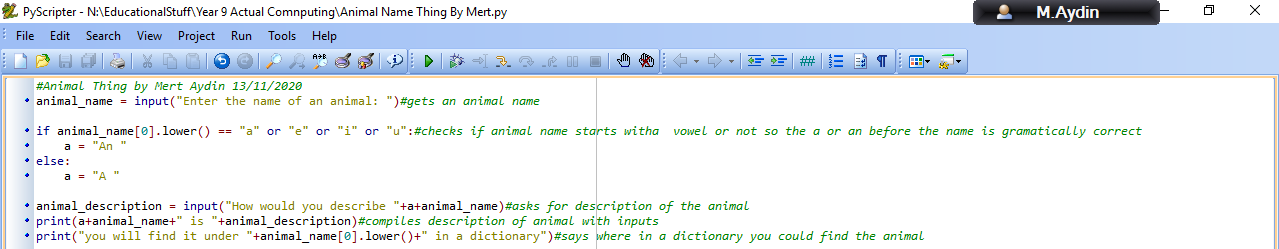
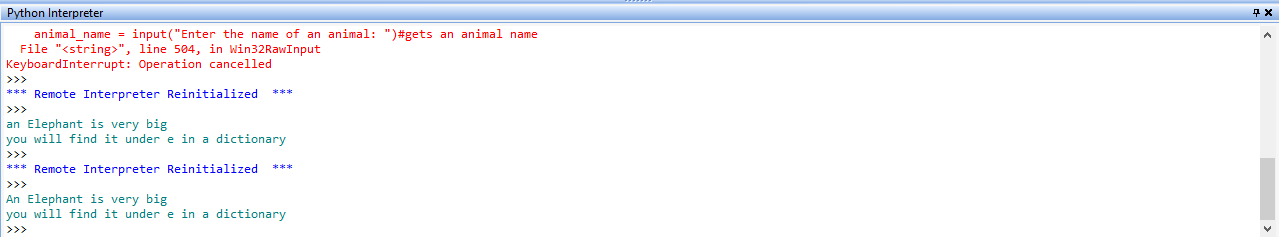
**Mert Evidence File**

Animal Name Program By Mert 13-11-2020



****

Check comments below:

#Animal Thing by Mert Aydin 13/11/2020

#gets an animal name

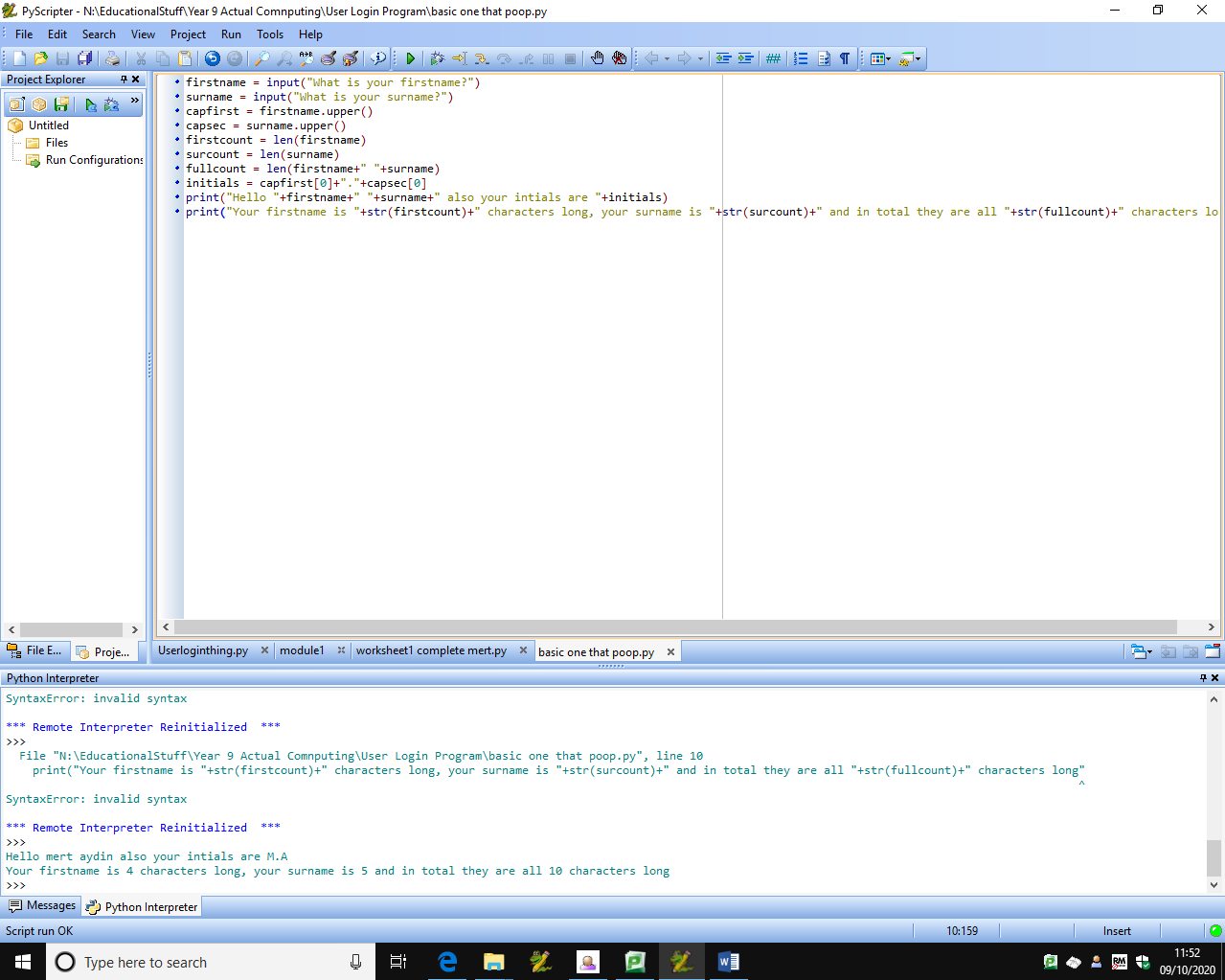
#checks if animal name starts with a vowel or not so the a or an before the name is gramatically correct

