**Universidad Nacional Autónoma de México**

**Facultad de Estudios Superiores Aragón**

**Ingeniería en Computación**

**\* Nombre del alumno: Pimentel Reyes José Luis.**

**\* Nombre del profesor: Marcelo Perez Medel.**

**\* Nombre de la materia: Diseño y Análisis de Algoritmos.**

**\* Nombre de la actividad: “*Tarea 5: Acertijo de Einstein por fuerza bruta*”.**

**\* Fecha de entrega: 31 de octubre del 2023.**

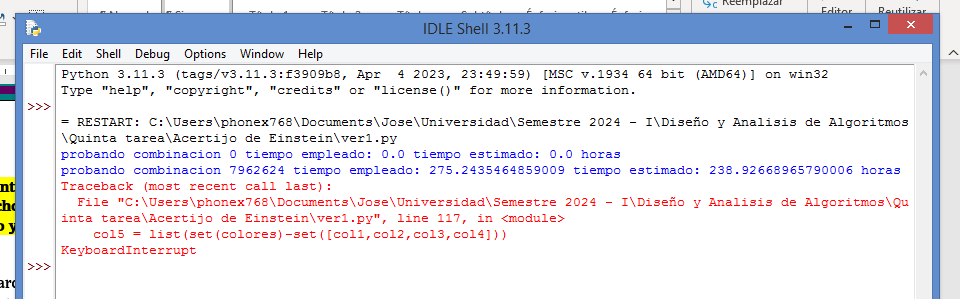
**\* Grupo: 1507.**

**\* Semestre: 2024 - I.**

***Tarea***

**a) Aquí se encuentra el programa que resuelve el *acertijo de Eisntein* por fuerza bruta, tarda mucho. Ejecútelo, después de unos instantes aparecerá el tiempo estimado, anótelo y deténgalo presionando las teclas ctrl-C.**

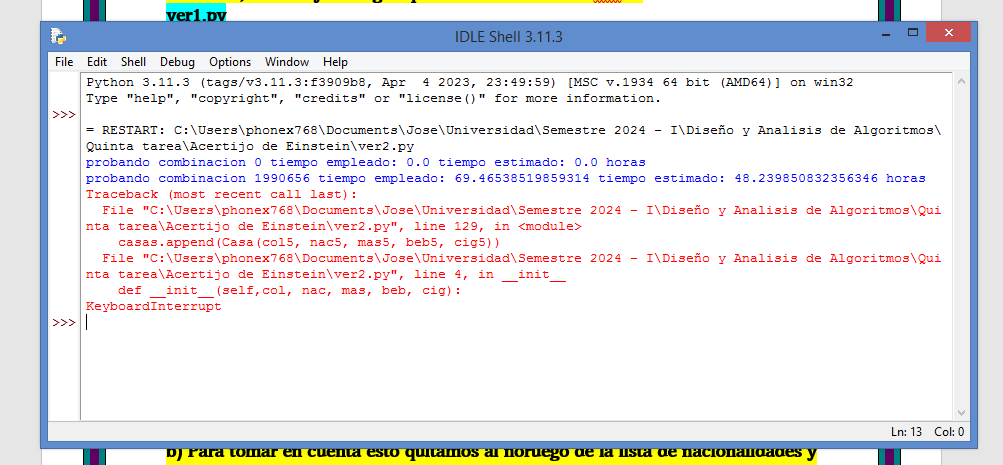
**ver1.py**



**El tiempo que tarda por fuerza bruta es mucho, algunos días, pero tenemos algunas pistas, por ejemplo sabemos que el noruego vive en la primera casa, así que podemos eliminar todas las miles de combinaciones donde el noruego vive en otras casas.**

