar en parejas y debe ser entregado en la fecha estipulada. A continuación, se presenta la sección del planteamiento, a la que se recomienda que se haga un seguimiento secuencial con el fin de mantener el orden en el informe.

**PLANTEAMIENTO – PARTE A (Valor 3.0)**

**Desarrollar el ejercicio presentado a continuación, en lenguaje Python y en un archivo de nombre *informe11\_A\_NombreApellido.py*. El código debe estar adecuadamente documentado y se deben realizar los respectivos *commits* en cada literal.**

La cadena de panaderías Pan de Mía, ha tenido un crecimiento acelerado en los últimos meses gracias a campañas virales como “Ven con nosotros, no te quedes en casa” y quieren hacer una simulación de los posibles ingresos y egresos que obtendrán en el presente año. Para ello, han decidido tabular la información como se muestra en la Tabla 1, de manera que puedan generar una tabla de ingresos, egresos y ganancias/pérdidas.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Enero | Febrero | Marzo | Abril | Mayo | Junio | Julio | Agosto | Septiembre | Octubre | Noviembre | Diciembre |
| Bucaramanga | 100 | 110 | 178 | 107 | 155 | 127 | 106 | 114 | 130 | 169 | 127 | 103 |
| Floridablanca | 131 | 158 | 103 | 160 | 105 | 127 | 114 | 155 | 114 | 178 | 104 | 160 |
| Girón | 104 | 127 | 114 | 153 | 139 | 150 | 162 | 137 | 173 | 150 | 127 | 114 |
| Piedecuesta | 149 | 150 | 100 | 130 | 159 | 107 | 178 | 101 | 109 | 127 | 130 | 178 |

Tabla 1. Ejemplo de tabulación de valores de cada ciudad.

Los valores que se almacenarán representan millones de pesos y se generarán de manera aleatoria, de acuerdo con rangos sugeridos, los cuales fueron calculados a partir de un riguroso análisis predictivo. Dada la experiencia del líder de su equipo de desarrollo, le propone seguir los siguientes pasos para resolver este problema:

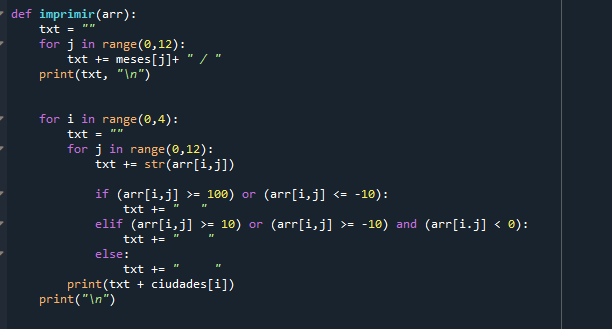
1. Cree una función llamada **generador**, que, dados como parámetros un valor mínimo y un valor máximo, retorne un arreglo bidimensional de 4 filas x 12 columnas que contenga valores al azar entre esos dos rangos.



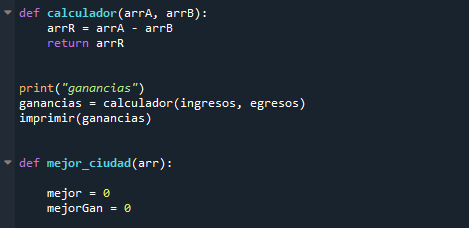
1. Utilice la función **generador** creada, para obtener un arreglo con valores que oscilen entre 100 y 180 y almacene el retorno de dicha función en un arreglo llamado **ingresos**. Utilice nuevamente la función **generador**, para crear un arreglo con valores que oscilen entre 60 y 130 y almacene el retorno de dicha función en un arreglo llamado **egresos**.



1. Cree una función llamada **imprimir**, que, dado un arreglo bidimensional como parámetro, imprima las dimensiones de dicho arreglo y los elementos que lo conforman de manera comprensible para los usuarios (en el contexto del presente problema). Esta función puede serle de utilidad en cada uno de los pasos posteriores, para verificar los resultados que se van obteniendo. Esta función no tiene retorno, ya que sólo se limita a mostrar valores en pantalla.