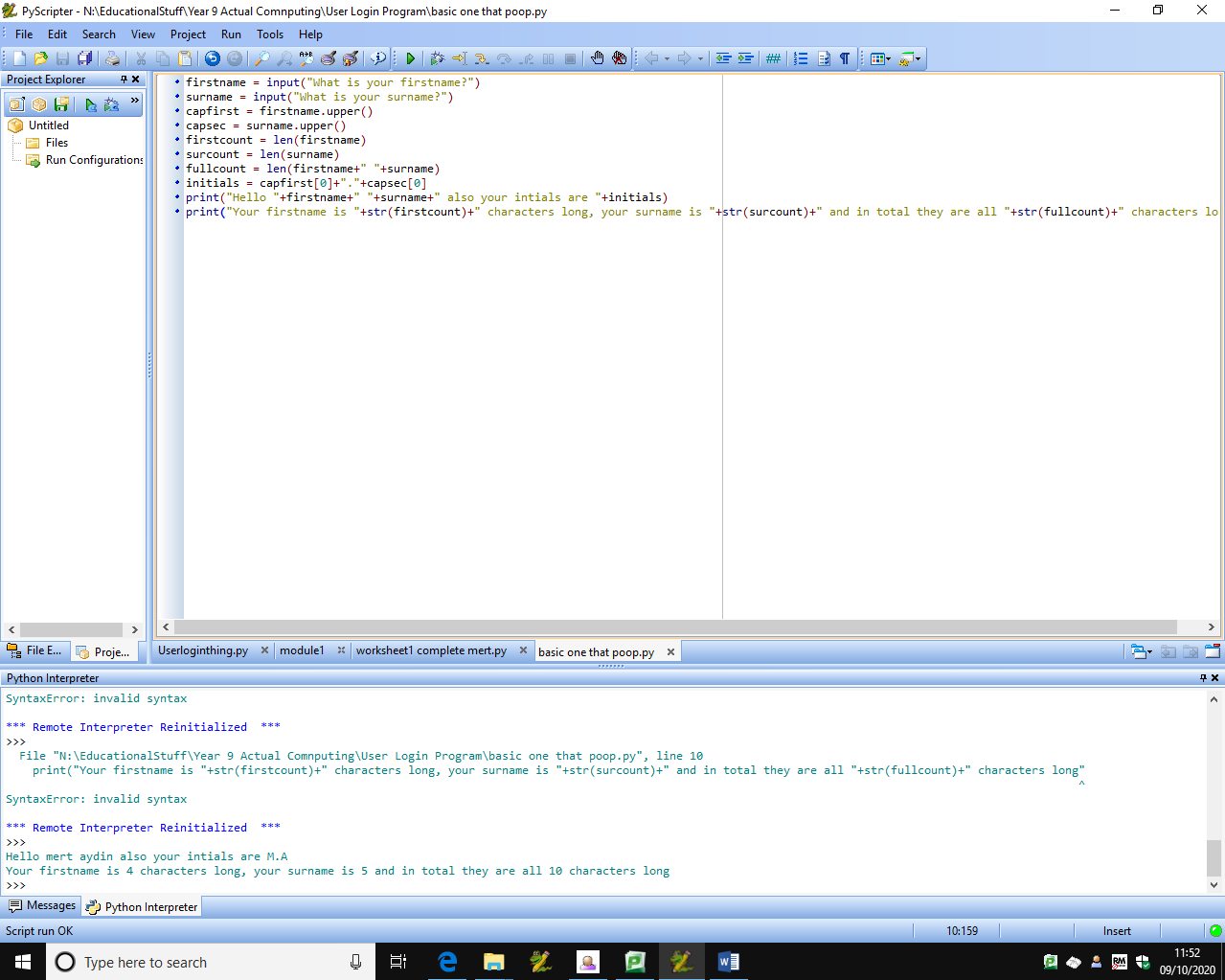
#asks for description of the animal

#compiles description of animal with inputs

#says where in a dictionary you could find the animal

Basic User Login Program Mert



****

Check below for comments:

#Gets user’s firstname

#Gets user’s surname

#Capitalises and puts firstname in variable

#Capitalises and puts surname in variable

#Counts firstname length and puts it in a variable

#Counts surname length and puts it in a variable

#Counts Length of the user’s full name and puts it in a variable

#Constructs user’s initials from given inputs

