Theory :

Partitioning the tables/databases is very important step in parallelizing the database activities. By partitioning the data equally into many different processors’ workload, we can achieve better performance (better parallelism) of the whole system.

#### A. Round-Robin Partitioning:

In this strategy we partition records in a round-robin manner using the function i mod n, where i is the record position in the table and n is the number of partitions/disks which is in our case 3. On the application of partitioning technique, first record goes into D1, second record goes into D2, third record goes into D0, fourth record goes into D1, and so on. After distribution of records, we will get the following partitions;

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition0 | | |
| ENAME | GRADE | DNAME |
| FORD | 4 | RESEARCH |
| MILLER | 2 | ACCOUNTING |
| MARTIN | 2 | SALES |
| JAMES | 1 | SALES |

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition1 | | |
| ENAME | GRADE | DNAME |
| SMITH | 1 | RESEARCH |
| KING | 5 | ACCOUNTING |
| TURNER | 3 | SALES |
| ADAMS | 1 | RESEARCH |
| CLARK | 4 | ACCOUNTING |

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition2 | | |
| ENAME | GRADE | DNAME |
| BLAKE | 4 | SALES |
| SCOTT | 4 | RESEARCH |
| WARD | 2 | SALES |
| JONES | 4 | RESEARCH |
| ALLEN | 3 | SALES |

#### B. Hash Partitioning:

Let us take GRADE attribute of the Emp\_table to explain Hash partitioning. Let us choose a hash function as follows;

h(GRADE) = (GRADE mod n)

where GRADE is the value of GRADE attribute of a record and n is number of partitions which is 3 in our case. While applying the hash partitioning on GRADE, we will get the following partitions of Emp\_table. For example, the GRADE of ‘Smith’ is 1 and while hashing the function shows partition 1 (i.e 1 mod 3 = 1). The GRADE of ‘Blake’ is 4, then (4 mod 3) directs to partition 1. The GRADE of ‘King’ is 5 which directs to partition 2 (5 mod 3 = 2).

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition0 | | |
| ENAME | GRADE | DNAME |
| TURNER | 3 | SALES |
| ALLEN | 3 | SALES |

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition1 | | |
| ENAME | GRADE | DNAME |
| SMITH | 1 | RESEARCH |
| BLAKE | 4 | SALES |
| FORD | 4 | RESEARCH |
| SCOTT | 4 | RESEARCH |
| ADAMS | 1 | RESEARCH |
| JONES | 4 | RESEARCH |
| JAMES | 1 | SALES |
| CLARK | 4 | ACCOUNTING |

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition2 | | |
| ENAME | GRADE | DNAME |
| KING | 5 | ACCOUNTING |
| MILLER | 2 | ACCOUNTING |
| WARD | 2 | SALES |
| MARTIN | 2 | SALES |

#### C. Range Partitioning:

Let us consider GRADE of Emp\_table to partition under range partitioning. For applying range partition, we need to first identify partitioning vector, [v0, v1, …, vn-2]. Let us choose the following vector as range partitioning vector for our case;

[2, 4]

According to the vector, the records having the GRADE value 2 and less will go into partition 0, greater than 2 and less than or equal to 4 will go into partition 1, and all the other values (greater than 4) will go into partition 2 as depicted in the following tables.

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition0 | | |
| ENAME | GRADE | DNAME |
| SMITH | 1 | RESEARCH |
| MILLER | 2 | ACCOUNTING |
| WARD | 2 | SALES |
| MARTIN | 2 | SALES |
| ADAMS | 1 | RESEARCH |
| JAMES | 1 | SALES |

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition1 | | |
| ENAME | GRADE | DNAME |
| BLAKE | 4 | SALES |
| FORD | 4 | RESEARCH |
| SCOTT | 4 | RESEARCH |
| TURNER | 3 | SALES |
| JONES | 4 | RESEARCH |
| CLARK | 4 | ACCOUNTING |
| ALLEN | 3 | SALES |

|  |  |  |
| --- | --- | --- |
| Emp\_table\_Partition2 | | |
| ENAME | GRADE | DNAME |
| KING | 5 | ACCOUNTING |

Program Code :

**Round Robin Partitioning**

from tkinter import \*

from tkinter.ttk import \*

import mysql.connector

#user info

def clicked():

    root.withdraw()

    window = Tk()

    window.title("Student Details")

    window.geometry('600x400')

    window.configure(bg='white')

    connection = mysql.connector.connect(

        host="localhost",

        user="root",

        passwd="root",

        database="student"

    )

    cursor = connection.cursor()

    #display option

    def Display():

        # window.withdraw()

        newwin = Toplevel(window)

        newwin.title("Display Details")

        newwin.geometry("1000x600")

        list = Listbox(newwin, height=40, width=80, bg="black", activestyle='dotbox', font="Helvetica", fg="white")

        list.place(x=340, y=30)

        def show():

            con = mysql.connector.connect(

                host="localhost",

                user="root",

                passwd="root",

                database="student"

            )

            cursor = con.cursor()

            cursor.execute("select \* from studenttable")

            rows = cursor.fetchall()

            for row in rows:

                insertData = str(row[0]) + '   ' + str(row[1]) + '    ' + str(row[2])

                list.insert(list.size() , insertData)

            con.commit()

            con.close()

        show()

        #display screen back button

        def backbutton():

            newwin.withdraw()

            window.deiconify()

        btns1 = Button(newwin, text="back", command=backbutton)

        btns1.grid(column=2, row=16)

        def rpartition():

            newwin = Toplevel(window)

            newwin.geometry("1000x600")

            newwin.title("Round Robin Partitioned")

            list=[]

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[0].place(x=70, y=30)

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[1].place(x=440, y=30)

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[2].place(x=790, y=30)

            for j in range(3):

                con = mysql.connector.connect(

                    host="localhost",

                    user="root",

                    passwd="root",

                    database="student"

