PYTHON CODES:  
  
Q1. # 3 rows x 2 cols

matrix= [ [2,3],[4,5],[6,7]]

print(matrix)

# 3x3 identity matrix

identity = [[1,0,0],[0,1,0],[0,0,1]]

print (identity)

# 3x3

matrix\_A = [[10,20,20],[40,51,60],[70,80,91]]

print(matrix\_A)

# matrix\_B = matrix\_A + identity

matrix\_B= [[0,0,0],[0,0,0],[0,0,0]]

nrows = ncols = 3

for i in range(0,nrows): #0,1,2

    for j in range(0,ncols): #0,1,2

        matrix\_B[i][j] = matrix\_A[i][j]+identity[i][j]

print(matrix\_B)

print("\n") #Style 1

for i in range(nrows):

    for j in range(ncols):

        print(matrix\_B[i][j], end="\t")

    print()

print("\n") #Style 2

for i in range(nrows):

    print("|", end='\t')

    for j in range(ncols):

        print(matrix\_B[i][j], end="\t")

    print("|")

Q2. (YOU NEED TO DO)while (True):

    s = ''

    print("Enter the details for matrix 1")

    rows\_1=int(input("Enter the numbers of rows:"))

    cols\_1=int(input("Enter the numbers of columns:"))

    print("Enter the details for matrix 2")

    rows\_2=int(input("Enter the numbers of rows:"))

    cols\_2=int(input("Enter the numbers of columns:"))

    if cols\_1==rows\_2:

        break

    else:

        print("Cannot multiply the two matrices")

        print("No. of cols of matrix 1 should be equal to no.of rows in matrix 2")

        print("If you wish to end the program, type \"end\"")

        s=input("Type anyhting else to continue")

        if s=="end": break

if s=="end":

    print("Thank You!")

else:

    print("Enter the elements of Matrix 1:")

    matrix\_1 = [[int(input())for x in range(cols\_1)]for y in range(rows\_2)]

    print(matrix\_1)

    print("Enter the elements of Matrix 2:")

    matrix\_2=[[int(input())for x in range(cols\_2)] for y in range(rows\_1)]

    print(matrix\_2)

    matrix\_result = [[0 for\_ in range(cols\_2)] for\_ in range(rows\_1)]

    print(matrix\_result)

for i in range(rows\_1):

    for j in range(cols\_2):

        for k in range(cols\_1):

            matrix\_result[i][j]+=matrix\_1[i][k]\*matrix

Q3. citizen = ("Tom", 123456, "A+")

print(citizen)

print(citizen[0],citizen[2])

for elem in citizen:

    print(elem)

citizen.append(987654321)

citizen[2] = "B-"

citizen\_twice=citizen\*2

print("citizen\_twice")

if("A+" in citizen):

    print("Citizen's blood group is A+")

Q4. (THERE IS AN ERROR IN THIS PLEASE CHECK)myset = {1,2,3}

print(myset, type(myset), len(myset))

myset={1,2,3,3,2}

print(myset, len(myset))

mylist = [1,2,3,3,2]

mylist = list(set(mylist))

print(mylist)

myset = {"vit","vellore","python","vit"}

print(myset)

print(myset[0])

for elem in myset:

    print(elem)

" " "\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_"""

myset={1,2,3}

myset.add(100)

print(myset)

mylist=[200,300,400]

myset.update(mylist)

print(myset)

mystring="vit"

myset.update(mystring)

print(myset)

myset.add(mystring)

print(myset)

myset.remove(200)

print(myset)

first\_element= myset.pop()

print(first\_element)

print(myset)

Q5. set1={1,2,3}

set2={3,4,5}

print(set1.union(set2))

print(set1.intersection(set2))

print(set1.difference(set2))

print(set1.isdisjoint(set2))

print(set1.issubset(set2))

print(set1.issuperset(set2))

set={1,2}

set={1,2,3,4,5}

print(set1.issubset(set2))

print(set1.issuperset(set2))

Q6. (CHECK THE GENTRIFICATION PART)empty\_dict = {}

print(len(empty\_dict))

fruitbasket = {"apple":3,"banana":5,"cherry":50}

print(len(fruitbasket))

oxford\_dict = {"gentrification":"meaning of gentrification"}

print(oxford\_dict["gentrification"])

numeric\_dict = {1:1, 2:4, 3:9}

print(numeric\_dict[2])

student = {'name':'Abraham','dept':'SENSE','aadhar':123456}

student['aadhar']=999999

print(student['name'], student['aadhar'])

student = {'name':'Abraham', 'dept':'SENSE','aadhar':123456}

student["city"]= 'Vellore'

print(student)

print(student['city'])

del student['aadhar']

print(student)

student['belongings']=['watch','bag','suticase','phone','laptop']

print(student['belongings'])

print(student)

student.clear()

print(student)