#Displays inputs given via terminal text, specifically the inputted firtsname and surname and the constructed initials

#Displays the length of some of the inputs given and constructed strings from the input’s length

Constants And Variables Demo Mert

#these are constants

CONST\_1=2

CONST\_2=42

CONST\_3=25.13/8

var\_1="Hello"

var\_2="Bye"

var\_3=0

print("pi is roughly "+str(CONST\_3))

while var\_3 != CONST\_2:

var\_3=var\_3+CONST\_1

print(var\_1)

print(str(var\_3))

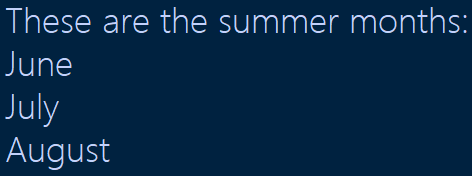
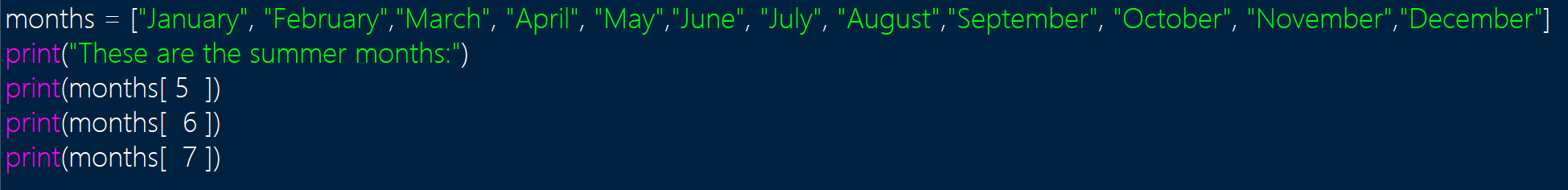
print(var\_2)

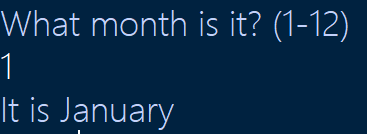
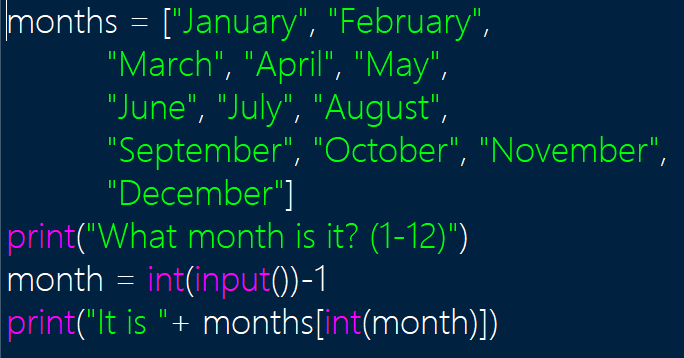
var\_2="var\_1"

print(var\_2)