                )

                cursor = con.cursor()

                cursor.execute("DROP TABLE IF EXISTS std"+str(j))

                cursor.execute("CREATE TABLE std"+str(j)+" AS SELECT \* FROM studenttable WHERE rollno % 3 = " + str(j) +";")

                cursor.execute("Select \* from std"+str(j)+ " ; ")

                rows = cursor.fetchall()

                for row in rows:

                    insertData = str(row[0]) + '   ' + str(row[1]) + '    ' + str(row[2])

                    list[j].insert(list[j].size() , insertData)

        btn2=Button(newwin,text="Partition",command=rpartition)

        btn2.grid(column=2, row=18)

    #display button of user info

    btn1 = Button(window, text="Display", command=Display)

    btn1.grid(column=2, row=14)

    #back button of user info page

    def back():

        window.withdraw()

        root.deiconify()

    btns = Button(window, text="back", command=back)

    btns.grid(column=2, row=16)

#main

root = Tk()

root.title("Round Robin Partitioning")

root.geometry("580x200")

root.configure(bg='grey')

lbl = Label(root, text="Round Robin", font=("Times New Roman Bold", 20))

lbl.config(anchor=CENTER)

lbl.pack()

menubar = Menu(root)

filemenu = Menu(menubar, tearoff=0)

filemenu.add\_command(label="Table", command=clicked)

filemenu.add\_separator()

filemenu.add\_command(label="Exit", command=root.quit)

menubar.add\_cascade(label="Dashboard", menu=filemenu)

editmenu = Menu(menubar, tearoff=0)

editmenu.add\_separator()

root.config(menu=menubar)

root.mainloop()

**Hash Partitioning**

from tkinter import \*

from tkinter.ttk import \*

import mysql.connector

#user info

def clicked():

    root.withdraw()

    window = Tk()

    window.title("Student Details")

    window.geometry('600x400')

    window.configure(bg='white')

    connection = mysql.connector.connect(

        host="localhost",

        user="root",

        passwd="root",

        database="student"

    )

    cursor = connection.cursor()

    #display option

    def Display():

        # window.withdraw()

        newwin = Toplevel(window)

        newwin.title("Display Details")

        newwin.geometry("900x600")

        list = Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white")

        list.place(x=340, y=30)

        def show():

            con = mysql.connector.connect(

                    host="localhost",

                    user="root",

                    passwd="root",

                    database="student"

                )

            cursor = con.cursor()

            cursor.execute("select \* from studenttable")

            rows = cursor.fetchall()

            for row in rows:

                insertData = str(row[0]) + '   ' + str(row[1]) + '    ' + str(row[2])

                list.insert(list.size() , insertData)

            con.commit()

            con.close()

        show()

        #display screen back button

        def backbutton():

            newwin.withdraw()

            window.deiconify()

        btns1 = Button(newwin, text="back", command=backbutton)

        btns1.grid(column=2, row=16)

        def partition():

            newwin = Toplevel(window)

            newwin.title("Hash Partitioned")

            newwin.geometry("1000x900")

            list=[]

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[0].place(x=70, y=70)

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[1].place(x=440, y=70)

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[2].place(x=790, y=70)

            def HashPart():

                conn = mysql.connector.connect(

                            host="localhost",

                            user="root",

                            passwd="root",

                            database="student"

                        )

                cursor = conn.cursor()

                for i in range(3):

                    cursor.execute("DROP TABLE IF EXISTS stud"+str(i))

                    sql = "CREATE TABLE stud"+str(i)+" AS (SELECT \* FROM studenttable WHERE (marks%10)%3 = " +str(i)+" ) ; "

                    cursor.execute(sql)

                for j in range(3):

                    cursor.execute("Select \* from stud"+str(j)+ " ; ")

                    rows = cursor.fetchall()

                    for row in rows:

                        insertData = str(row[0]) + '   ' + str(row[1]) + '    ' + str(row[2])

                        list[j].insert(list[j].size() , insertData)

            btn2=Button(newwin,text="HashPartition",command=HashPart)

            btn2.grid(column=4, row=20)

        btn2=Button(newwin,text="Partition",command=partition)

        btn2.grid(column=2, row=18)

    #display button of user info

    btn1 = Button(window, text="Display", command=Display)

    btn1.grid(column=2, row=14)

    #back button of user info page

    def back():

        window.withdraw()

        root.deiconify()

    btns = Button(window, text="back", command=back)

    btns.grid(column=2, row=16)

#main

root = Tk()

root.title("Hash Partitioning")

root.geometry("580x200")

root.configure(bg='grey')

lbl = Label(root, text="Hash Partitioning", font=("Times New Roman Bold", 20))

lbl.config(anchor=CENTER)

lbl.pack()

menubar = Menu(root)

filemenu = Menu(menubar, tearoff=0)

filemenu.add\_command(label="Table", command=clicked)

filemenu.add\_separator()

filemenu.add\_command(label="Exit", command=root.quit)

menubar.add\_cascade(label="Dashboard", menu=filemenu)

editmenu = Menu(menubar, tearoff=0)

editmenu.add\_separator()

root.config(menu=menubar)

root.mainloop()

**Range Partitioning**

from tkinter import \*

from tkinter.ttk import \*

import mysql.connector

#user info

def clicked():

    root.withdraw()

    window = Tk()

    window.title("Student Details")

    window.geometry('600x400')

    window.configure(bg='white')

    connection = mysql.connector.connect(

        host="localhost",

        user="root",

        passwd="root",

        database="student"

    )

    cursor = connection.cursor()

    #display option

    def Display():

        # window.withdraw()

        newwin = Toplevel(window)

        newwin.title("Display Details")

        newwin.geometry("900x600")

        list = Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white")

        list.place(x=340, y=30)

        def show():

            con = mysql.connector.connect(

                    host="localhost",

                    user="root",

                    passwd="root",

                    database="student"

