**Report on Studio Project**

**TIMETABLE MANAGEMENT SYSTEM**

**-by**

**Rakshith Subramanian (3122 21 5001 078)**

**Prajit.R (3122 21 5001 068)**

**S. Rohitram (3122 21 5001 086)**

1. **Abstract:**

This project aims to develop a system to produce a university timetable adhering to a list of constraints. The system takes in inputs such as semester, section, faculty names, subject handled by faculties and their classes and provides a comprehensive timetable for both students and faculties which would otherwise take longer and be more tedious if done manually. This project will make use of a programming language, a database management software and a GUI to develop an interface for the user to operate on. On completion, this system would be able to help large universities to develop timetables for all courses and departments with greater ease.

1. **Introduction:**

The university course timetabling problem looks for the best schedule to satisfy the given criteria as a set of given resources which may contain lecturers, groups of students, classrooms or laboratories. Developing a timetable is a fundamental requirement for the healthy functioning of any educational and administrative part of an academic institution. However, factors such as availability of hours, number of subjects and the allocation of teachers make the problem complex. Thus, the proposal to automate this process intends to meet real needs with the aim of reducing the time taken to complete it. Computerization of the said process also increases accuracy and efficiency. This project aims to utilize, computational techniques to provide a timetable to students and faculties by abiding to necessary constraints. Our project revolves around the fact that priority order of the subject for each section and hours per week for each subject. The subject with maximum priority and more hours per week is assigned first and this keeps on running as a timetable is generated for each section.

1. **Existing Work:**

**[1]**A presentation is made of a blend of Constraint Programming and Local Search Helped us

**[[2]** “Timetable Handling Mechanism Using Python by Manpreet Kaur, Jasdev Bhatti, Mohit Kumar Kakkar, Deepika Goyal” further refined the way we were working on our algorithm and helped us discover new ways to work on the code.

1. **Design**

**Architecture Diagram:**

**Diagram

Description automatically generated**

**Module Design:**

The project is done such a way that Lab sessions are assigned first:

First set of inputs will:

Semester(1-8) (Enter semester in integer from 1-8): say, 1.

Section(A/B) (Enter section in Upper case):say, A.

Then we enter the day on which you wish to schedule your lab class(twice): “Monday” or “Tuesday” or “Wednesday” or “Thursday” or “Friday” (First letter alone in Upper case), say “Monday” and “Wednesday”.

The above loop will run for 16 times, so as to cover all 8 semesters and both the sections.

After the above loop, we enter the total number of faculties that have to be scheduled across all semesters ,say 10 faculty.

Then the unique name of the faculty is entered(as a string): say, John.

Now the total number of subject handled by that faculty across dept is entered (as an integer): say, 3.

Enter the semester no. (in integer): say, 1

Enter section (in upper case) : say, A

Enter the subject handled by that faculty for that semester and section that is 1A: for example, Maths.

Hence, John takes Mathematics for Semester 1 and Section A.

This inner loop runs for (total number of subject of individual faculty) times.

It has to be ensured that, while entering the above data:

\*They are not in the same index of the priority list of the subjects which is mentioned in the top of code.\*

eg :

If Ramesh Takes 2 Classes

i,e Sem 2 and A sec and Sem 5 and B sec:

The priority order of the subject is [7 hours,7 hours,6 hours,6 hours,6 hours]

For sem2 secA : ["C","Data Science","Maths","HSE","BEEE"]

For sem5 secb : ["M.I","Computer Networks","ToC","Crypto","A.I"]

Hence, C,Data Science,M.I, Computer Networks are all 7 hour subject per week for the student while the others are for 6 hours.

While choosing the subject it has to be ensured they are not in same index and do not have same subject hour across all their subjects handled i.e for example if Maths is chosen for Ramesh to handle Sem2 Sec A, then avoiding "ToC" for Ramesh in Sem5 SecB and is chosen from "Crypto" and "A.I“.

The priority order for all faculties while giving their subject has to be maintained .

It has to be ensured that the maximum number of class handled by the faculty is 3 if he/she is teaching a 6 hour course and 2 for a 7 hour course.

**5)Implementation and Results:**

S\_D=["Monday","Tuesday","Wednesday","Thursday","Friday"]

Total\_Sub={1:[["UMA2176","UGE2176","UPH2176","UCY2176","UGE2177"],

                ["UGE2176","UMA2176","UCY2176","UGE2177","UPH2176"]],

           2:[["UCS2201","UCS2202","UMA2276","UEN2276","UEE2276"],

              ["UCS2202","UCS2201","UEN2276","UEE2276","UMA2276"]],

           3:[["UCS2303","UCS2302","UCS2301","UMA2377","UHS2351"],

              ["UCS2302","UCS2303","UMA2377","UHS2351","UCS2301"]],

           4:[["UCS2404","UCS2401","UCS2403","UCS2402","UMA2455"],

              ["UCS2401","UCS2404","UCS2402","UMA2455","UCS2403"]],

           5:[["UCS2501","UCS2502","UCS2504","UCS2503","UCS2520"],

              ["UCS2502","UCS2501","UCS2503","UCS2520","UCS2504"]],

           6:[["UCS2601","UCS2602","UCS2603","UCS2604","UCS2620"],

              ["UCS2602","UCS2601","UCS2604","UCS2620","UCS2603"]],

           7:[["UCS2701","UCS2703","UCS2702","UCS2720","UCS2730"],

              ["UCS2703","UCS2701","UCS2720","UCS2730","UCS2702"]],

           8:[["UCS2820","UCS2830","UCS2818","UCS2818","UCS2818"],

              ["UCS2830","UCS2820","UCS2818","UCS2818","UCS2818"]]}

Abbv={1:[["Matrices and Calculus","Problem Solving and Programming in Python","Engineering Physics","Engineering Chemistry","Engineering Graphics"],