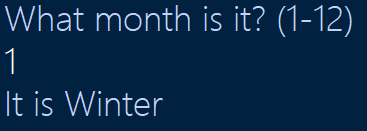
Date Program Things Mert

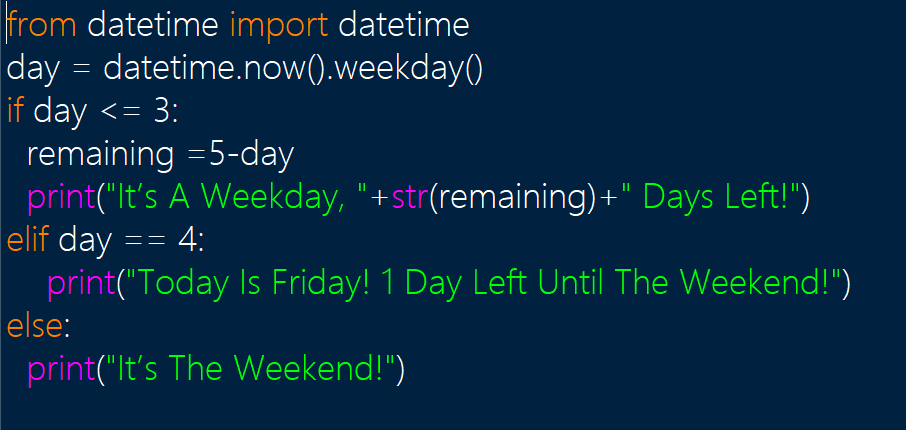
Note:These are all from lockdown

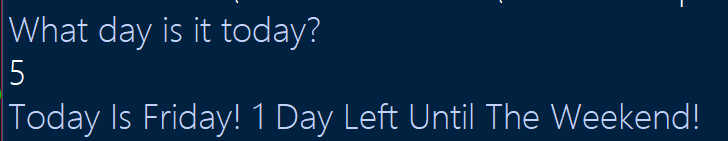
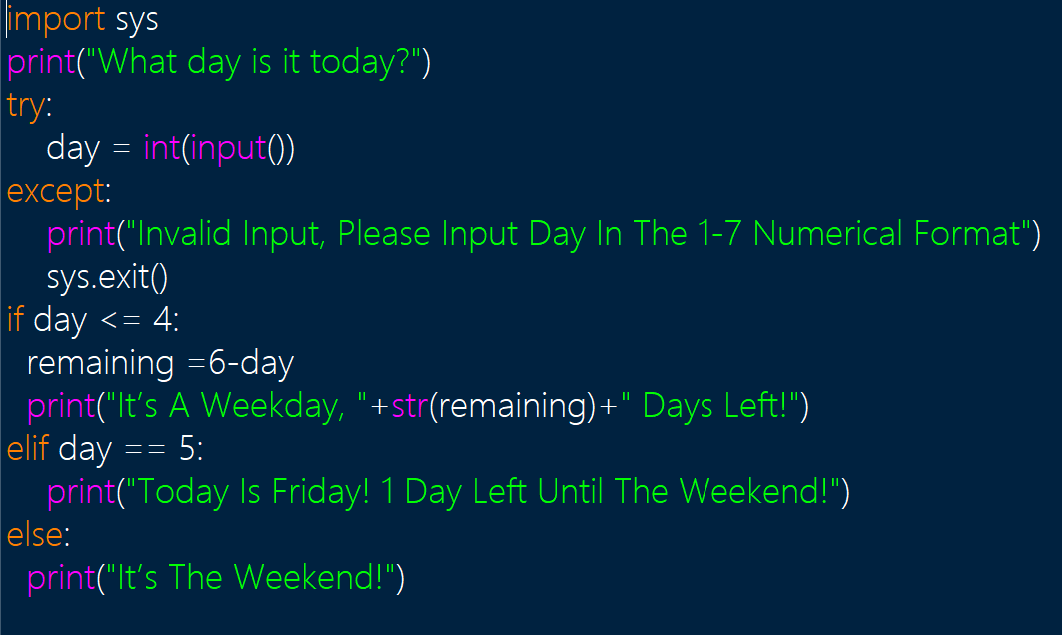




