#question1:

A:

l1=['HTTP','HTTPS','FTP','DNS'] القائمة الأولى

l2=[80,443,21,53] القائمة الثانية

d=dict(zip(l1,l2)) استخدمنا التابع للدمج بين معلومات القائمتين zip

print(d)



B:

x=int(input()) #input number ندخل عدد صحيح اكبر من صفر

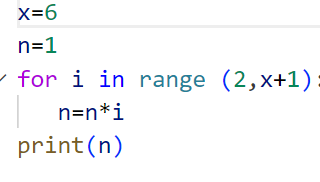
n=1 عرفنا حيادي الضرب

for i in range (2,x+1):

   n=n\*I ضربنا الحيادي بالعدد

print(n)

ex:

output:



C:

l=['Network','Bio','Programming','Physics','Music']

x=[item for item in l if item.startswith('B')]

for item in x:

    print(item)

output:



D:

x={i:i+1 for i in range (11)} عرفنا متحول يبدأ من صفر و ينتهي ب 11

print(x)

output:



#question2:

p=int(input())

i=result=0

while p!=0:

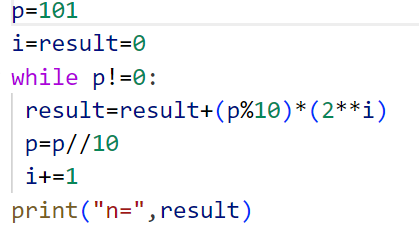
 result=result+(p%10)\*(2\*\*i)

 p=p//10

 i+=1

print("n=",result)

output:





#question3:

import json

# Initialize variables

quiz = []

scores = 0

number = 1

# Load questions from file

with open("quiz.txt", 'r') as f:

    quiz = json.load(f)

print("Python Quiz Program")

print("Enter 't' for True or 'f' for False (if applicable)")

name = input("Enter your full name: ")

# Display and process questions

for item in quiz:

    ques = item["question"]

    correct\_answer = item["answer"].strip().lower()

    print("Question", number, ":", ques)

    ans = input("The answer is: ").strip().lower()  # Convert answer to lowercase for case-insensitive comparison

    if ans == correct\_answer:  # Compare answers ignoring case

        scores += 1

        print("Correct")

    else:

        print("Wrong")

    number += 1

# Write result to file

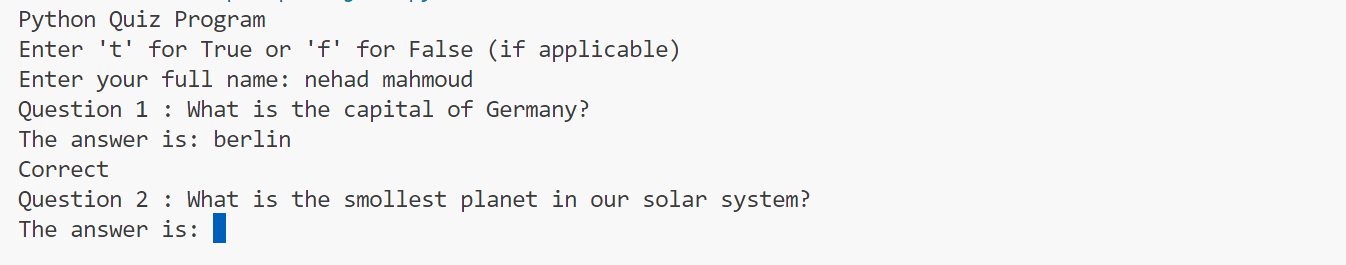
result = {name: scores}

with open("results.txt", 'a') as m:  # Use 'a' to append to the file

    json.dump(result, m)

    m.write('\n')  # Add a new line for each result

output:



#question4:

class Bankaccount:

    def \_\_init\_\_(self,account\_number,account\_holder,initial\_balance=0.0):

        self.account\_number=account\_number

        self.balance=initial\_balance

        self.account\_holder=account\_holder

    def deposit(self,amount):

        self.balance +=amount

        print("deposit=",+amount)

    def withdraw(self,amount):

        if self.balance>=amount:

            self.balance -=amount

        print("withdraw=",+amount)

    def get\_balance(self):

        print("balance=",+self.balance)

n = Bankaccount("11254","nehad")

print("account number is:",n.account\_number," ","Full Name is:",n.account\_holder)

n.deposit(100)

n.withdraw(50)

n.get\_balance()

class savingaccount(Bankaccount):

    def \_\_init\_\_(self,account\_number,account\_holder, balance=0,interset\_rate=0):

        super().\_\_init\_\_(account\_number,account\_holder,balance)

        self.interest\_rate=interset\_rate

    def apply\_interest(self):

        self.balance +=self.balance \* (self.interest\_rate / 100)

    def overrideprint(self):

        print("new balance:" ,self.balance ,"new interest rate:" , self.interest\_rate)

v=savingaccount("123456","nehad",8,4)

v.apply\_interest()

v.overrideprint()

output:

