Universidad Nacional Mayor de San Marcos
Universidad del Perú, Decana de América

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
PROGRAMA: ESPECIALIZACION EN PYTHON

MÓDULO I

Reforzamiento N°04

Importante: Subir todas sus soluciones a su repositorio donde se encuentra las soluciones de los anteriores reforzamientos.

Diccionarios:

1. Crea correctamente un diccionario con los campos de: nombre, edad, salario y edad.

   Convierte tu diccionario finalmente a una lista y muestra el resultado en la terminal.

2. Agrega un nuevo key llamado "dni" con su respectivo valor y luego mostrar el valor del salario y DNI en consola. También elimina el key edad de tu diccionario, incluyendo su valor. Mostrar finalmente el diccionario actualizado.

3. Convertir tu diccionario a una lista y mostrar en consola el tipo de datos final que tienes.

4. Crear un diccionario con 6 departamentos del país.
   - Borrar cualquier departamento, usando la palabra reservada del.
   - Actualizar el penúltimo departamento por otro.

- Comprobar que no existe este departamento borrado dentro del diccionario.

5. Ingresar el nombre de tu carrera dentro de los valores que tienes en tu diccionario.
   - Mostrar en consola los valores de tu carrera y nombre agregándolos a una variable c/u

6. Ingresar por consola 4 números mediante consola, crear un diccionario donde los 'key' serán los números indicados y los valores serán los cubos de las estos keys. Mostrar finalmente este diccionario.

7. Realizar un programa donde se ingresarán por consola los nombres de los alumnos (indicar previamente la cantidad de alumnos a ingresar) de un curso y las notas de c/u. Guardarás la información en un diccionario donde las claves serán los nombres de c/u de estos alumnos y sus valores serán las notas de esto alumnos.
Finalmente mostrarás los alumnos con sus notas en un mensaje similar a "Pedro tiene la nota de 15" y también la media de todas las notas.

8. Crear una agenda basada en un diccionario donde los key serán los nombres de las personas y sus "values" serán los números de teléfono de c/u.

Ingresarás por consola el nombre y el número de cada persona que serán registrados en la agenda.

El programa también te permitirá buscar por nombre en el diccionario en caso no exista mostrar un mensaje de "No se encuentra registrado en la agenda"

9. Una empresa desea gestionar las facturas pendientes que tiene por pagar, para esto se creará un diccionario donde tendrá por key el número de factura "00054" y su value será el coste de la factura. El programa tendrá la opción de pedir nueva factura (por consola) que se agregará al diccionario. Cada vez que el área de contabilidad pague una factura se pedirá el número de factura que fue cancelada, si existe mostrar un mensaje donde indicará "La factura ya está cancelada" caso contrario "El número de factura no existe"

Considerar que cada vez que se realice algún pago o ingreso de una nueva factura se mostrará inmediatamente al diccionario actualizado.

Strings:

1. Dada una frase u oración encontrar que palabra es la que tiene más caracteres y cuántos caracteres tiene

Input: "La programación en Python es poderosa"

Output: "programación" – 12 caracteres

2. Crear un programa que cuente cuántas veces aparece cada vocal en la oración. Ignorar mayúsculas/minúsculas

Input: "Programación en Python"

Output:

a: 2

e: 1

i: 1

o: 2

u: 0

Métodos útiles: lower() y count()

3. Pide al usuario que ingrese una frase y una palabra, obtén si la palabra está dentro de la oración sin importar si está en mayúsculas o minúsculas.
   En caso que aparezca, indica la posición del primer carácter en donde empieza

Input: frase = "Python y sus enormes ventajas", palabra = "Python"

Output: "Python aparece en la posición 0"

Métodos útiles: lower() y find()

Funciones:

1. Pedir dos números positivos mediante terminal al usuario. Mostrar como salida el número cuya sumatoria de dígitos es el mayor y los números cuya sumatoria de dígitos es menor que 10. Utilizar una o más funciones, según sea conveniente.

2. Crea una función que al ingresar dos números por parámetro mostrará todos los cuadrados de los números que hay entre ellos (Usar la función una vez y mostrar el resultado por consola). Los números serán ingresados y solicitados al usuario.

3. Crear una función que sume los dígitos del número ingresado y muestre por consola la suma de todos estos dígitos.

4. Pedir al usuario que ingrese un **nombre y apellidos** el cual será usada por un parámetro para una función que se creará e indicará cuantas letras tiene el nombre solamente. Usar la función un mínimo de dos veces para dos personas e indicar quien tiene el nombre con mayor número de caracteres (condicional)

5. Crear una función que aceptará por parámetro dos valores que serán ingresados por el usuario, una lista donde los valores serán llenados por el usuario también y un segundo parámetro que eliminará de la lista que fue ingresada a la función, finalmente el **output** de la función será la lista actualizada sin el valor que se sacará de la lista. Mostrar también la lista original y el número que fue eliminado.

