

**DELHI PUBLIC SCHOOL**

**RAJNAGAR**

**COMPUTER SCIENCE**

**PROGRAMMING FILE**

**MADE BY:-**

**GAURANSH GOEL**

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **DATE** | **PROGRAMS** | **PAGE** | **SIGNATURE** |
| **1)** |  | **WRITE A FUNCTION TO FIND THE**  **FACTORIAL OF A NATURAL NUMBER.** |  |  |
| **2)** |  | **WRITE A FUNCTION TO FIND HCF OR GCD.** |  |  |
| **3)** |  | **WRITE A PYTHON PROGRAM TO GENERATE A FIBONACCI SEQUENCE UP TO N TERMS.** |  |  |
| **4)** |  | **WRITE A PROGRAM TO CALCULATE:**  **Area of circle[A=πr2]**  **Area of square [A=a\*a]**  **Area of rectangle[A=l\*b]** |  |  |
| **5)** |  | **WRITE A PYTHON PROGRAM TO TEST A STRING IS A PALINDROME OR NOT.** |  |  |
| **6)** |  | **WRITE A PROGRAM TO COPY ALL THE**  **LINES THAT CONTAIN THE CHARACTER `A&#39; IN A FILE AND WRITE IT TO ANOTHER FILE.** |  |  |
| **7)** |  | **WRITE A PROGRAM TO COUNT THE NUMBER OF LINES IN A TEXT FILE.** |  |  |
| **8)** |  | **WRITE A PROGRAM TO COUNT THE FREQUENCY OF WORDS IN A FILE.** |  |  |
| **9)** |  | **WRITE A PYTHON PROGRAM TO READ A RANDOM LINE FROM A FILE.** |  |  |
| **10)** |  | **WRITE A PROGRAM TO DISPLAY ALL**  **THE STRINGS STORED IN IT IN UPPER CASE AND ALSO FIND THE FILE SIZE OF A PLAIN FILE.** |  |  |
| **11)** |  | **WRITE A PROGRAM TO IMPLIMENT 2**  **USER DEFINED FUNCTIONS ONE IS TO**  **CREATE BINARY FILE AND ANOTHER IS TO READ BINARY FILE (MAKE USE OF PICKLE MODULE).** |  |  |
| **12)** |  | **WRITE A PROGRAM THAT GENERATES RANDOM NUMBERS BETWEEN 1 AND 6 (SIMULATES A DICE).** |  |  |
| **13)** |  | **WRITE A PROGRAM TO INSERT LIST DATA IN CSV FILE.** |  |  |
| **14)** |  | **WRITE A PROGRAM TO IMPLEMENT**  **SEARCHING METHODS BASED ON USER**  **CHOICE USING A LIST DATA- STRUCTURE.(LINEAR/BINARY)** |  |  |
| **15)** |  | **WRITE A PROGRAM TO IMPLEMENT A STACK USING A LIST DATA-STRUCTURE.** |  |  |
| **S.NO** | **DATE** | **PROGRAMS** | **PAGE** | **SIGNATURE** |
| **16)** |  | **WRITE A PROGRAM TO IMPLEMENT A QUEUE USING A LIST DATA-STRUCTURE.** |  |  |
| **17)** |  | **WRITE A PROGRAM TO WRITE DICTIONARY DATA IN CSV FILE.** |  |  |
| **18)** |  | **WRITE A PYTHON DATABASE**  **CONNECTIVITY SCRIPT THAT DELETES**  **RECORDS FROM CATEGORY TABLE OF DATABASE ITEMS THAT HAVE NAME=’Stockable’.** |  |  |
| **19)** |  | **WRITE A PROGRAM TO INTEGRATE SQL WITH PYTHON BY IMPORTING THE MYSQL MODULE.** |  |  |
| **20)** |  | **WRITE A PROGRAM TO CONNECT**  **PYTHON WITH MYSQL USING**  **DATABASE CONNECTIVITY AND**  **PERFORM THE FOLLOWING**  **OPERATIONS ON DATA IN DATABASE:**  **FETCH, UPDATE AND DELETE THE DATA.** |  |  |

**WRITE A FUNCTION TO FIND THE FACTORIAL OF A NATURAL NUMBER.**

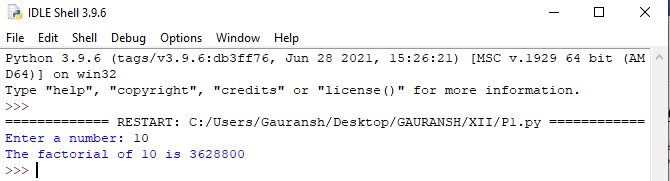
**CODE:**

*num = int(input("Enter a number: ")) factorial = 1 if num < 0:*

*print(" Factorial does not exist for negative numbers") elif num == 0:*

*print("The factorial of 0 is 1") else: for i in range(1,num + 1): factorial = factorial\*i print("The factorial of",num,"is",factorial)*