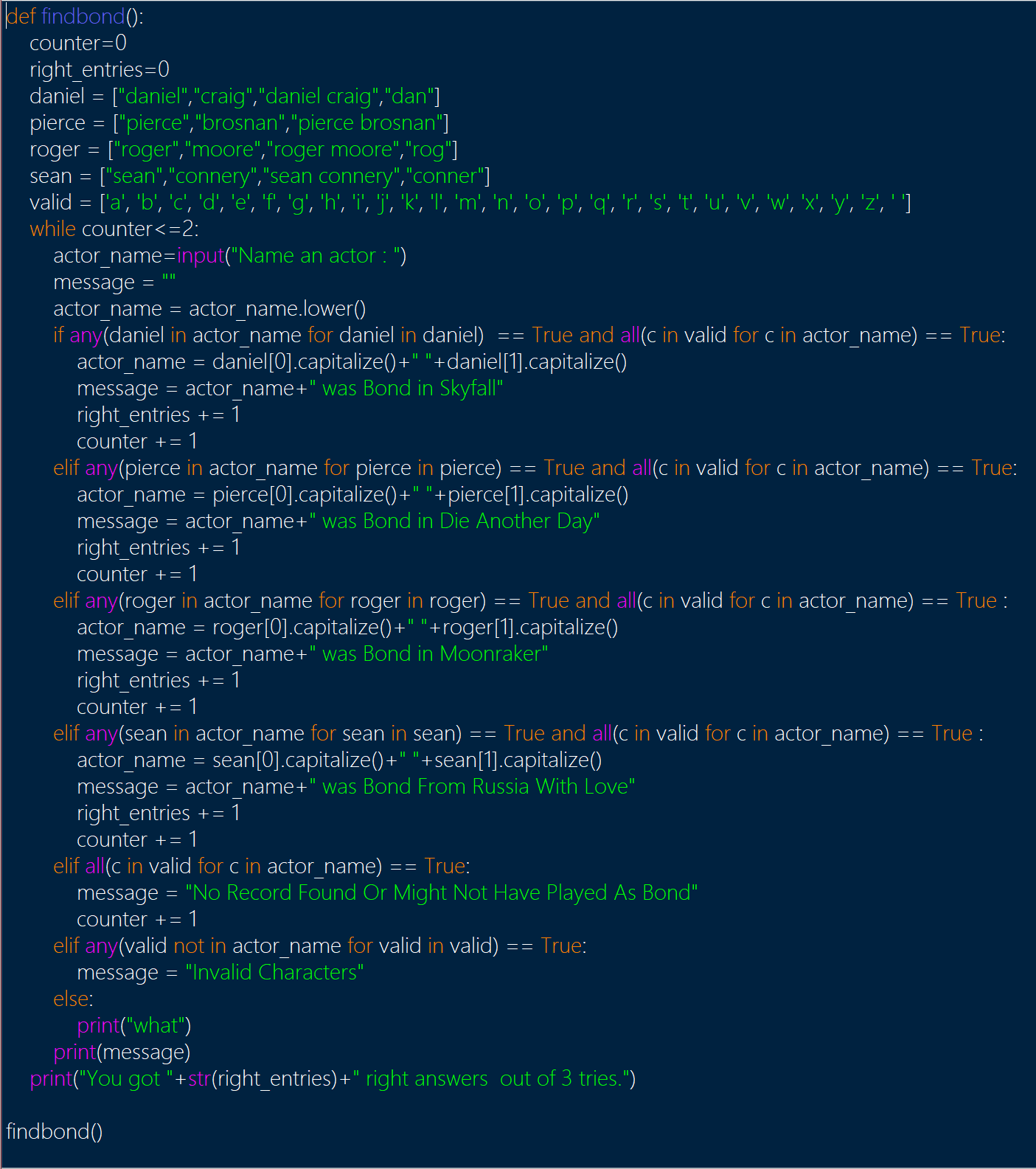








Mert 18/12/2020



code:

def findbond():#starts function

counter=0#resets counter

right\_entries=0#resets right entires

daniel = ["daniel","craig","daniel craig","dan"]#these define right names

pierce = ["pierce","brosnan","pierce brosnan"]

roger = ["roger","moore","roger moore","rog"]

sean = ["sean","connery","sean connery","conner"]

valid = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', ' ']#defines valid characters

while counter<=2:#asks for input while counter is not 4 or above

actor\_name=input("Name an actor : ")

message = ""

actor\_name = actor\_name.lower()