                )

            cursor = con.cursor()

            cursor.execute("select \* from studenttable")

            rows = cursor.fetchall()

            for row in rows:

                insertData = str(row[0]) + '   ' + str(row[1]) + '    ' + str(row[2])

                list.insert(list.size() , insertData)

            con.commit()

            con.close()

        show()

        #display screen back button

        def backbutton():

            newwin.withdraw()

            window.deiconify()

        btns1 = Button(newwin, text="back", command=backbutton)

        btns1.grid(column=2, row=16)

        def rpartition():

            newwin = Toplevel(window)

            newwin.title("Range Partitioned")

            list=[]

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[0].place(x=70, y=70)

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[1].place(x=440, y=70)

            list.append(Listbox(newwin, height=25, width=30, bg="black", activestyle='dotbox', font="Helvetica", fg="white"))

            list[2].place(x=790, y=70)

            l1 = Label(newwin, text="Range 1 :")      #range label

            l1.grid(column=0, row=1)

            val1 = Entry(newwin, width=30)

            val1.grid(column=1, row=1)

            l2 = Label(newwin, text="Range 2 :")      #range label

            l2.grid(column=0, row=2)

            val2 = Entry(newwin, width=30)

            val2.grid(column=1, row=2)

            def rangepartition2():

                con = mysql.connector.connect(

                        host="localhost",

                        user="root",

                        passwd="root",

                        database="student"

                    )

                cursor = con.cursor()

                for i in range(3):

                    cursor.execute("DROP TABLE IF EXISTS stud"+str(i))

                    if i==0:

                        sql = "CREATE TABLE IF NOT EXISTS studen0 AS (SELECT \* FROM studenttable WHERE marks <="+ val1.get()+" ) ; "

                    elif i==1:

                        sql = "CREATE TABLE IF NOT EXISTS studen"+str(i)+" AS (SELECT \* FROM studenttable WHERE marks >" + val1.get() +" and marks <="+ val2.get()+" ) ; "

                    else:

                        sql = "CREATE TABLE IF NOT EXISTS studen"+str(i)+" AS (SELECT \* FROM studenttable WHERE marks >" + val2.get()+" ) ; "

                    cursor.execute(sql)

                for j in range(3):

                    cursor.execute("Select \* from studen"+str(j)+ " ; ")

                    rows = cursor.fetchall()

                    for row in rows:

                        insertData = str(row[0]) + '   ' + str(row[1]) + '    ' + str(row[2])

                        list[j].insert(list[j].size() , insertData)

            btn2=Button(newwin,text="RangePartition",command=rangepartition2)

            btn2.grid(column=4, row=20)

        btn2=Button(newwin,text="Partition",command=rpartition)

        btn2.grid(column=2, row=18)

    #display button of user info

    btn1 = Button(window, text="Display", command=Display)

    btn1.grid(column=2, row=14)

    #back button of user info page

    def back():

        window.withdraw()

        root.deiconify()

    btns = Button(window, text="back", command=back)

    btns.grid(column=2, row=16)

#main

root = Tk()

root.title("Range Partitioning")

root.geometry("580x200")

root.configure(bg='grey')

lbl = Label(root, text="Range Partitioning", font=("Times New Roman Bold", 20))

lbl.config(anchor=CENTER)

lbl.pack()

menubar = Menu(root)

filemenu = Menu(menubar, tearoff=0)

filemenu.add\_command(label="Table", command=clicked)

filemenu.add\_separator()

filemenu.add\_command(label="Exit", command=root.quit)

menubar.add\_cascade(label="Dashboard", menu=filemenu)

editmenu = Menu(menubar, tearoff=0)

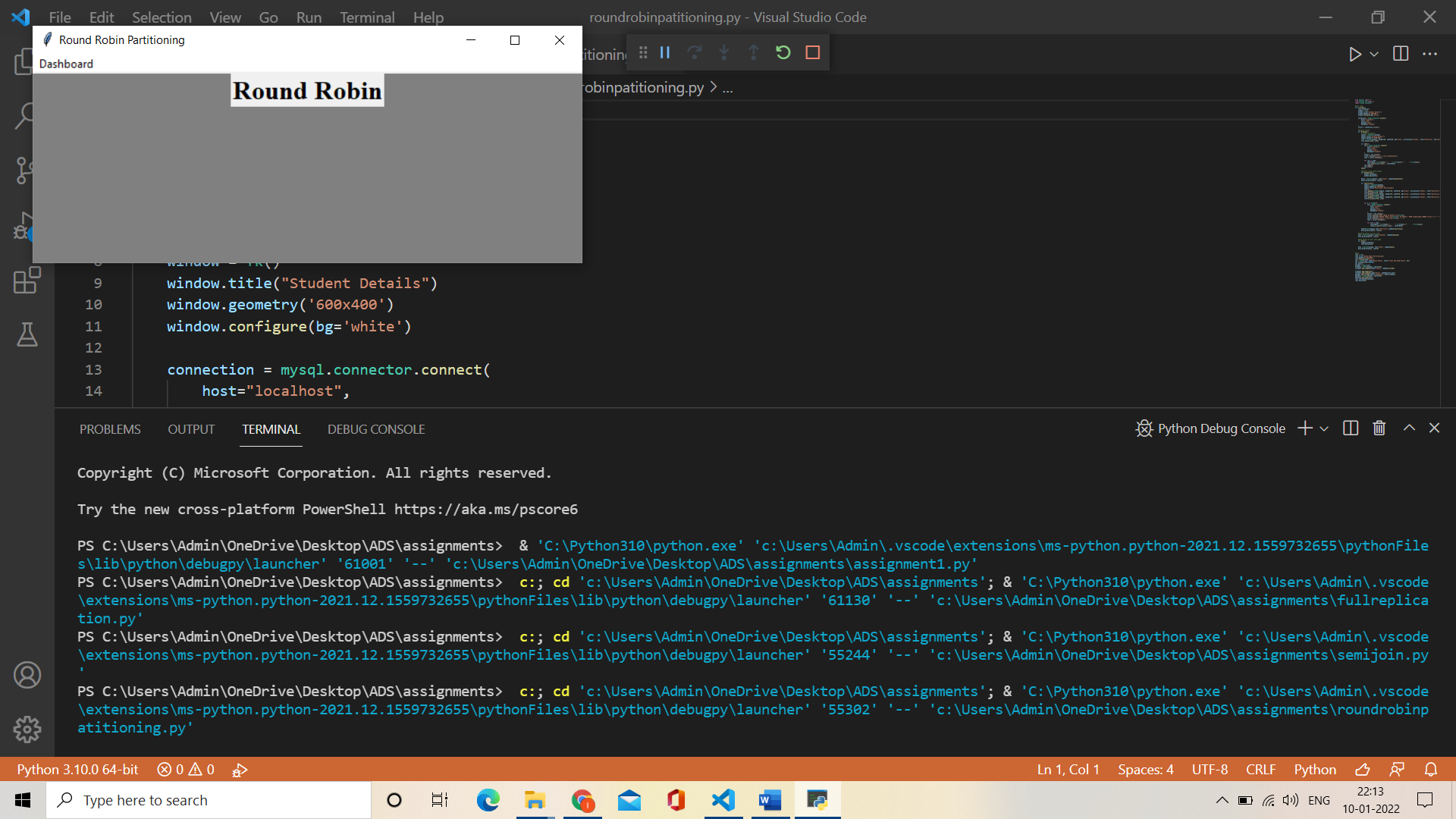
editmenu.add\_separator()