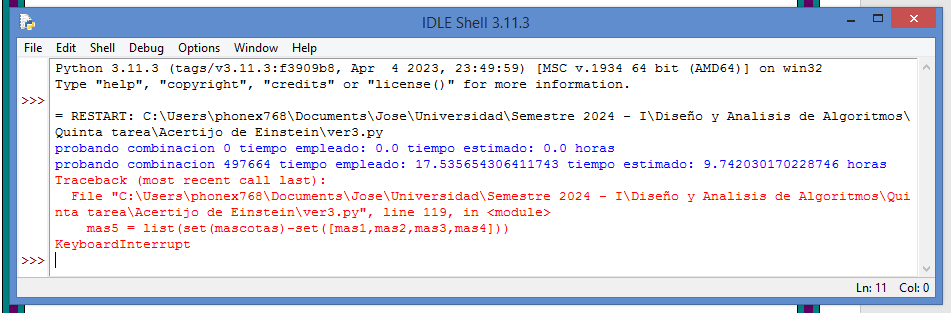
**b) Para tomar en cuenta esto quitamos al noruego de la lista de nacionalidades y lo asignamos a la primera casa, el programa modificado queda de la siguiente forma: (ejecútelo y tome nota del tiempo que tardaría en terminar).**

**ver2.py**



**c) Aun así el tiempo es mucho tiempo, si tomamos en cuenta que la bebida de la casa del centro es leche podemos modificar el programa de la siguiente forma:**

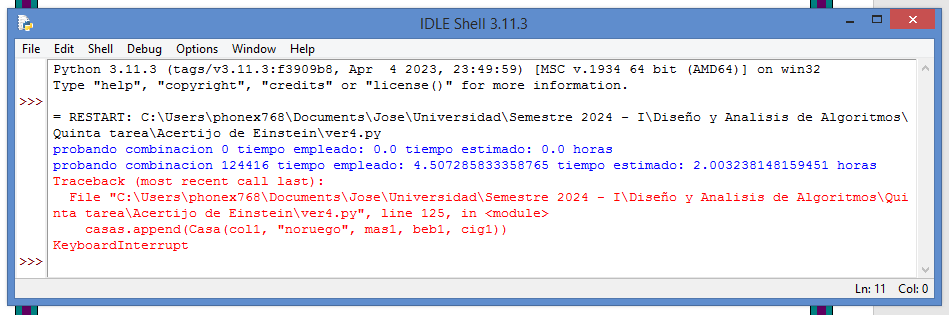
**ver3.py**



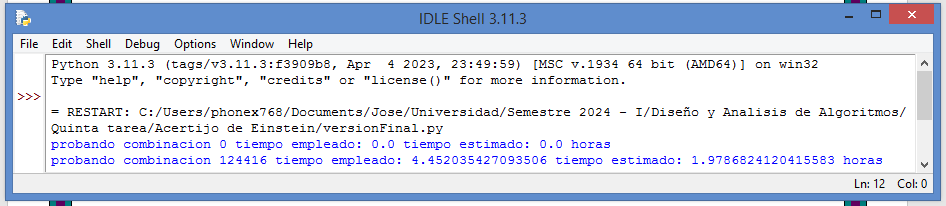
**d) Por último sabemos que la segunda casa es de color azul:**

**import time**

**ver4.py**



**e) Modifique el programa para que cuando encuentre la solución la imprima, tome el tiempo de esta última versión.**



***Código***

import time

class Casa:

def \_\_init\_\_(self,col, nac, mas, beb, cig):

self.color = col

self.nacion = nac

self.mascota = mas

self.bebida = beb

self.cigarro = cig

def checaCondiciones(casas):

c1 = False; c2 = False; c3 = False

c4 = False; c5 = False; c6 = False

c7 = False; c8 = False; c9 = False

c10 = False; c11 = False; c12 = False

c13 = False; c14 = False; c15 = False

for casa in casas:

if (casa.nacion == "britanico"):

if (casa.color=="roja"):

c1 = True

for casa in casas:

if (casa.nacion == "sueco"):

if (casa.mascota=="perro"):

c2 = True

for casa in casas:

if (casa.nacion == "danes"):

if (casa.bebida=="te"):

c3 = True

for casa in casas:

if (casa.color == "verde"):

if (casa.bebida=="cafe"):

c5 = True

for casa in casas:

if (casa.cigarro == "Pall Mall"):

if (casa.mascota=="pajaro"):

c6 = True

for casa in casas:

if (casa.color == "amarilla"):

if (casa.cigarro=="Dunhill"):

c7 = True

for casa in casas:

if (casa.cigarro == "Bluemasters"):

if (casa.bebida=="cerveza"):

c12 = True

for casa in casas:

if (casa.nacion == "aleman"):

if (casa.cigarro=="prince"):

c13 = True

if (c1 and c2 and c3 and c5 and c6 and c7 and c12 and c13):

print ("solución encontrada!!!")