         ["Problem Solving and Programming in Python","Matrices and Calculus","Engineering Chemistry","Engineering Graphics","Engineering Physics"]],

      2:[["Fundamentals and Practice of Software Development (TCP)","Foundations of Data Science","Complex Functions and Laplace Transformations","Humanities Elective","Basic Electrical and Electronics"],

       ["Foundations of Data Science","Fundamentals and Practice of Software Development (TCP)","Humanities Elective","Engineering","Complex Functions and Laplace Transformations"]],

      3:[["Object Oriented Programming","Data Structures","Digital Principles and System Design","Discrete Mathematics","Universal Human Values II"],

       ["Data Structures","Object Oriented Programming","Discrete Mathematics","Universal Human Values II","Digital Principles and System Design"]],

      4:[["Database Management Systems","Computer Organization and Architecture","Design and Analysis of Algorithms","Operating Systems","Probablity and Statistics"],

      ["Computer Organization and Architecture","Database Management Systems","Operating Systems","Probablity and Statistics"]],

      5:[["Computer Networks","Microprocessors, microcontrollers,and Interfacing","AFoundations of Artificial Intelligence(TCP)","ToC","Professional Elective-I"],

      ["Microprocessors, microcontrollers,and Interfacing and Interfacing","Computer Networks","ToC","Professional Elective-I","Foundations of Artificial Intelligence(TCP)"]],

      6:[["Internet Programming","Software System Security","Theory of Computation","Proffesional Elective-2","Open Elective-II"],

       ["Software System Security","Internet Programming","Proffesional Elective-2","Open Elective-II","Theory of Computation"]],

      7:[["Distributed Systems","Software Architecture","Compiler Design (TCP)","Proffesional Elective-4","Proffesional Elective-3"],

      ["Software Architecture","Distributed Systems","Proffesional Elective-4","Proffesional Elective-3","Compiler Design (TCP)"]],

      8:[["Professional Elective-5","Proffesional Elective-6","Project Hour","Project Hour","Project Hour"],

      ["Proffesional Elective-6","Proffesional Elective-5","Project Hour","Project Hour","Project Hour"]]}

total\_fac={}

def calculate():

    #A Dictionary to hold the  respective timetable and list with days

    S\_TT={"Monday":[],"Tuesday":[],"Wednesday":[],"Thursday":[],"Friday":[]}

    #Lists to hold the subjects and the Hours of each one in priority order

    Sub\_L=[]

    S\_H=[7,7,6,6,6]

    Free=["Mentor","Library"]

    c=0

    F\_H=4

    #Manually assigning lab hours

    def LabDay():

        L1=input("Enter day on which lab will be conducted:")

        if L1 in S\_TT.keys():

            for i in range(0,3):

                S\_TT[L1].append("Lab")

    #Function used to find index of element with max hours in the hours list

    def max\_index(list0):

        index= 0

        max\_n= list0[0]

        for i in range(0, len(list0)):

            if list0[i] > max\_n:

                max\_n= list0[i]

                index = i

        return index

    #Function to check forenoon hours

    def FNC(Sub\_List,s,S\_DList,m,n):

        flag=0

        if Sub\_List[m] in s[S\_DList[n]]:

            return False

        else:

            return True

    #Collecting Inputs:

    global sem

    global sec

    sem=int(input("Enter Semester:"))

    sec=input("Enter Section(A/B):")

    if sec=='A':

        Sub\_L=Total\_Sub[sem][0]

        LabDay()

        LabDay()

    elif sec=='B':

        Sub\_L=Total\_Sub[sem][1]

    for j in range(5):

        k=max\_index(S\_H)

        for x in range(5):

            if FNC(Sub\_L,S\_TT,S\_D,k,x) is True:

                S\_TT[S\_D[x]].append(Sub\_L[k])

                S\_H[k] =  S\_H[k] - 1

                k=max\_index(S\_H)

    for i in range(5):

        for q in range(5):

            if Sub\_L[i] not in S\_TT[S\_D[q]]:

                S\_TT[S\_D[q]].append(Sub\_L[i])

                S\_H[i]=S\_H[i]-1

    if sec=='B':

        LabDay()

        LabDay()

    for i in range(3):

        for j in range(5):

            if len(S\_TT[S\_D[j]])<8:

                   w=max\_index(S\_H)

                   if S\_H[w]>0:

                       if Sub\_L[w] not in S\_TT[S\_D[j]][5:]:

                           S\_TT[S\_D[j]].append(Sub\_L[w])

                           S\_H[w]-=1

                   elif S\_H[w]==0 :

                       S\_TT[S\_D[j]].append(Free[c])

                       c+=1

    if sec=='B':

        S\_TT["Monday"][3],S\_TT["Monday"][4]=S\_TT["Monday"][4],S\_TT["Monday"][3]

        S\_TT["Wednesday"][3],S\_TT["Wednesday"][4]=S\_TT["Wednesday"][4],S\_TT["Wednesday"][3]

    print("  Day  ","  1st ","   2nd   ","3rd   "," 4th  ","5th   ","6th  ","  7th   ","8th   ")

    for i in S\_D:

        print(i,":",end=" ")

        for j in S\_TT[i]:

            print(j,end="\t")

        print()

    print()

    print()

    return S\_TT

Clg\_TT={1:[0,0],2:[0,0],3:[0,0],4:[0,0],5:[0,0],6:[0,0],6:[0,0],7:[0,0],8:[0,0]}

for i in range(4):

    for i in range(2):