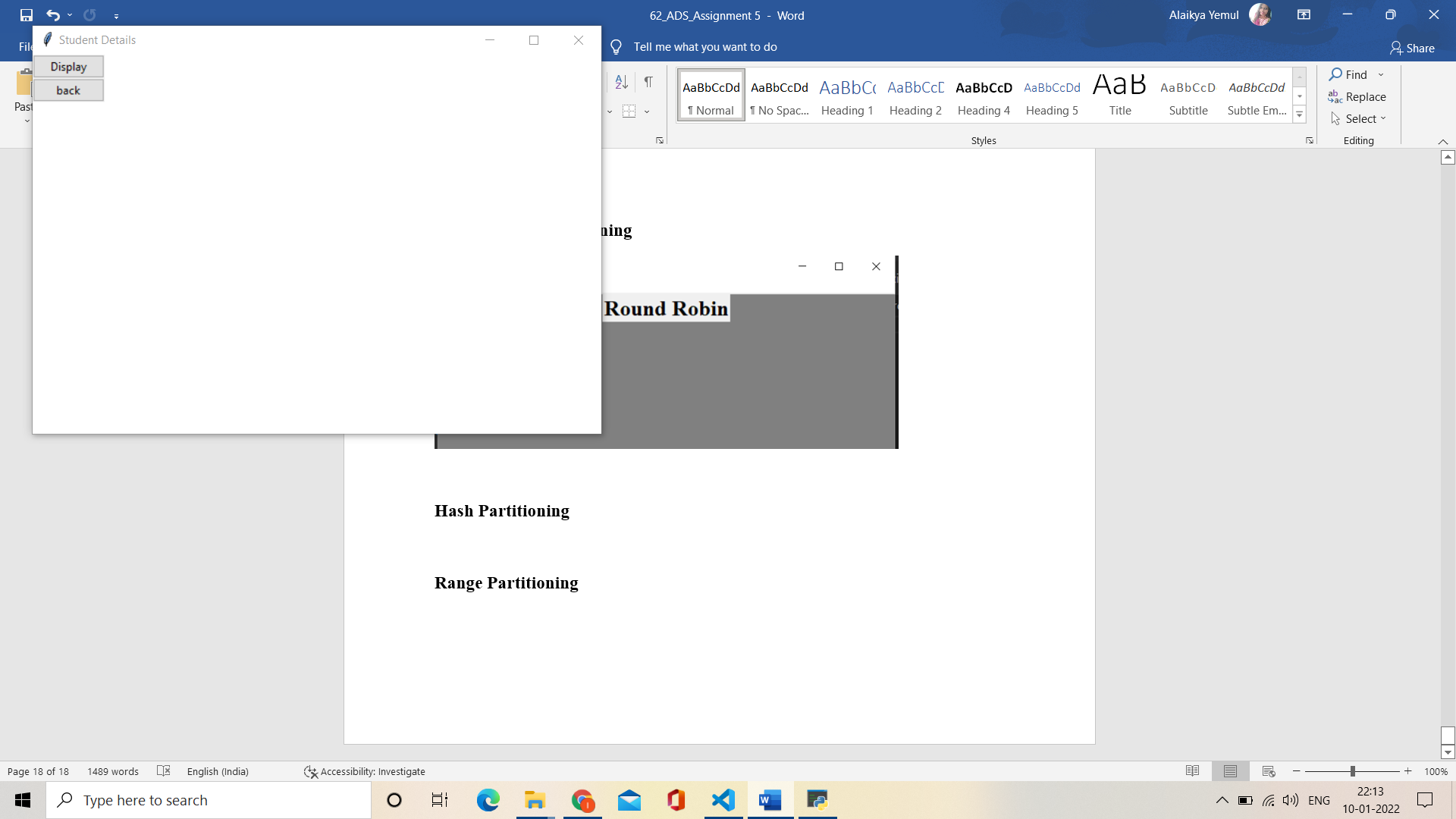
root.config(menu=menubar)