**OUTPUT:**



**WRITE A FUNCTION TO FIND HCF OR GCD.**

**CODE:**

*#define a function*

*#def compute\_hcf(x, y): # choose the smaller number*

*if x > y:*

*smaller = y else:*

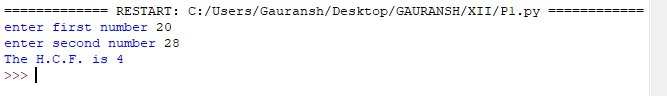
*smaller = x for i in range(1, smaller+1): if((x % i == 0) and (y % i == 0)):*

*hcf = i*

*return hcf*

*num1 = int(input("enter first number")) num2 = int(input("enter second number")) print("The H.C.F. is", compute\_hcf(num1, num2))*

**OUTPUT:**



**WRITE A PYTHON PROGRAM TO GENERATE FIBONACCI SEQUENCE UP TO N TERMS.**

**CODE:**

*nterms = int(input("How many terms? "))*

*n1, n2 = 0, 1 count = 0 if nterms <= 0:*

*print("Please enter a positive integer") elif nterms == 1:*

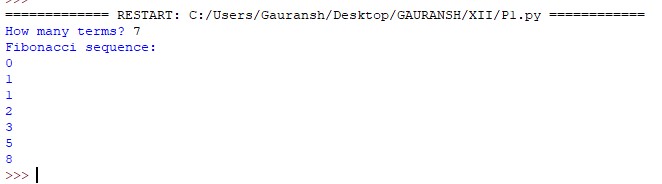
*print("Fibonacci sequence upto",nterms,":")*

*print(n1) else:*

*print("Fibonacci sequence:") while count < nterms:*

*print(n1) nth = n1 + n2 n1 = n2 n2 = nth count += 1*

**OUTPUT:**



# PROGRAM 4

**WAP TO CALCULATE: Area of circle[A=πr2] Area of square [A=a\*a] Area ofrectangle[A=l\*b] CODE:**

*# define a function for calculating*

*# the area of a shapes def calculate\_area(name):\*

*# converting all characters # into lower cases name = name.lower() # check for the conditions if name == "rectangle":*

*l = int(input("Enter rectangle's length: ")) b = int(input("Enter rectangle's breadth: ")) rect\_area = l \* b print("The area of rectangle is",{rect\_area}) elif name == "square":*

*s = int(input("Enter square's side length: ")) sqt\_area = s \* s print(f"The area of square is",{sqt\_area}) elif name == "circle":*

*r = int(input("Enter circle's radius length: ")) pi = 3.14 circ\_area = pi \* r \* r print(f"The area of triangle is,",{circ\_area})*

*else:*

*print("Sorry! This shape is not available")*

*# driver code*

*if \_\_name\_\_ == "\_\_main\_\_" : print("Calculate Shape Area") shape\_name = input("Enter the name of shape whose area you want to find: ")*

*# function calling*

*calculate\_area(shape\_name)*

**OUTPUT:**



**WRITE A PYTHON PROGRAM TO TEST A STRING IS A PALINDROME OR NOT.**

**CODE:**

*my\_str = input("enter a string:")*

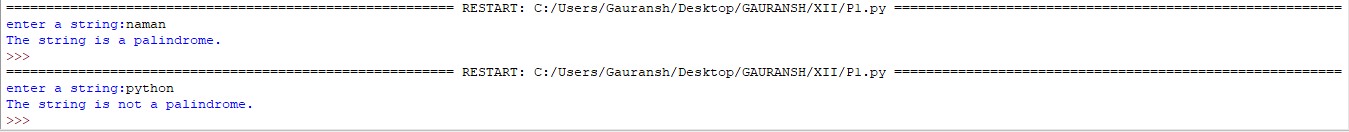
*# make it suitable for caseless comparison my\_str = my\_str.casefold() # reverse the string rev\_str = reversed(my\_str)*

*# check if the string is equal to its reverse if list(my\_str) == list(rev\_str):*

*print("The string is a palindrome.") else:*

*print("The string is not a palindrome.")*

**OUTPUT:**



**WRITE A PROGRAM TO COPY ALL THE LINES THAT CONTAIN**

**THE CHARACTER `A’ IN A FILE AND WRITE IT TO ANOTHER**

**FILE.**

**CODE:**

*fin=open("D:\\book.txt","r") fout=open("D:\\story.txt","w") s=fin.readlines()*

*for j in s: if 'a' in j:*

*fout.write(j)*

*fout.close() fin.close()*

INPUT:



**OUTPUT:**



**WRITE A PROGRAM TO COUNT THE NUMBER OF LINES IN A TEXT**

**FILE:**

**CODE:**

*S A M P L E . T X T*

*Hello World*

*Hello Again*

*Goodbye*

*file = open("sample.txt", "r")*

*number\_of\_lines = 0*

*for line in file: line = line.strip("\n") number\_of\_lines += 1 print(“lines:”, number\_of\_lines)*

**OUTPUT:**



**WRITE A PROGRAM TO COUNT THE FREQUENCY OF WORDS IN A**

**FILE.**

**CODE:**

*SAMPLE TEST.TXT*

*VTU BGM*

*VTU BELGAVI*

*VTU BGM Karnataka*

*#Read file name and count how many times each word appears*

*fname = input('Enter the file name: ')*

*try:*

*fhand = open(fname)*

*counts = dict() for line in fhand:*

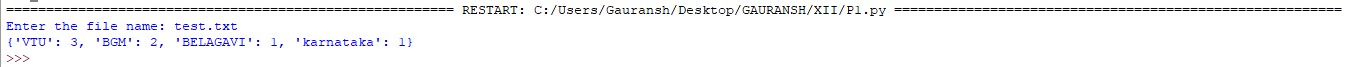
*words = line.split() for word in words: if word in counts: counts[word] += 1*

*else:*

*counts[word] = 1 print(counts) except:*

*print('File cannot be opened:', fname)*

**OUTPUT:**



**WRITE A PYTHON PROGRAM TO READ A RANDOM LINE FROM A FILE.**

**CODE:**

*SAMPLE FILE1.TXT*

*HELLO HI*

*I AM STUDENT*

*I LOVE PYTHON*

*import random def randnum(fname):*

*lines=open(fname).read().splitlines() print(lines)*

*return random.choice(lines) print(randnum('file1.txt'))*

**OUTPUT:**

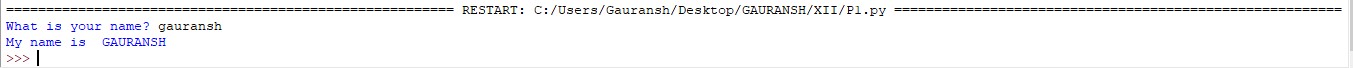


**WRITE A PROGRAM TO DISPLAY ALL THE STRINGS STORED IN IT IN UPPER CASE AND ALSO FIND THE FILE SIZE OF A PLAIN FILE.**

**CODE (STRING STORED IN UPPER CASE):**

*user\_input = input("What is your name? ") print("My name is ", user\_input.upper())*

OUTPUT:



CODE (SIZE OF A PLANE FILE):

*def file\_size(fname):*

*import os statinfo = os.stat(fname) return statinfo.st\_size print("File size in bytes of a plain file: ",file\_size("test.txt"))*

OUTPUT:



**WRITE A PROGRAM TO IMPLEMENT 2 USER DEFINED FUNCTIONS ONE IS TO CREATE BINARY FILE AND ANOTHER IS TO READ BINARY FILE (MAKE USE OF PICKLE MODULE).**

**CODE:**

*import pickle*

*#dictionary objects that are stored in the binary file Emp.dat emp1={'Empno':1201,'Name':'Anushree','Age': 25,'Salary': 47000} emp2={'Empno':1211,'Name':'Zoya','Age': 30,'Salary': 48000} emp3={'Empno':1251,'Name':'Simarjeet','Age': 27,'Salary': 49000} emp4={'Empno':1266,'Name':'Alex','Age': 29,'Salary': 50000}*

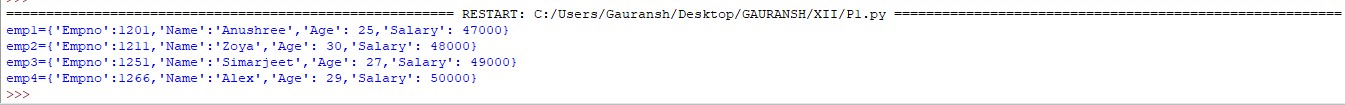
*#openfile in read mode empfile=open('Emp.dat','rb')*

*#read from a file try: while True: emp=pickle.load(empfile) print(emp) except EOFError:*

*empfile.close(*

*)*

OUTPUT:



WRITE A PROGRAM THAT GENERATES RANDOM NUMBERS BETWEEN 1 AND 6 (SIMULATES A DICE).