colores =["roja","blanca","amarilla","verde"]

naciones =["aleman","ingles", "danes","britanico"]

mascotas = ["pez", "caballo", "perro", "gato", "pajaro"]

bebidas =["te", "cafe", "cerveza", "agua"]

cigarros=["Pall Mall", "Dunhill", "Brends", "Bluemasters", "prince"]

fc = len(colores)

fn = len(naciones)

fm = len(mascotas)

fb = len(bebidas)

fcig = len(cigarros)

factor = fc\*fn\*fm\*fb\*fcig

prueba = 0

vuelta = 0

inicio = time.time()

for col1 in colores:

for nac1 in naciones:

for mas1 in mascotas:

for beb1 in bebidas:

for cig1 in cigarros:

fin = time.time()

if (vuelta==0):

print ("probando combinacion "+str(prueba)+" tiempo empleado: "+str(fin-inicio)+" tiempo estimado: "

+str((fin-inicio)\*factor/3600/(vuelta+1))+" horas")

else:

print ("probando combinacion "+str(prueba)+" tiempo empleado: "+str(fin-inicio)+" tiempo estimado: "

+str((fin-inicio)\*factor/3600/(vuelta))+" horas")

vuelta = vuelta + 1

## Crear la segunda casa para esta combinacion

colores2 = list(set(colores)-set([col1]))

naciones2 = list(set(naciones)-set([nac1]))

mascotas2 = list(set(mascotas)-set([mas1]))

bebidas2 = list(set(bebidas)-set([beb1]))

cigarros2 = list(set(cigarros)-set([cig1]))

for col2 in colores2:

for nac2 in naciones2:

for mas2 in mascotas2:

for beb2 in bebidas2:

for cig2 in cigarros2:

## Crear la 3era casa

colores3 = list(set(colores)-set([col1,col2]))

naciones3 = list(set(naciones)-set([nac1,nac2]))

mascotas3 = list(set(mascotas)-set([mas1,mas2]))

bebidas3 = list(set(bebidas)-set([beb1,beb2]))

cigarros3 = list(set(cigarros)-set([cig1,cig2]))

for col3 in colores3:

for nac3 in naciones3:

for mas3 in mascotas3:

for beb3 in bebidas3:

for cig3 in cigarros3:

## Crear la 4ta casa

colores4 = list(set(colores)-set([col1,col2,col3]))

naciones4 = list(set(naciones)-set([nac1,nac2,nac3]))

mascotas4 = list(set(mascotas)-set([mas1,mas2,mas3]))

bebidas4 = list(set(bebidas)-set([beb1,beb2,beb3]))

cigarros4 = list(set(cigarros)-set([cig1,cig2,cig3]))

for col4 in colores4:

for nac4 in naciones4:

for mas4 in mascotas4:

for beb4 in bebidas4:

for cig4 in cigarros4:

## Determinar la 5ta casa

col5 = list(set(colores)-set([col1,col2,col3,col4]))

nac5 = list(set(naciones)-set([nac1,nac2,nac3,nac4]))

mas5 = list(set(mascotas)-set([mas1,mas2,mas3,mas4]))

beb5 = list(set(bebidas)-set([beb1,beb2,beb3,beb4]))

cig5 = list(set(cigarros)-set([cig1,cig2,cig3,cig4]))

#Creamos cada combinacion valida

prueba = prueba + 1

casas=[]

casas.append(Casa(col1, "noruego", mas1, beb1, cig1))

casas.append(Casa("azul", nac2, mas2, beb2, cig2))

casas.append(Casa(col3, nac3, mas3, "leche", cig3))

casas.append(Casa(col4, nac4, mas4, beb4, cig4))

casas.append(Casa(col5, nac5, mas5, beb5, cig5))

checaCondiciones(casas)

if (c1 and c2 and c3 and c5 and c6 and c7 and c12 and c13):

fin = time.time()

print(f"Tiempo empleado: {fin - inicio} segundos")

else:

print("No se encontró ninguna solución.")