        L1=calculate()

        if sec=='A':

            x=0

        else:

            x=1

        Clg\_TT[sem][x]=L1

print()

print()

fac=int(input("enter total number of faculties whose timetable is to be checked:"))

for i in range(fac):

    print()

    name=input("enter the unique name of the faculty:")

    total\_fac[name]={"Monday":['0','0','0','0','0','0','0','0'],"Tuesday":['0','0','0','0','0','0','0','0'],"Wednesday":['0','0','0','0','0','0','0','0'],"Thursday":['0','0','0','0','0','0','0','0'],"Friday":['0','0','0','0','0','0','0','0']}

    numclass=int(input("enter total number of classes taken by above faculty:"))

    for j in range(numclass):

        semester=int(input("enter semester(1-8) taken: "))

        section=input("enter section (A/B) ")

        if section=='A':

            newsection=0

        else:

            newsection=1

        clas=Clg\_TT[semester][newsection]

        A=str(semester)+"-"+str(section)

        print("for ",A)

        """print("please avoid choosing subject of same priority index")"""

        subject=input("enter subject handled for the above class:")

        for day in S\_D:

            for period in range(8):

                if clas[day][period]==subject:

                    total\_fac[name][day][period]=A

for i in total\_fac.keys():

    print()

    print()

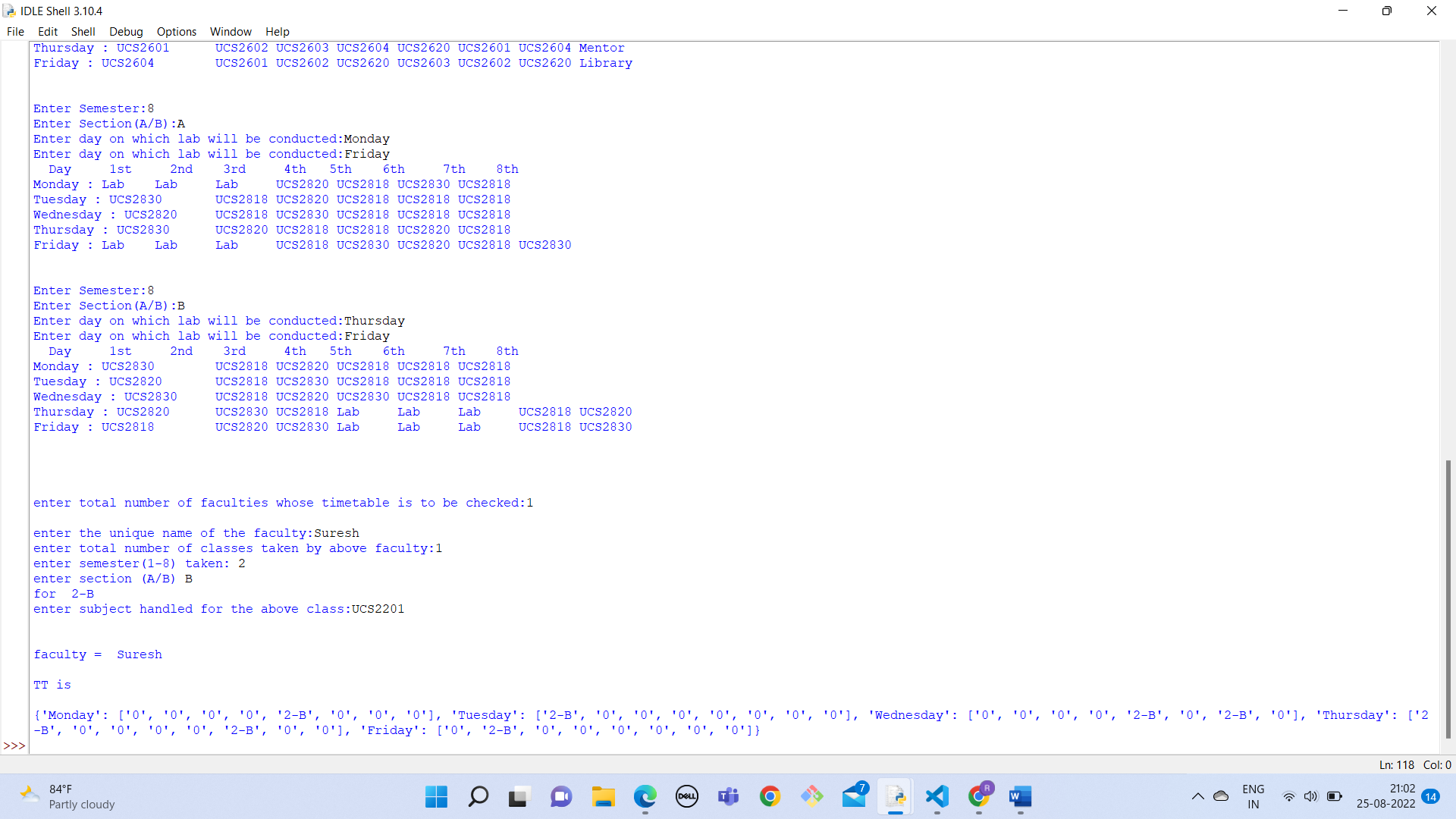
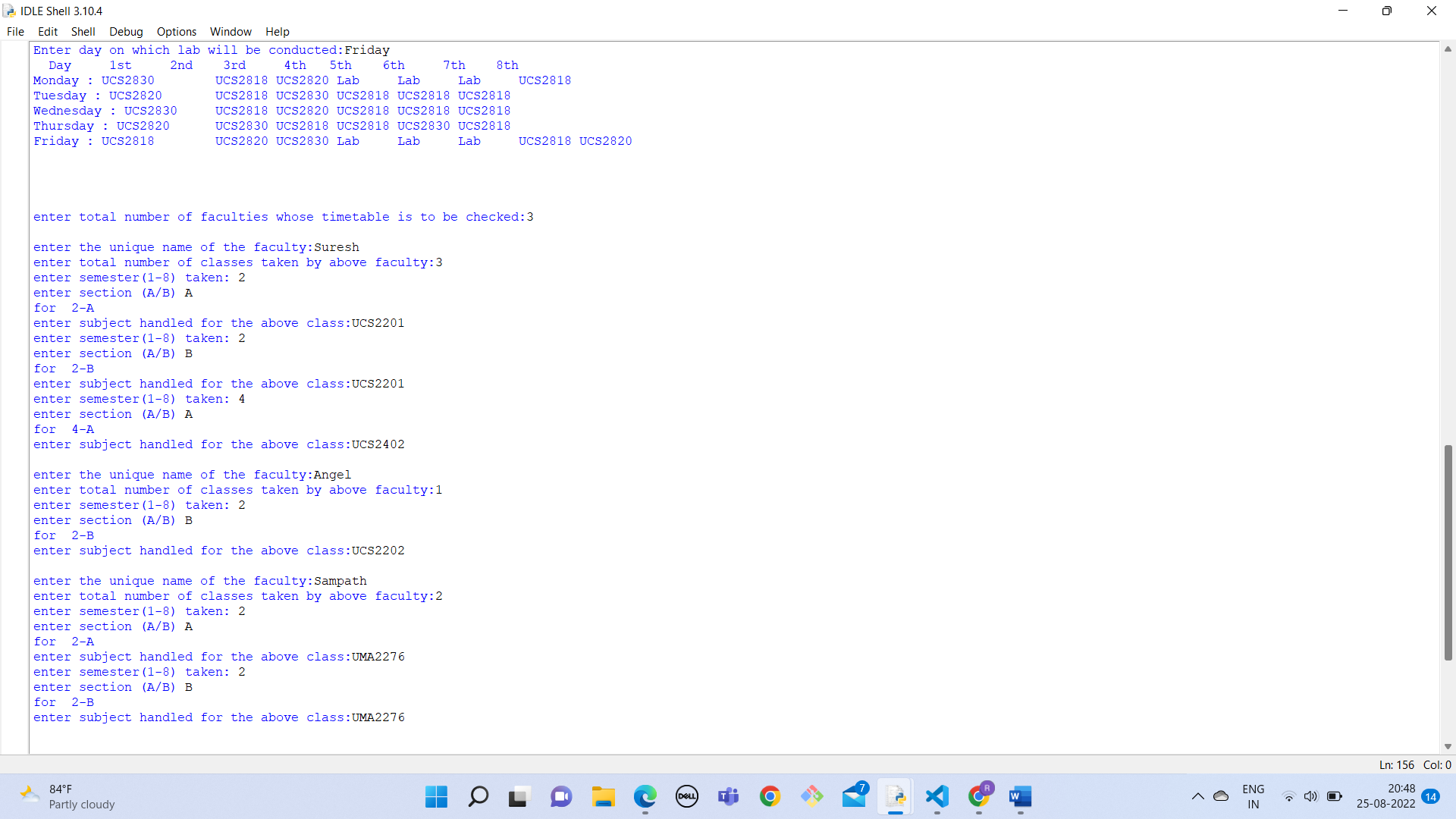
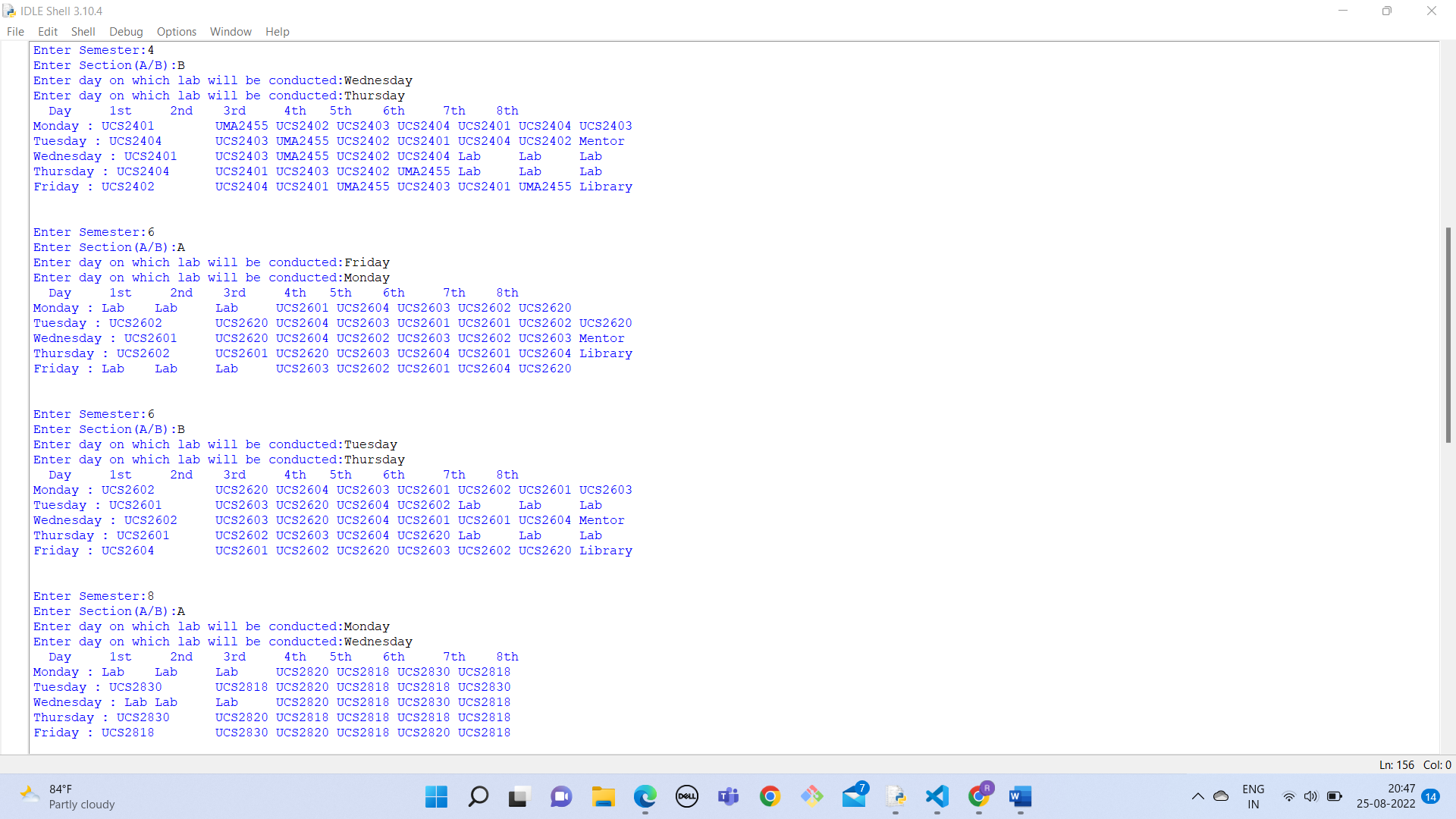
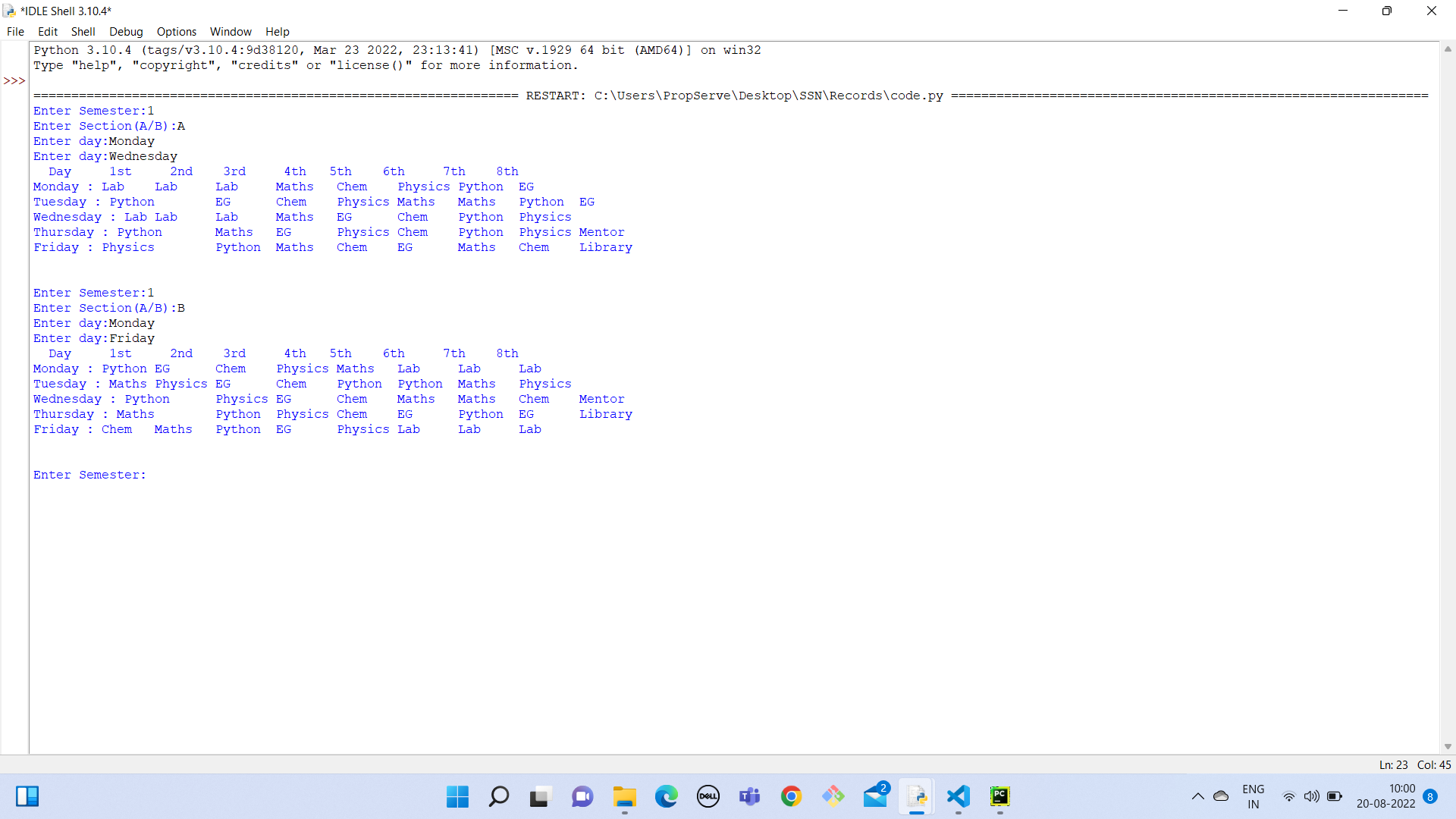