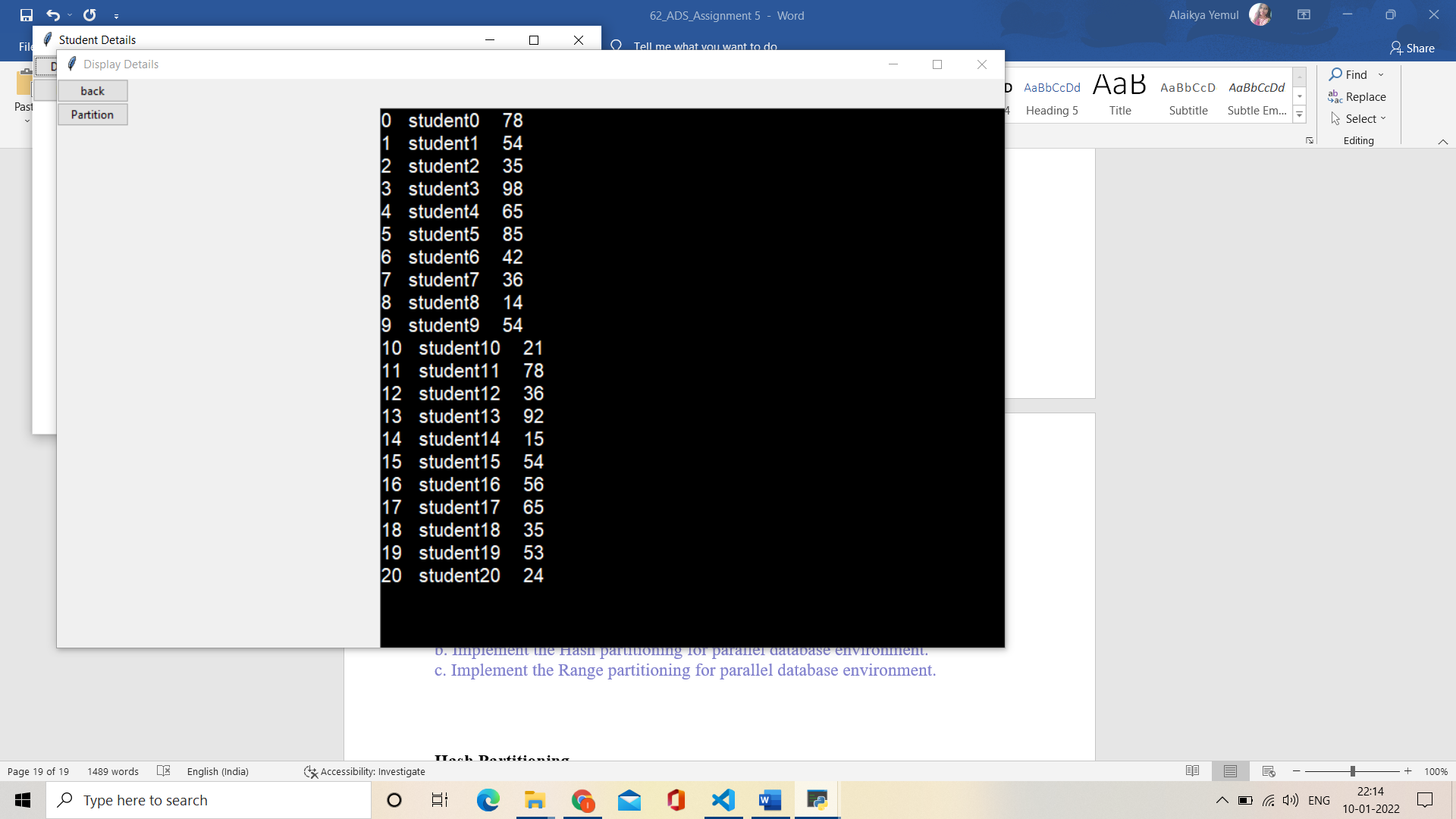
root.mainloop()

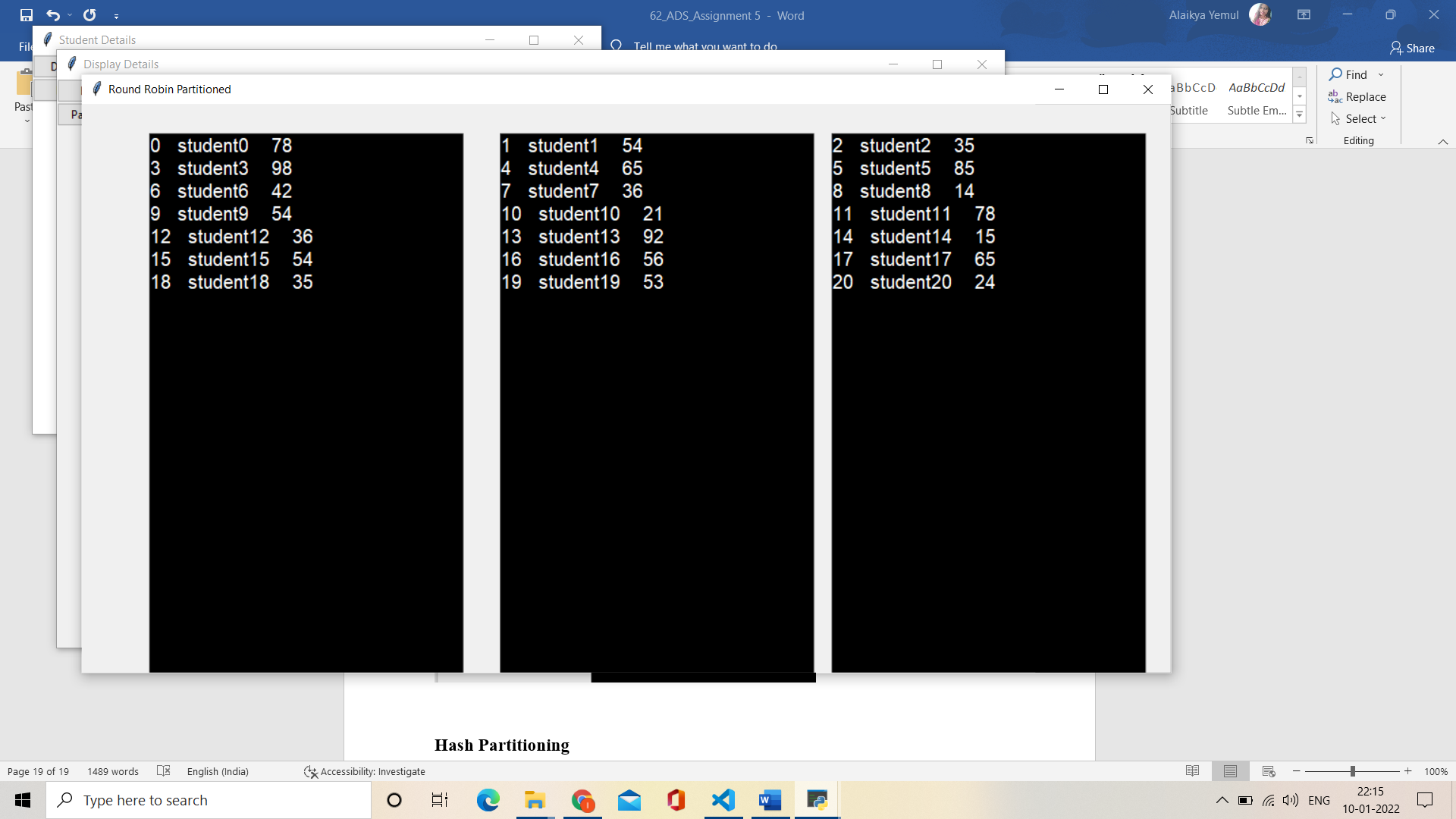
Screenshots :