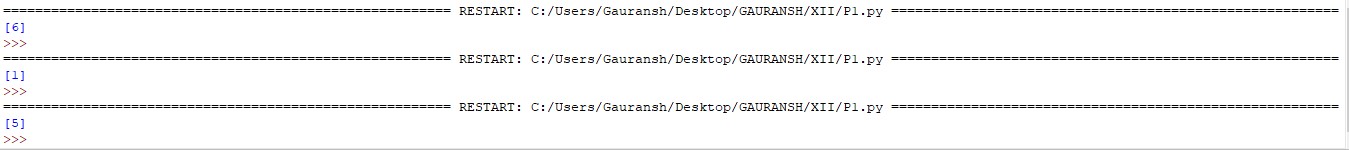
**CODE:**

*import random def rolladice(): counter=0 myList=[] while(counter)<6:*

*randomNumber = random.randint(1,6) myList.append(randomNumber) counter = counter+1 if (counter)>6:*

*pass else: return myList print(rolladice())*

**OUTPUT:**



**WRITE A PROGRAM TO INSERT LIST DATA IN CSV FILE.**

**CODE:**

*import csv*

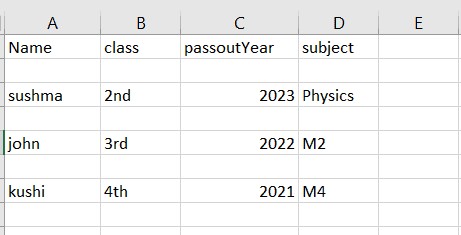
*Details = ['Name', 'class', 'passoutYear', 'subject']*

*rows = [ ['sushma', '2nd', '2023', 'Physics'], ['john', '3rd', '2022', 'M2'], ['kushi',*

*'4th', '2021', 'M4']] with open('student.csv', 'w') as f:*

*write = csv.writer(f) write.writerow(Details) write.writerows(rows)*

**OUTPUT:**



**WRITE A PROGRAM TO IMPLEMENT SEARCHING METHODS BASED ON USER CHOICE USING A LIST DATA-STRUCTURE(LINEAR/BINARY).**

**CODE:**

*input1=input("FOR LINEAR SEARCH TYPE LINEAR")*

*if input1=="LINEAR":*

*def search(arr, n, x):*

*for i in range(0, n):*

*if (arr[i] == x):*

*return i*

*return -1*

*# Driver Code arr = [2, 3, 4, 10, 40]*

*x = 10*

*n = len(arr) # Function call result = search(arr, n, x) if(result == -1):*

*print("Element is not present in array") else:*

*print("Element is present at index", result)*

*input2=input("FOR BINARY SEARCH TYPE BINARY")*

*if input2=="BINARY":*

*def binarySearch (arr, l, r, x):*

*# Check base case if r >= l:*

*mid = l + (r - l) // 2*

*# If element is present at the middle itself*

*if arr[mid] == x: return mid*

*# If element is smaller than mid, then it # can only be present in left subarray elif arr[mid] > x:*

*return binarySearch(arr, l, mid-1, x) else:*

*return binarySearch(arr, mid + 1, r, x)*

*else:*

*# Element is not present in the array return -1*

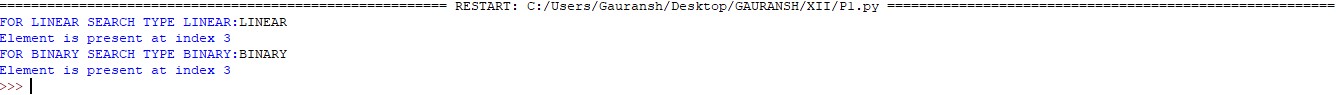
*# Driver Code arr = [ 2, 3, 4, 10, 40 ] x = 10*

*# Function call result = binarySearch(arr, 0, len(arr)-1, x) if result != -1:*

*print ("Element is present at index % d" % result) else:*

*print ("Element is not present in array")*

**OUTPUT:**



WRITE A PROGRAM TO IMPLEMENT A STACK USING A LIST DATA-STRUCTURE.