if any(daniel in actor\_name for daniel in daniel) == True and all(c in valid for c in actor\_name) == True:#all these check if the input has one of the names from the lists and they do not contain invalid characters, if so they add to the right entries

actor\_name = daniel[0].capitalize()+" "+daniel[1].capitalize()

message = actor\_name+" was Bond in Skyfall"#defines message

right\_entries += 1#adds to right entries

counter += 1#adds to counter to eventually end loop

elif any(pierce in actor\_name for pierce in pierce) == True and all(c in valid for c in actor\_name) == True:

actor\_name = pierce[0].capitalize()+" "+pierce[1].capitalize()

message = actor\_name+" was Bond in Die Another Day"

right\_entries += 1

counter += 1

elif any(roger in actor\_name for roger in roger) == True and all(c in valid for c in actor\_name) == True :

actor\_name = roger[0].capitalize()+" "+roger[1].capitalize()

message = actor\_name+" was Bond in Moonraker"

right\_entries += 1

counter += 1

elif any(sean in actor\_name for sean in sean) == True and all(c in valid for c in actor\_name) == True :

actor\_name = sean[0].capitalize()+" "+sean[1].capitalize()

message = actor\_name+" was Bond From Russia With Love"

right\_entries += 1

counter += 1

elif all(c in valid for c in actor\_name) == True:#checks if valid characters are in there

message = "No Record Found Or Might Not Have Played As Bond"

counter += 1

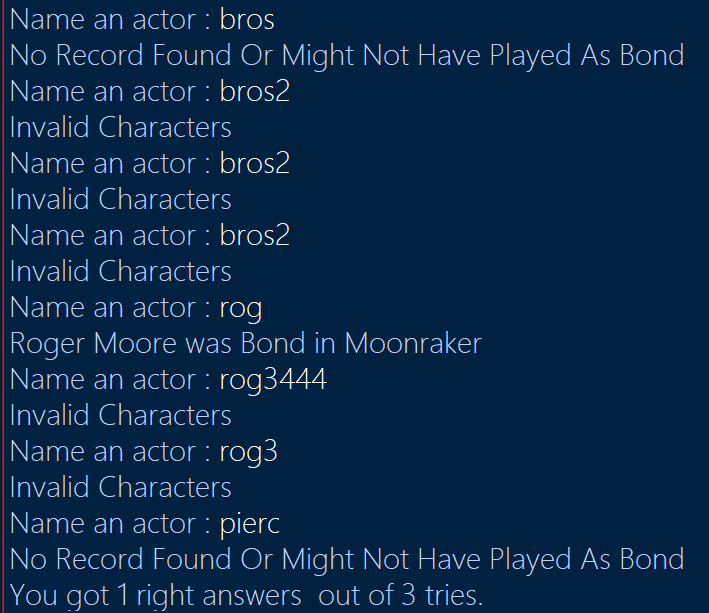
elif any(valid not in actor\_name for valid in valid) == True:#checks if characters are invalid if all other checks fail

message = "Invalid Characters"