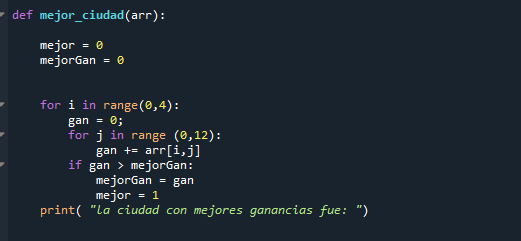
1. Utilice la función **imprimir** creada, para que imprima los elementos contenidos en los arreglos **ingresos** y **egresos** que fueron creados en el punto anterior.
2. Cree una función llamada **calculador**, que, dado como parámetros dos arreglos bidimensionales **A** y **B** (de mismas dimensiones), retorne un arreglo bidimensional **R** que contenga los valores resultantes de restar los elementos del arreglo **A** menos los elementos del arreglo **B**.



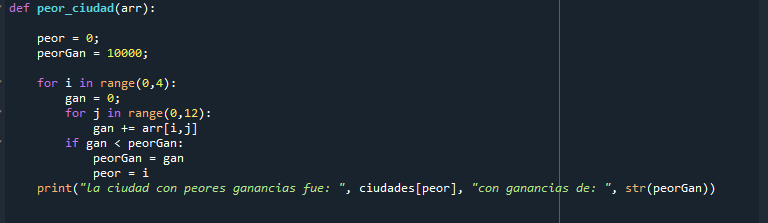
1. Utilice la función **calculador** creada, ingresando como parámetros el arreglo **ingresos** y el arreglo **egresos**. Almacene el retorno de dicha función en un arreglo **ganancias** que contendrá el resultado de restar los valores del arreglo **ingresos** y el arreglo **egresos**, para determinar las ganancias o pérdidas de cada ciudad en cada mes.

Finalmente, cree las siguientes funciones:

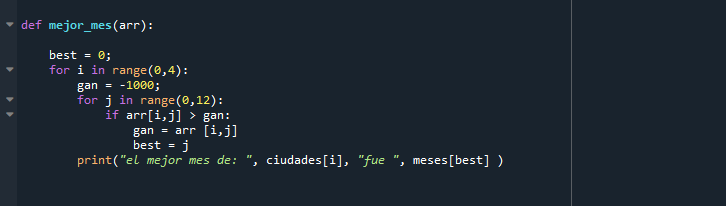
1. Cree una función llamada **mejor\_ciudad**, que, dado como parámetro el arreglo **ganancias,** calcule e imprima la ciudad con mejores ganancias.



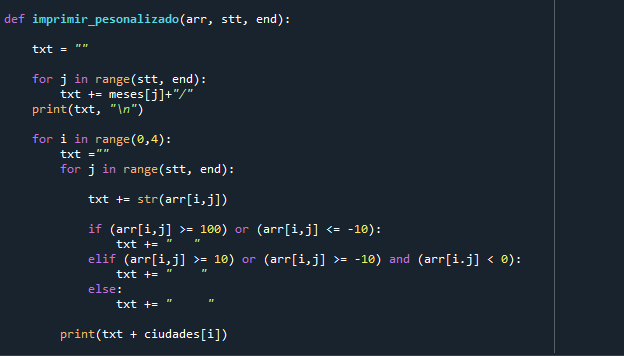
1. Cree una función llamada **peor\_ciudad**, que, dado como parámetro el arreglo **ganancias,** calcule e imprima la ciudad con menores ganancias (o con mayor pérdida).



1. Cree una función llamada **mejor\_mes**, que, dado como parámetro el arreglo **ganancias,** calcule e imprima cual fue el mejor mes de ganancias en cada una de las ciudades.



1. Cree una función llamada **imprimir\_personalizado**, que, dado como parámetro cualquiera de los arreglos creados en este laboratorio y un mes de inicio y un mes de fin**,** imprima los valores contenidos en todas las ciudades en los meses indicados. Así, si se recibe el arreglo **ingresos**, el mes 3 y el mes 6; la función deberá imprimir los ingresos de las ciudades desde el mes de marzo hasta el mes de julio.