**Round Robin Partitioning**

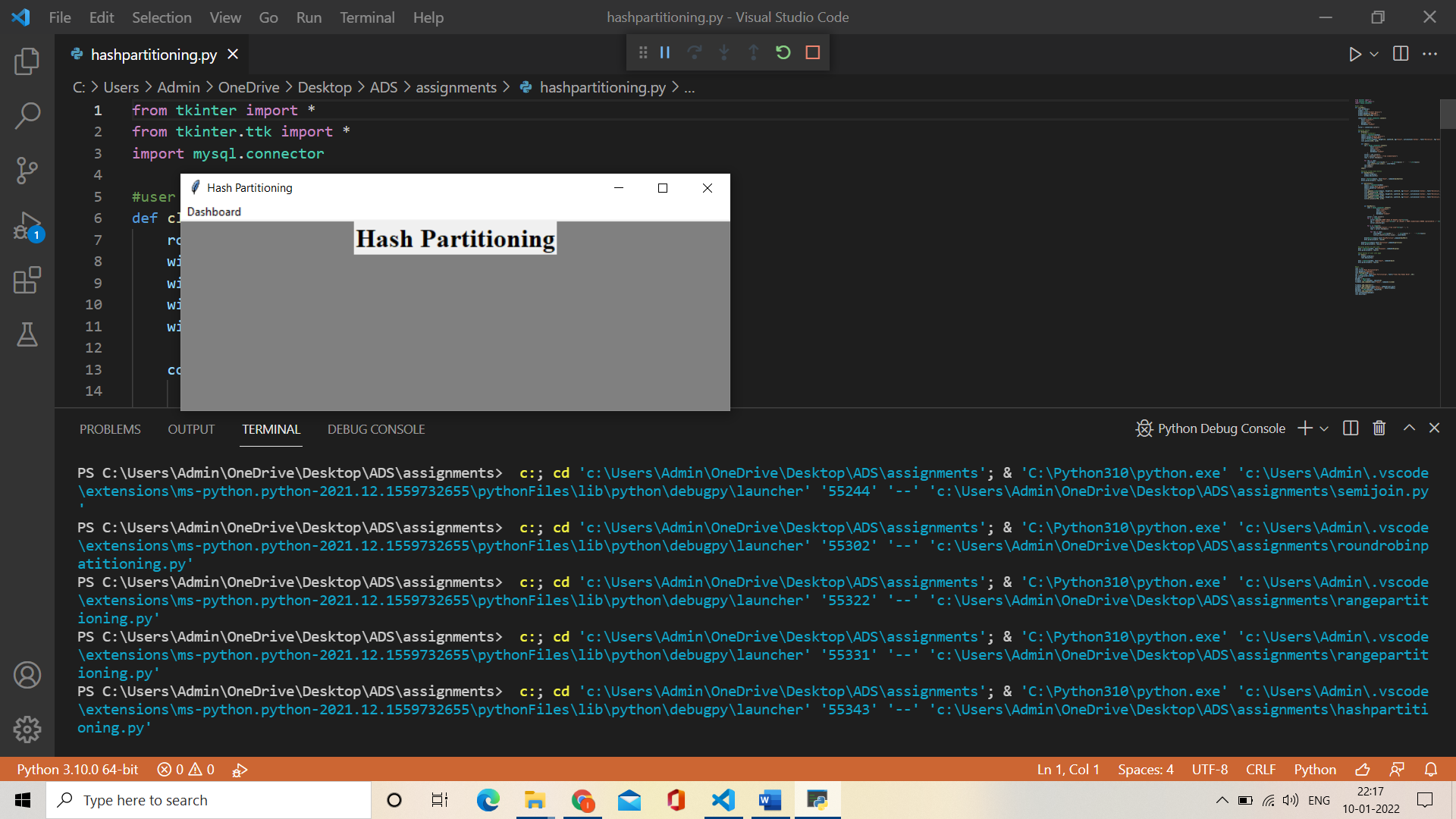