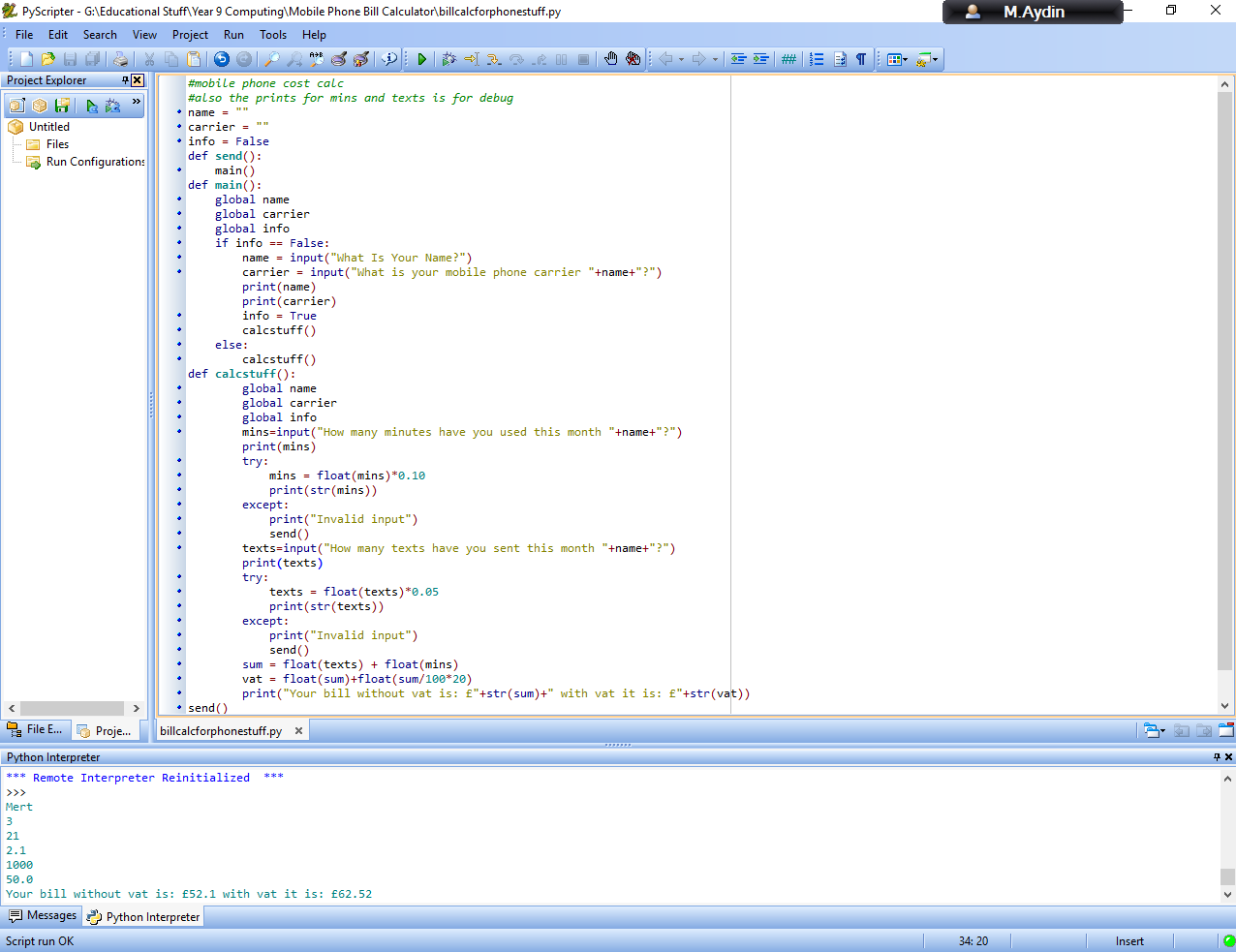
else:#this is just in case some weird thing happens

print("what")

print(message)#prints score



Mert Aydin 27-11-2020 Mobile Phone Bill Calculator



Code:

#mobile phone cost calc

#also the prints for mins and texts is for debug

name = ""

carrier = ""

info = False

def send():

main()

def main():

global name

global carrier

global info

if info == False:

name = input("What Is Your Name?")

carrier = input("What is your mobile phone carrier "+name+"?")

print(name)

print(carrier)

info = True

calcstuff()

else:

calcstuff()

def calcstuff():

global name

global carrier

global info

mins=input("How many minutes have you used this month "+name+"?")

print(mins)

try:

mins = float(mins)\*0.10

print(str(mins))

except:

print("Invalid input")

send()

texts=input("How many texts have you sent this month "+name+"?")

print(texts)

try:

texts = float(texts)\*0.05

print(str(texts))

except:

print("Invalid input")

send()

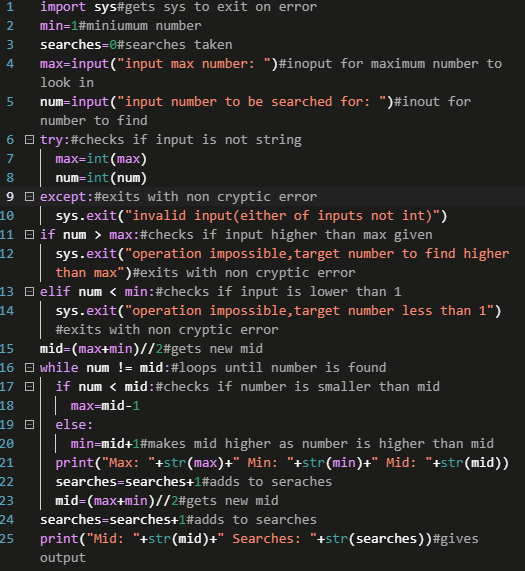
sum = float(texts) + float(mins)

vat = float(sum)+float(sum/100\*20)