**PLANTEAMIENTO – PARTE B (Valor 2.0)**

**Desarrollar el ejercicio presentado a continuación, en lenguaje Python y en un archivo de nombre *informe11\_B\_NombreApellido.py*. El código debe estar adecuadamente documentado y se deben realizar los respectivos *commits* en cada literal.**

Diego, Carlos, Álvaro y Fernando quieren volver a jugar *Dungeons and Dragons*, un reconocido juego de rol que se caracteriza por manejar un sistema de clases y puntuaciones propio, el cual ha sido utilizado por muchos videojuegos. Dado que han jugado por mucho tiempo, no quieren crear de manera manual sus personajes (especialmente porque usualmente quieren realizar los mismos personajes), por lo cual proponen escoger al azar un personaje entre las clases disponibles para jugar: Bárbaro, Bardo, Clérigo, Druida, Guerrero, Mago, Paladín y Pícaro.

Como solución, deciden crear un arreglo tridimensional que almacene algunos personajes generados al azar del cual se seleccionará un personaje para cada uno de ellos. Este vector tridimensional creará tres personajes de una cada de las ocho categorías, incluyendo sus seis puntuaciones de característica (Fuerza, Destreza, Constitución, Inteligencia, Sabiduría y Carisma) las cuales también se crearán aleatoriamente desde 10 hasta 18.

Una vez creado el arreglo y generados los personajes de cada clase, se escogerá aleatoriamente una categoría y se seleccionará al azar un personaje de los tres generados para dicha categoría.

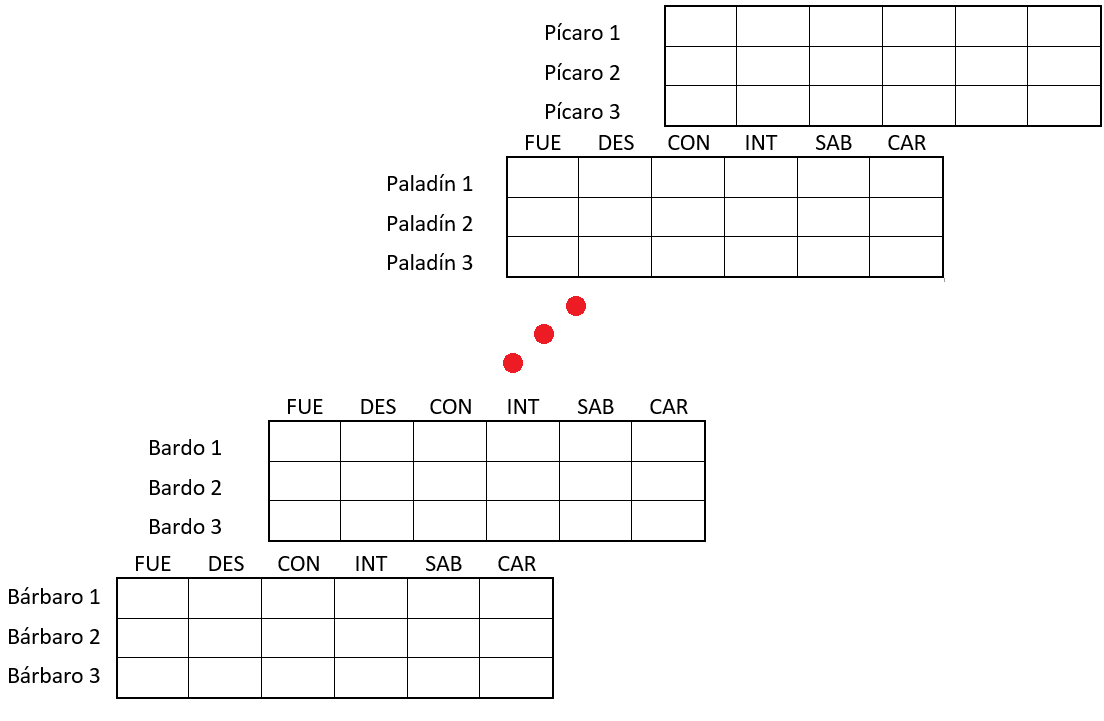


Imagen 1. Organización de personajes

1. Al llenar el arreglo tridimensional se sugiere que se creen los tres personajes para cada categoría en el orden: Bárbaros, Bardos, Clérigos, Druidas, Guerreros, Magos, Paladines y Pícaros, teniendo en cuenta que:

* Las mejores puntuaciones generadas se deberán agregar los atributos físicos (FUE, DES, CON) para el Bárbaro, Guerrero, Paladín y Pícaro.
* Las mejores puntuaciones generadas se deberán agregar los atributos mentales (INT, SAB, CAR) para el Bardo, Clérigo, Druida, Mago.