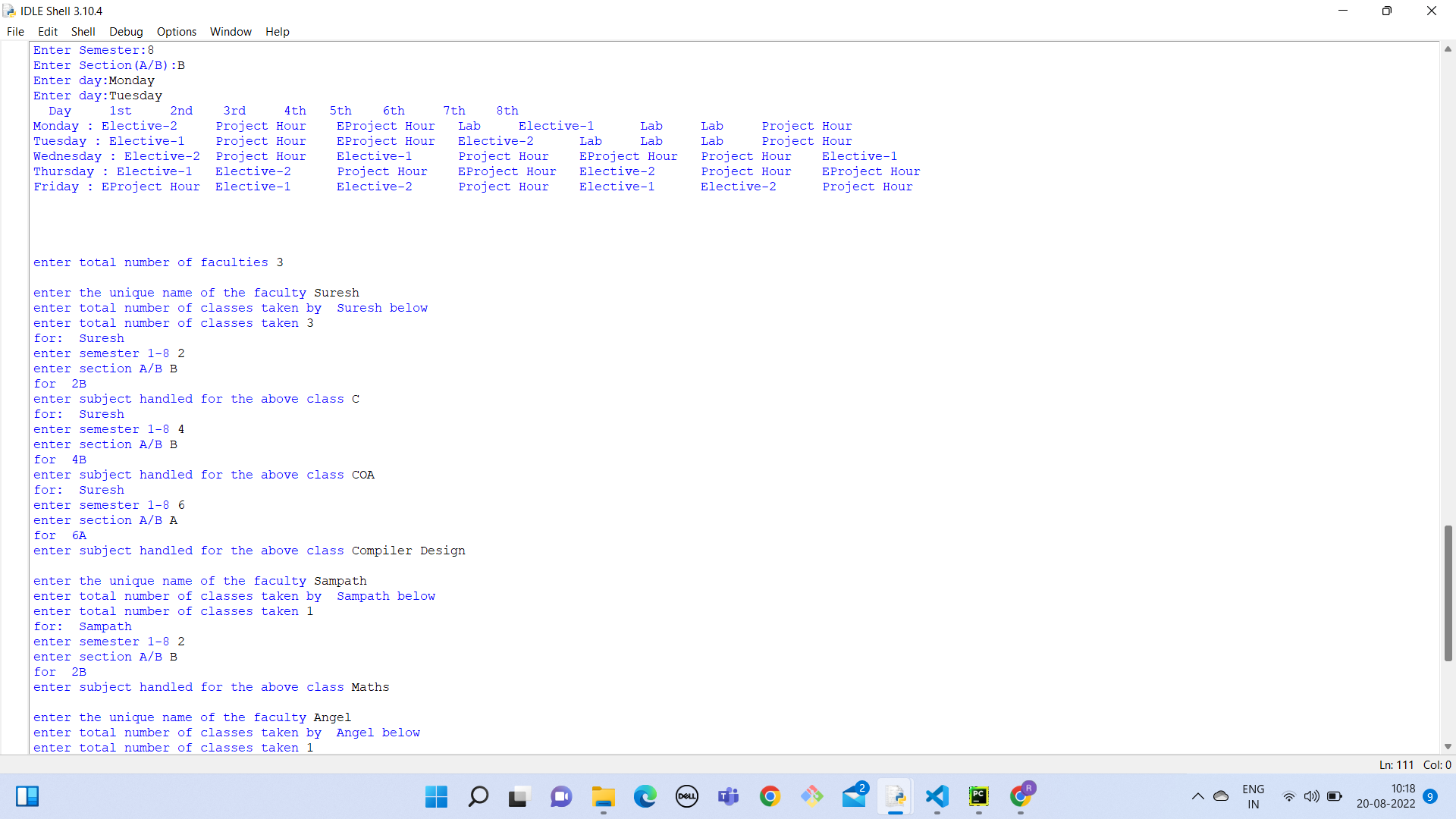
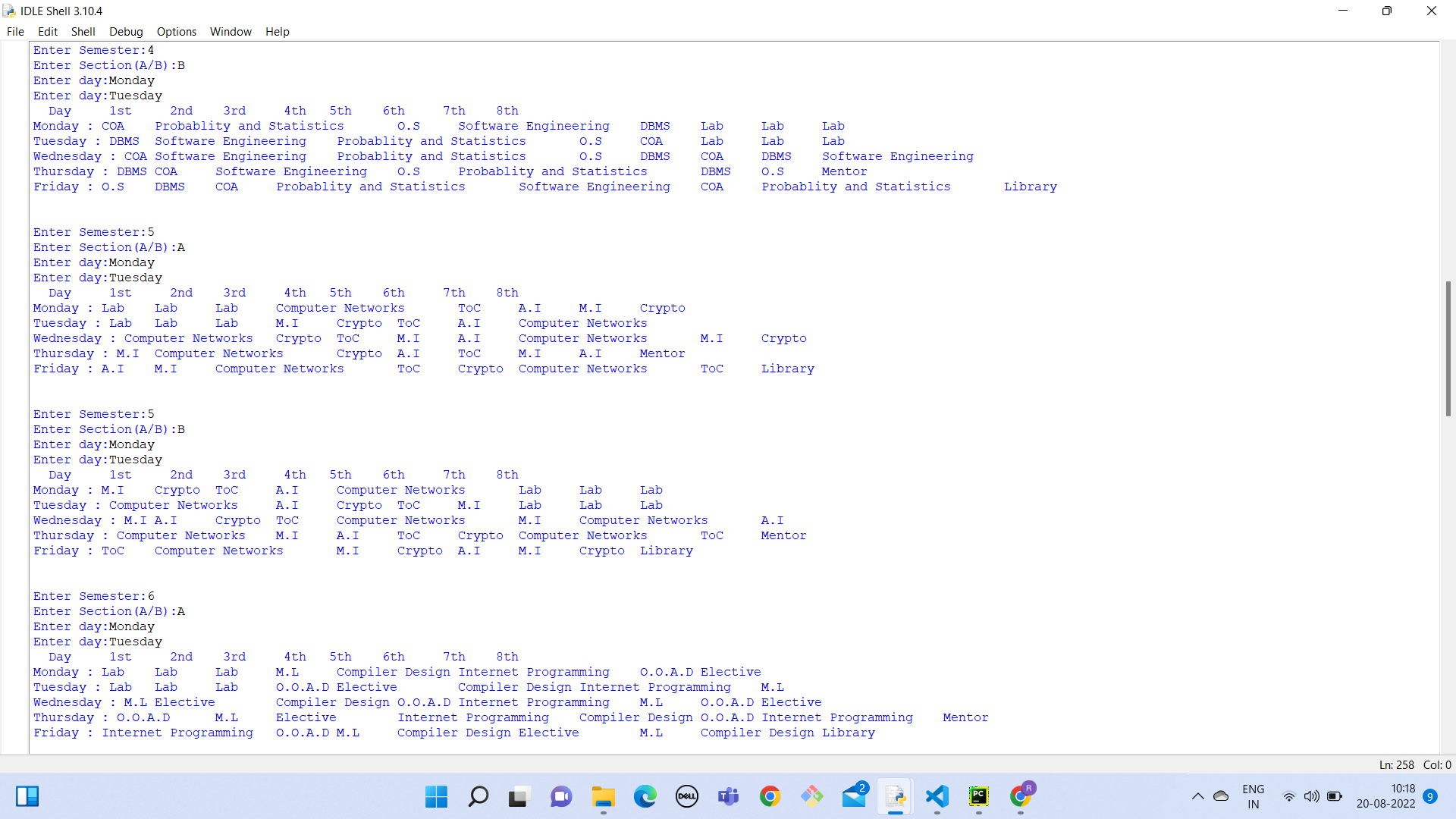
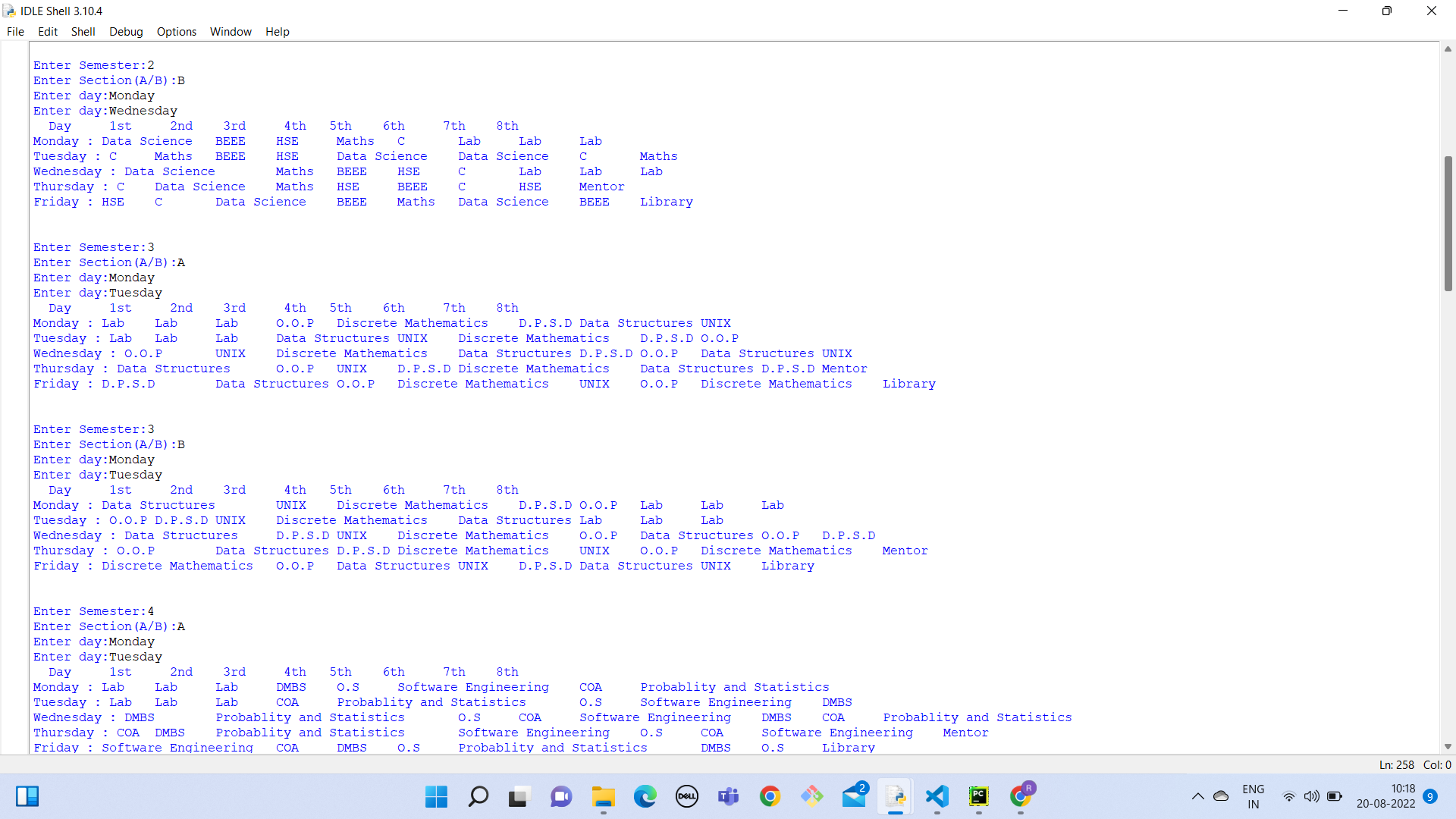
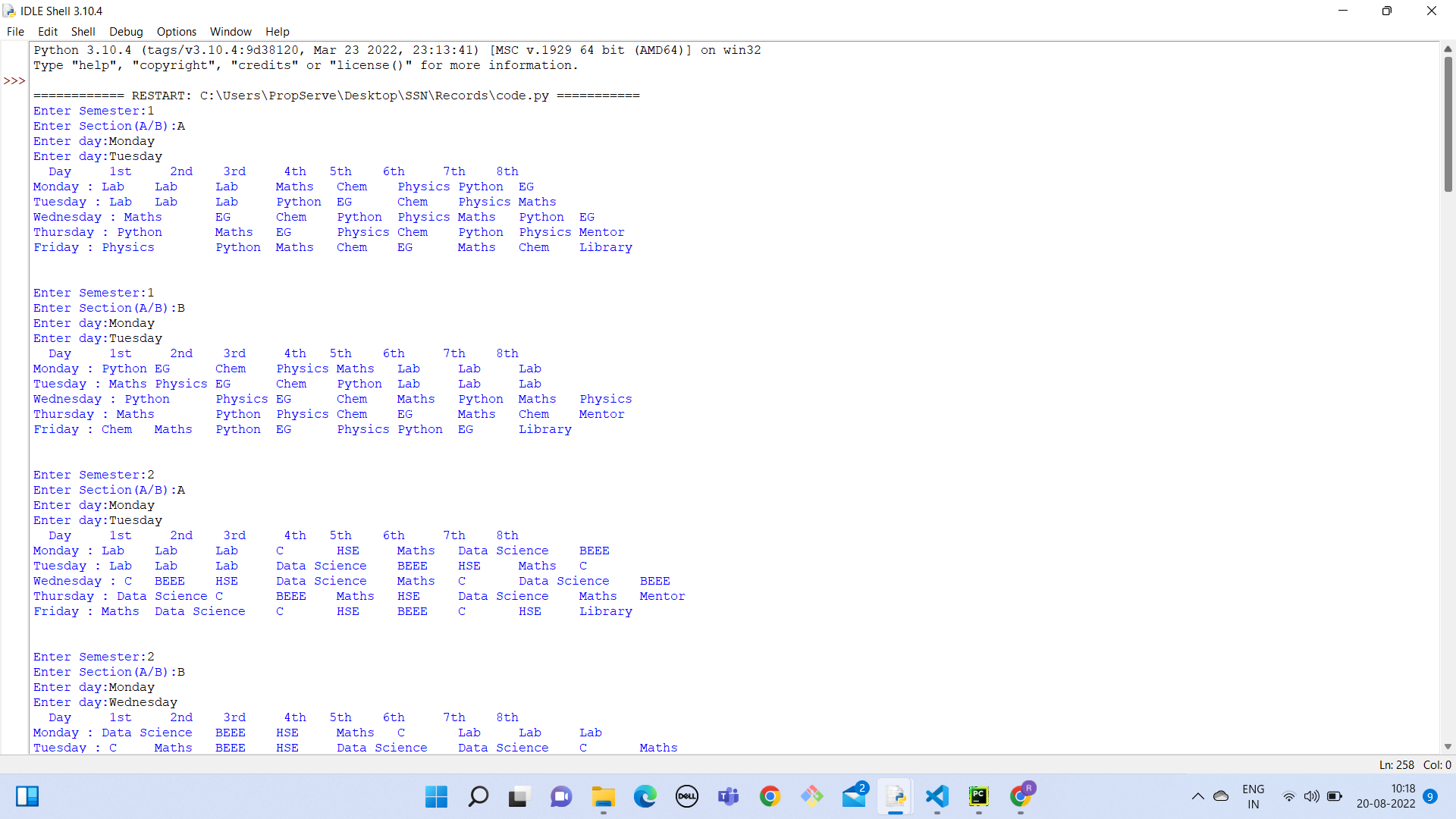
    print("faculty = ",i)

    print()

    print("TT is")

    print()

    print(total\_fac[i])

****

**6)Conclusion:**

The time tabling problem for a department was a intriguing and challenging task. The first problem was satisfying the various constraints of avoiding clashes between teachers’ timetables while simultaneously slotting student timetables. The use of a priority order with subjects and hours occupied by each resolved the issue of slotting and further, the top to bottom approach of filling in the timetables instead of filling it row by row nullified various other issues that would have been faced while satisfying necessary constraints. All in all, a viable solution for timetable management has been formed that uses a simplistic approach while satisfying necessary and primarily evident rules and constraints.

**7)References:**

• "AUTOMATIC TIMETABLE GENERATION SYSTEM", International Journal of Emerging Technologies and Innovative Research ,

• [2] Francis, kussu. 2022. "Scheduling Algorithm For University Timetabling Problem". Academia.Edu. https://www.academia.edu/29810480/Scheduling\_Algorithm\_for\_University\_Timetabling\_Problem • Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm (0) by Dilip Datta, Kalyanmoy Deb, Carlos M Fonseca)

Top of Form

 Download Report

Bottom of Form