Q7. (CHECK THE OUTPUT FOR THIS ONE)  
student={'name':'Abraham','dept':'SENSE','aadhar':123456}

print(list(student.keys()))

print(list(student.values()))

print(list(student.items()))

print(student.get("phone"))

if(student.get("phone")==None):

    if(student.get("phone")==None):

        phone\_number=int(input("Enter your phone number:"))

        student["phone"]=phone\_number

        print(student)

        for key in list(student.keys()):

            print(key)

        for val in list(student.values()):

            print(val)

        print(student.items())

        for key,val in list(student.items()):

            print(key,val)

Q8. str1="vit"

str2="vit"

if(str1==str2):

    print("Both are same")

else:

    print("Not same")

print("Geek"<"geek")

print("Geek">"geek")

print("Geek"!= "Geek")

print("Abhi"!= "VIT")

str3= 'vit'

str4= 'vit'

print(id(str3),id(str4))

print(str3 is str4)

str5= 'vit'

str6=(str5 + '.')[:-1]

print(id(str5),id(str6))

print(str5==str6)

print(str5 is str6)

if str5 is not str6:

    print("str5:",str5)

    print("str6:",str6)

else:

    print("They are the same instance")

    print(25\*"=")

Q9. (CHECK THE OUTPUT FOR THIS)name="website"

errorno=100

str1="The"  +str(name)+  "has an error number" +str(100)

print(str1)

str2="the {} is a student with reg.no {}"

print(str2.format("Tom",249876))

str3="the {0} has an error number {1}"

print(str3.format(name,errorno))

print('{2} {1} {0}'.format('directions' , 'the', 'Read'))

print("the {var1} had an error number{var2}".format(var1="website", var2=errorno))

student = "Tom"

age= 16

height= 178.32

print(f"{student} is {age} years old and is {height}cms tall")

print(f"{student} is {age} years old and is {height:1f}cms tall")

Q10. (CHECK OUTPUT)str1 = "vit"

print(str1[1])

str2 = "vit vellore"

print(str2[4:11])

print(str2[-7:0])

sentence = "I study in Vit"

university = sentence[-3:]

print(university.upper())

print(university.lower())

str1="vit.vellore"

print(str1.split(" "))

str2="my name is vit"

print(str2.split('m'))

str3="hello world lol"

print(str3.split("lo"))

str1= "      hello world "

str2= str1.lstrip()

print(str2,len(str2))

string = "the king has the largest army in the entire world"

print(string.strip("the"))

string =" the king has the largest army in the entire world"

print(string.strip("the"))

str1="vit vallora"

print(str1.replace("a","e"))

Q11. (CHECK THIS CODE LAST PART)string=input("Enter a phrase or a sentence:")

l\_string=string.lower()

string\_list=l\_string.split(" ")

print(string\_list)

words={}

for x in string\_list:

    if x in words:

        words[x]+=1

    else:

        words[x]=1

print(words)

temp=0

multiple= False

for x in words:

    if words[x]>temp:

        max=x

        temp=words[x]

    elif words[x]==temp:

        multiple=True

    else:

        pass

    if multiple==True:

        print("Multiple words have greatest number of occurances")

    else:

        print("The word\"",max,"\"occurs",temp,"times and")

Q12. Tom={"sem1":["A","S","C","D","B","A"],

      "sem2":["B","A","S","S","C","B"],

       "sem3":["C","B","N","F","A","S"]}

std\_names=("Tom","Jerry")

std\_sgpa={}

std\_cgpa={}

no\_of\_stds=1

for i in range(no\_of\_stds):

    sgpa\_l=[]

    name=std\_names[i]

    for sem in Tom:

        SGPA=0.0

        for grade in Tom[sem]:

            if grade=="S":

                SGPA+=10

            elif grade =="A":

                SGPA+=9

            elif grade=="B":

                SGPA+=8

            elif grade =="C":

                SGPA+=7

            elif grade =="D":

                SGPA+=6

            elif grade =="E":

                SGPA+=5

            else:

                SGPA+=4

        SGPA /=len(Tom[sem])

        sgpa\_l.append(SGPA)

    std\_sgpa[f"{name}\_SGPAs"] = sgpa\_l

print(std\_sgpa)

CGPA=0.0

for name in std\_sgpa:

    for sgpa in std\_sgpa[name]:

        CGPA += sgpa

    CGPA/=len(Tom)

    temp=name[:-6]

    std\_cgpa[f"{temp}\_CGPA"]=CGPA

print(std\_cgpa)