Curso Programación con Python Nivel 1

Tarea #1

Monserrath Aguilar Driggs

Parte I: Git

https://github.com/monsedriggs/python1\_mon.git

Parte II: Resolución de problemas

**Problema 1:**

*Paso 1.*

*Paso 2.*

Se necesita una interfaz de usuario, para poder pedir la entrada. Los inputs van a ser strings ingresados por el usuario. Los outputs son “si” o “no”. Los pasos necesarios para llegar al resultado son verificar que cada unidad de letra coincida de la siguiente manera: primera letra==ultima letra, segunda letra==penúltima letra, y así hasta completar las letras de la palabra.

El usuario ingresa una frase

Guardar la entrada en una variable llamada “frase”

Iniciar un contador “x” en cero (0)

Nombrar una variable “y”, con la longitud de la frase -1

Hasta que “x” sea igual a la mitad de “y”:

Recorrer cada posición de la hilera de derecha a izquierda, y de izquierda a derecha, si las letras son iguales se aumenta el contador “x” (x+=1) y la variable “y” (y-=1)

Si todas las letras coinciden, es verdadero imprime “si”

Si no cumple la condición, imprime “no”

*Paso 3.*

Crear una interfaz que pida al usuario ingresar la frase. Esa entrada queda guardada en la variable “frase”

Contador x=0.

Variable “y” para obtener la longitud de la frase y (-1) para poder recorrer las letras ya que inician en 0.

Asociar las posiciones de letras, la primera con la última y verificar si son iguales. Si es correcto se aumenta la “x” y se disminuye la “y” para ir con el siguiente par y verificar la igualdad. Esto hasta que “x” sea a la mitad de “y”

Se debe considerar que dentro de la entrada hay espacios “ ”

Si todas las parejas de letras son iguales, el programa debe imprimir “si”.

De lo contrario, si alguna pareja no es igual el programa imprime “no”.

**Problema 2:**

*Paso 1.*

*Paso 2.*

No se necesita un interfaz ya que se obtienen los datos de un servidor. El programa recibe logs. El output seria “si” si se debe enviar un correo al administrador. Para esto el programa debe borrar todos los logs, a excepción de los que contienen “error”, luego estos se agregan a un directorio para enviarlos.

Se obtienen los datos de un servidor y se guardan en una variable llamado “logs”

Recorrer cada log y revisar si contiene la palabra “error”

Si es verdadero, lo guarda en una variable llamada “Errores” e imprime “si”

Si no lo cumplen imprime “no”

*Paso 3.*

Cargar el archivo del servidor donde se obtienen los logs. Guardarlos en “logs”.

Por día, acceder a cada log en el archivo (variable “logs”) iterando. Busca la palabra “error” en cada log.

Si es correcto, guarda logs de error en variable “Errores”. Imprimir “si”

Si los logs no son errores el programa imprime “no”.

**Problema 3:**

*Paso 1.*

*Paso 2.*

Necesita una interfaz para que el usuario haga una entrada. El input es un número entero que ingresará el usuario. El output seria “si” o “no”. Para llegar a esto el programa necesita identificar si en el numero ingresado hay un 0 o un 5, lo que indica que si podría ser reordenado a un múltiplo de 5.

Usuario ingresa un número

Guardar el número ingresado en una lista llamada “digitos”

Iniciar un contador “x” en cero (0)

Nombrar una variable “y” para la longitud de la lista

Mientras que “x” sea menor o igual a y:

Revisa si cada digito es igual a 0 o a 5. Y aumenta el contador x+=1

Si algún digito lo cumple imprime “si”

Si no se cumple imprime “no”

*Paso 3.*

Pedir al usuario un número entero. La entrada se guarda en variable “num”.

Guardar número en una lista, para separar sus dígitos

Contador x=0

Variable “y” para obtener la longitud del número, y saber hasta cuando hacer la revisión.

Verificar que digito==0 or digito==5. Lo que indica que podría ser un múltiplo de 5.

Si existe al menos un 0 o un 5, el programa imprime “si”.

De lo contrario, imprime “no”.