Excepciones:

1. Crear una función llamada **division_segura(a, b)** que recibirá dos números e intentará devolver la división de a entre b

Si b es cero, debe capturar la excepción y mostrar mensaje "Error: no puedes dividir entre cero"

Si ambos valores son válidos deben imprimir el resultado Independientemente del resultado, debe imprimir "Operación finalizada" al final

- Usar try, except, finally

- Valida que los datos ingresados sean numéricos de lo contrario mostrar "Error: Entrada no numérica"

- Usarás la función al menos 3 veces incluyendo un caso de error

2. Crear una función y dentro la respectiva excepción o excepciones para el siguiente bloque de código para que tu programa no se bloquee, ya que solo aceptará datos tipos entero y además imprimir un mensaje al usuario la causa y/o solución. También debe indicar el índice donde ingresarás este nuevo dato, si el índice está fuera de rango indicárselo al usuario:

lista = [2, 6, 10, 4, 5, 23, 1]

lista[10]: No es posible ingresar el dato, índice fuera de rango

- Usarás dos tipos diferentes de excepciones (IndexError TypeError) y

- Usarás la función al menos 3 veces incluyendo un caso de error

3. Crear una función funciona_indice(persona, indice) y dentro la respectiva excepción para el siguiente bloque de código para que tu programa no se bloquee y además imprime un mensaje al usuario: "El índice "nombre_indice" ingresado no existe en el diccionario", tipo de datos, etc que más pueden incurrir para este caso (los datos se pedirán por consola):

```
persona= {'nombre':'Xavier', 'apellido':'Rodriguez', 'dni':'63325345'}
persona['profesion'] #El índice en este caso no existe
```

Usar la función al menos 2 veces incluyendo un caso de error

4. Crear una función saluda_fecha(nombre, dia, mes, anio) que contendrá una excepción para el siguiente bloque de código y tu programa no se bloquee. Además, imprime un mensaje al usuario la causa y/o solución (Pedir nombre, día, mes, año por consola):

Nombre: No debe recibir un dato numérico, sino decírselo al usuario Día, mes y año: No debe recibir un dato string, sino decírselo al usuario

```
cadena = "Hello NOMBRE, hoy estamos DÍA de MES del AÑO"

# Hello Leonardo, hoy estamos 04 de agosto del 2025
```

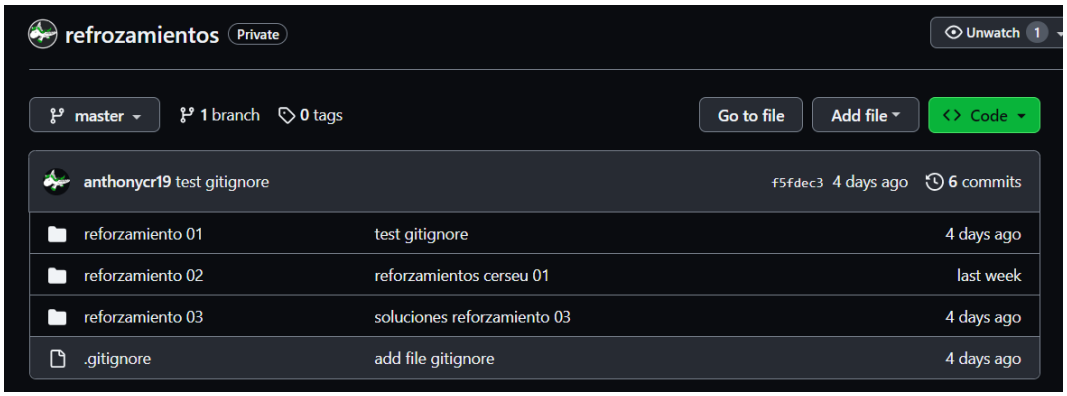
Independientemente del resultado, debe imprimir "Operación finalizada" al final

- Usar try, except, finally

Usar la función al menos 3 veces, incluyendo casos de error

Indicaciones:

a. Subir su siguiente carpeta de "reforzamientos 04" a su repositorio con el nombre de repositorio de reforzamientos y la estructura final en GitHub debería verse del siguiente modo:



- Cada problema debe tener la descripción de lo solicitado como comentario en la parte superior de sus soluciones.
- Cada problema será resuelto en un archivo .py distinto y todos comprimidos en un solo file con el nombre de: nombre apellido - reforzamiento-04.zip o .rar
- **Correo** a enviar soluciones y link de repositorio:

  docente.cerseu.unmsm@gmail.com
- **Asunto**: Reforzamiento 04 – Módulo I
- Fecha máxima de entrega: sábado 27 de set. hasta las 23:59 horas.