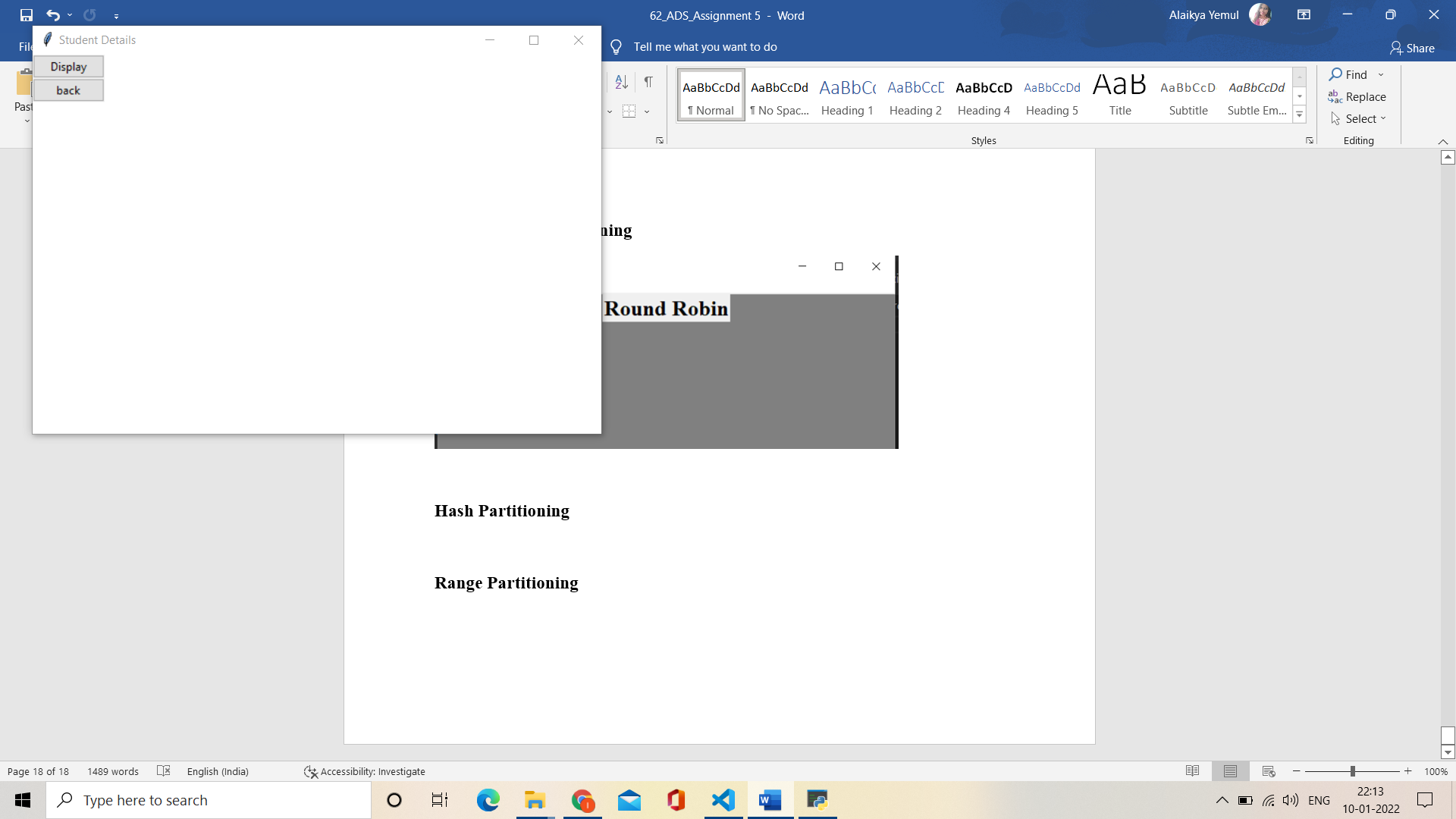


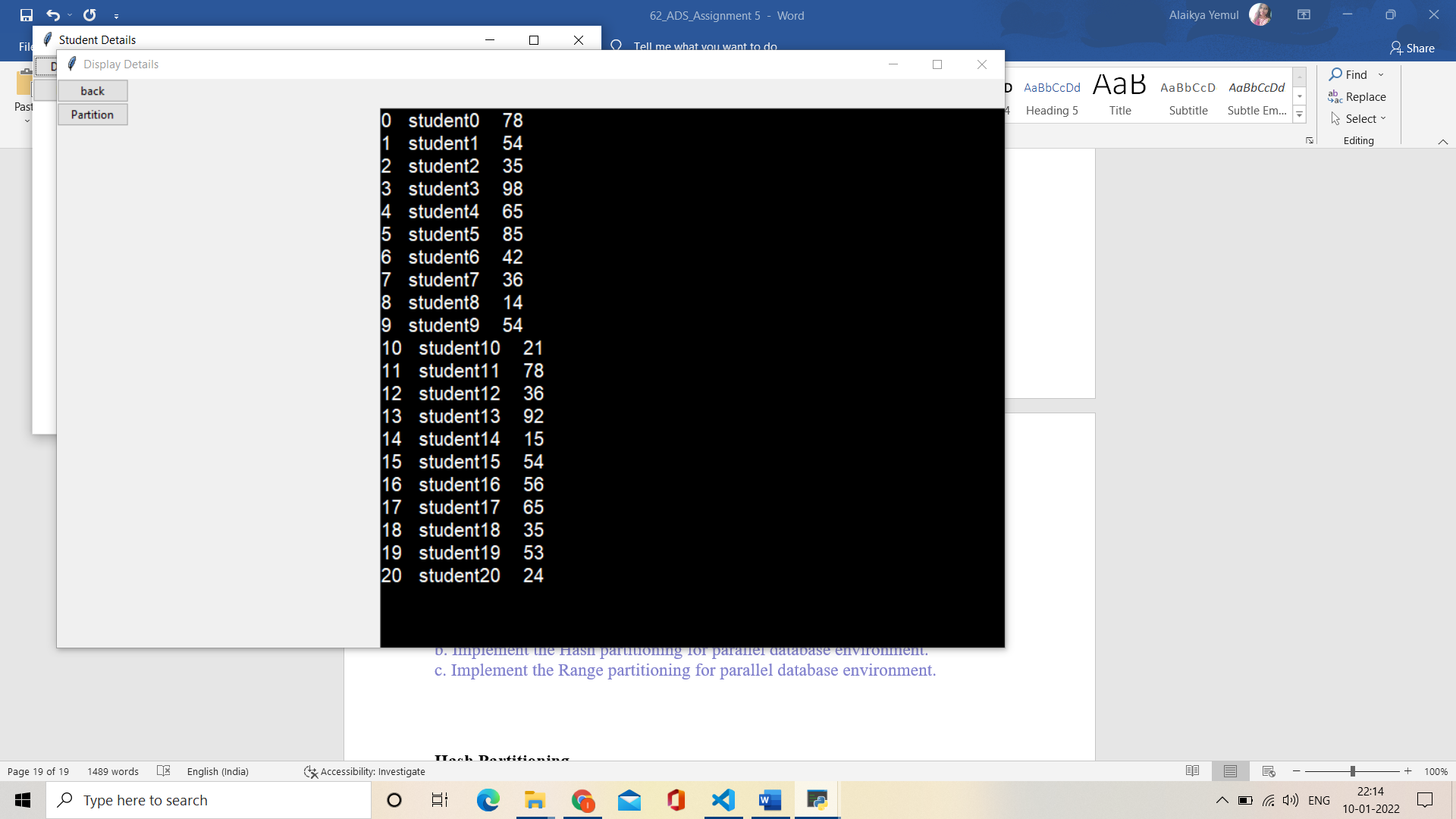