1. El algoritmo le pedirá al usuario un nombre de personaje y luego seleccionará del arreglo multidimensional previamente creado, una categoría al azar y posteriormente, seleccionará uno de los 3 personajes prediseñados. Al finalizar mostrará el nombre del personaje, la categoría generada y las puntuaciones de característica de cada atributo.

**NOTAS:**

**Las respuestas a las preguntas de cada punto deben ser escritas y subidas en un archivo de nombre *informe11\_NombresApellidos.docx.* Subir el archivo *informe11\_A\_NombreApellido.py* y *informe11\_B\_NombreApellido.py* al repositorio público de Github creado en el laboratorio 8 y anexar el enlace en el documento Word del informe entregado.**

**Deben adicionarse capturas de pantalla de los algoritmos y las respuestas de ambos planteamientos, de forma ordenada y explicada en el documento de Word.**

**RECURSOS ADICIONALES**

* Comparación de GitHub, GitLab y mención de algunas herramientas: <https://www.youtube.com/watch?v=EscDe0jG6XM>
* Git a fondo (lista de reproducción):

<https://www.youtube.com/watch?v=zH3I1DZNovk&list=PL9xYXqvLX2kMUrXTvDY6GI2hgacfy0rId>

* Aprendiendo Python (lista de reproducción): <https://www.youtube.com/watch?v=CjmzDHMHxwU&list=PLE549A038CF82905F>
* Aprendiendo JavaScript (lista de reproducción)

<https://www.youtube.com/watch?v=H5NdtjnO2Fk&list=PLpOqH6AE0tNhQ3KW1DPc18xZfLfjX5Xuz>

* Aprendiendo Ruby (lista de reproducción)

<https://www.youtube.com/watch?v=R3MhA8vWGsg&list=PLEFC2D43C36013A70>

* Aprendiendo Java (lista de reproducción)

<https://www.youtube.com/watch?v=Z0F7sJaOQtw&list=PL602060AB32FC864B>

* Aprendiendo C++ (lista de reproducción)

<https://www.youtube.com/watch?v=7x9369_-cWc&list=PL4A486BBFC5AD733B>

* Aprendiendo PHP

<https://www.youtube.com/watch?v=sEfcmo-THjM&list=PLF02659CF8418C229>

* Aprendiendo Kotlin (lista de reproducción)

<https://www.youtube.com/watch?v=CD9oIq7ti3s&list=PLh_neeN4BQCkH8lDUQ53tkFbRmLAV4g8P>

* Aprendiendo R (lista de reproducción)

<https://www.youtube.com/watch?v=TQJQ0fT5_rU&list=PLc2mD9gcCKBygAS07--5FhaGHw9UT8jVj&index=9>

* Aprendiendo Scala (lista de reproducción)

<https://www.youtube.com/watch?v=t6pl0gacuX0&list=PLTd5ehIj0goOlRPRfwGGu9Bc2ap4bt7wR>

* Aprendiendo C# (lista de reproducción)

<https://www.youtube.com/watch?v=pQ6Ezq72J5A&list=PLpOqH6AE0tNhmU9OUbm5FIJtnHAbJ4dKd>

* Aprendiendo Go

<https://www.youtube.com/watch?v=e37Q4SLC-30&list=PLdKnuzc4h6gFmPLeous4S0xn0j9Ik2s3Y>

* Aprendiendo Julia

<https://www.youtube.com/watch?v=LbTbs-0pOuc>

* Aprendiendo SWIFT

<https://www.youtube.com/watch?v=qAALd08Lf4c>