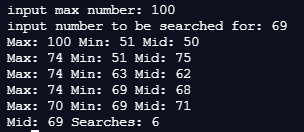
print("Your bill without vat is: £"+str(sum)+" with vat it is: £"+str(vat))

send()

Mert 7/5/2021 Binary Search Thing



Output:



Code:

import sys#gets sys to exit on error

min=1#miniumum number

searches=0#searches taken

max=input("input max number: ")#inoput for maximum number to look in

num=input("input number to be searched for: ")#inout for number to find

try:#checks if input is not string

max=int(max)

num=int(num)

except:#exits with non cryptic error

sys.exit("invalid input(either of inputs not int)")

if num > max:#checks if input higher than max given

sys.exit("operation impossible,target number to find higher than max")#exits with non cryptic error

elif num < min:#checks if input is lower than 1

sys.exit("operation impossible,target number less than 1")#exits with non cryptic error

mid=(max+min)//2#gets new mid

while num != mid:#loops until number is found

if num < mid:#checks if number is smaller than mid

max=mid-1

else:

min=mid+1#makes mid higher as number is higher than mid

print("Max: "+str(max)+" Min: "+str(min)+" Mid: "+str(mid))

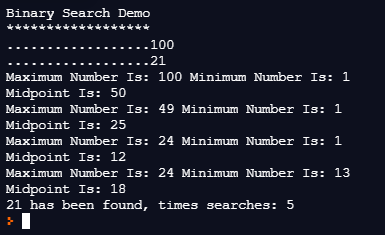
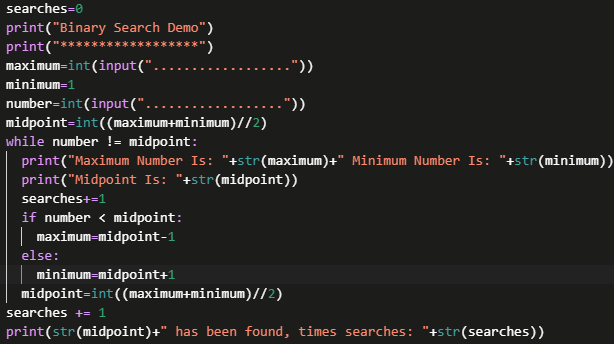
searches=searches+1#adds to seraches

mid=(max+min)//2#gets new mid

searches=searches+1#adds to searches

print("Mid: "+str(mid)+" Searches: "+str(searches))#gives output

Binary Search Correction 14/5/2021



Code:

searches=0

print("Binary Search Demo")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

maximum=int(input(".................."))

minimum=1

number=int(input(".................."))

midpoint=int((maximum+minimum)//2)

while number != midpoint:

print("Maximum Number Is: "+str(maximum)+" Minimum Number Is: "+str(minimum))

print("Midpoint Is: "+str(midpoint))

searches+=1

if number < midpoint:

maximum=midpoint-1

else:

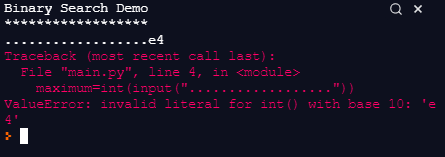
minimum=midpoint+1

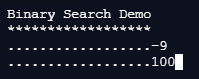
midpoint=int((maximum+minimum)//2)

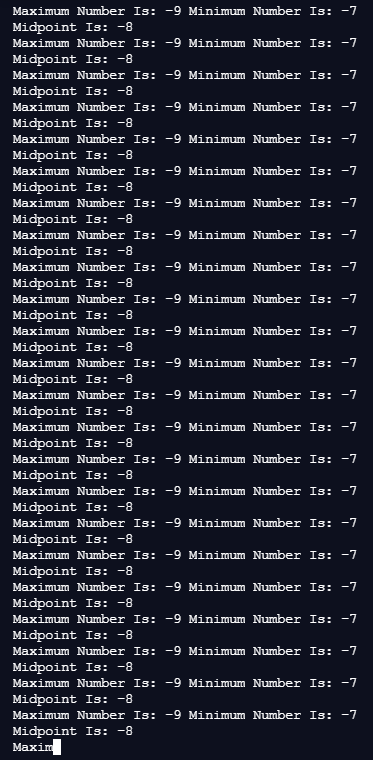
searches += 1

print(str(midpoint)+" has been found, times searches: "+str(searches))

more testing:







Note: This happens with this version of the binary search, but the other one I made is far more robust than this as it has checks for anything that will cause logic and runtime errors