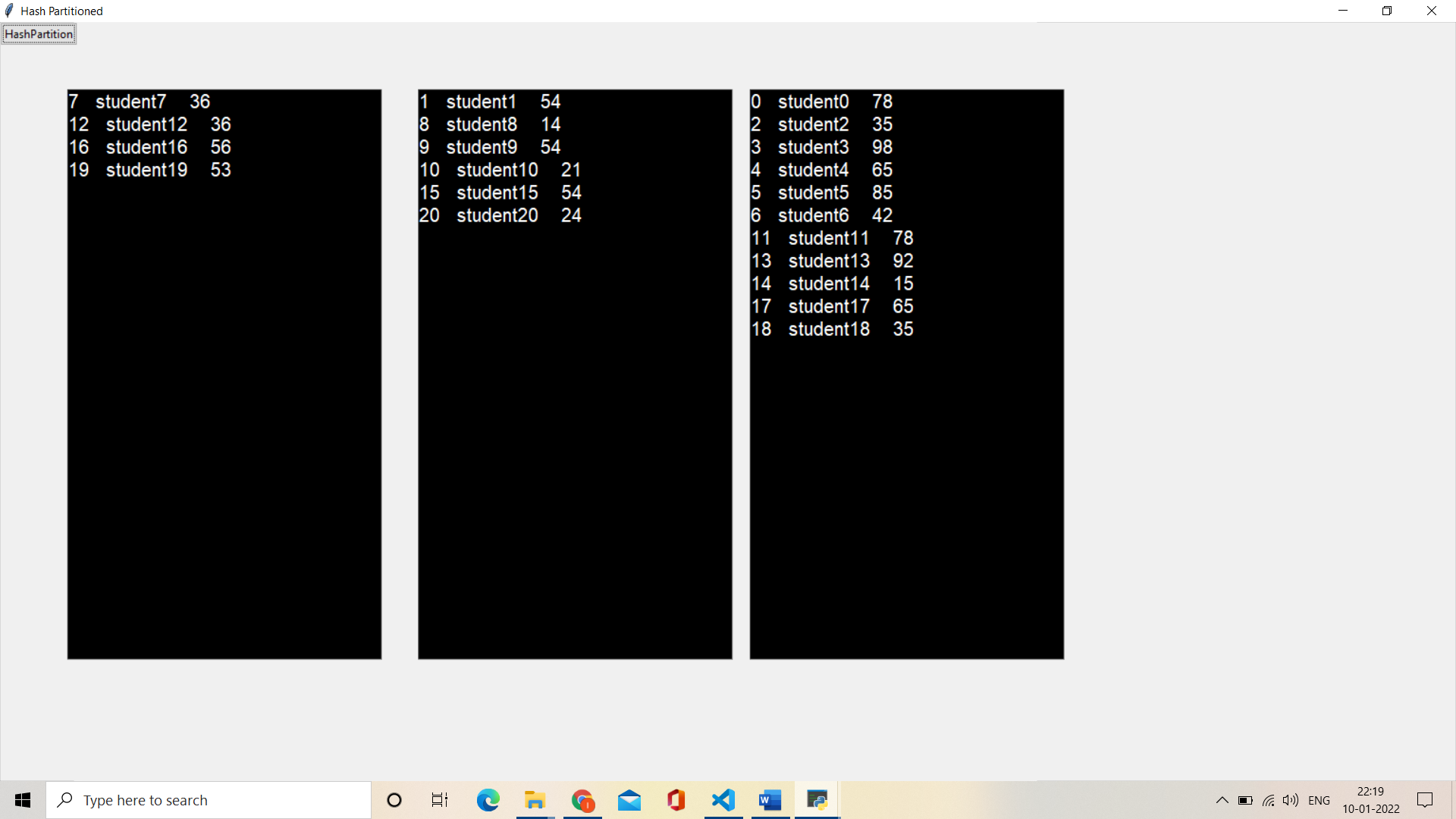


**Hash Partitioning**









**Range Partitioning**