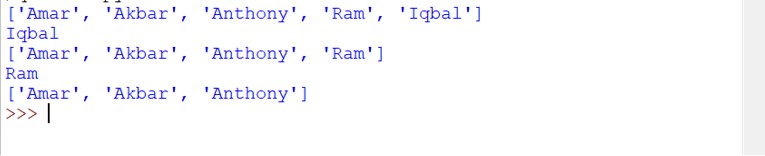
**CODE:**

stack = ["Amar", "Akbar", "Anthony"] stack.append("Ram") stack.append("Iqbal") print(stack)

# Removes the last item print(stack.pop()) print(stack)

# Removes the last item print(stack.pop()) print(stack)

**OUTPUT:**



**WRITE A PROGRAM TO IMPLEMENT A QUEUE USING A LIST DATA-STRUCTURE.**

**CODE:**

q=[]

q.append(10)

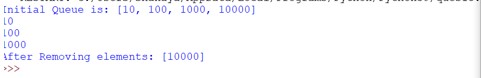
q.append(100)

q.append(1000)

q.append(10000)

print("Initial Queue is:",q) print(q.pop(0)) print(q.pop(0)) print(q.pop(0)) print("After Removing elements:",q)

**OUTPUT:**



WRITE A PROGRAM TO WRITE DICTIONARY DATA IN CSV FILE.

CODE:

*import csv*

*csv\_columns = ['id','Column1', 'Column2', 'Column3', 'Column4', 'Column5'] dict\_data = {'id':['1', '2', '3'], 'Column1':[33, 25, 56],*

*'Column2':[35, 30, 30],*

*'Column3':[21, 40, 55],*

*'Column4':[71, 25, 55], 'Column5':[10, 10, 40], } csv\_file = "temp.csv" try: with open(csv\_file, 'w') as csvfile:*

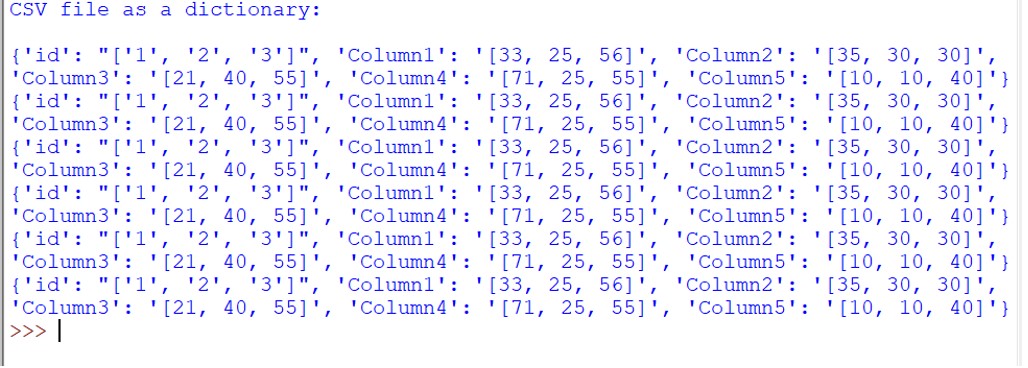
*writer = csv.DictWriter(csvfile, fieldnames=csv\_columns) writer.writeheader() for data in dict\_data:*

*writer.writerow(dict\_data) except IOError: print("I/O error")*

*data = csv.DictReader(open(csv\_file)) print("CSV file as a dictionary:\n") for row in data:*

*print(row)*

OUTPUT:



# PROGRAM 18

WRITE A PYTHON DATABASE CONNECTIVITY SCRIPT THAT DELETES RECORDS FROM CATEGORY TABLE OF DATABASE ITEMS THAT HAVE NAME=’Stockable’.

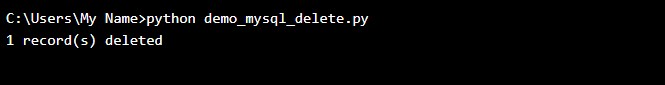
CODE:

*import mysql.connector mydb = mysql.connector.connect( host="localhost", user="myusername", password="mypassword", database="mydatabase") mycursor = mydb.cursor(*

*sql = "DELETE FROM customers WHERE address = 'Mountain 21'" mycursor.execute(sql) mydb.commit()*

*print(mycursor.rowcount, "record(s) deleted")*

OUTPUT:



# PROGRAM 19

WRITE A PROGRAM TO INTEGRATE SQL WITH PYTHON BY IMPORTING THE MYSQL MODULE.

**CODE:**

*import mysql.connector from mysql.connector import Error*

*try:*

*connection = mysql.connector.connect(host='localhost', database='Electronics',*

*user='pynative', password='pynative@#29')*

*if connection.is\_connected():*

*db\_Info = connection.get\_server\_info() print("Connected to MySQL Server version ", db\_Info) cursor = connection.cursor() cursor.execute("select database();") record = cursor.fetchone() print("You're connected to database: ", record) except Error as e:*

*print("Error while connecting to MySQL", e) finally: if connection.is\_connected():*

*cursor.close() connection.close() print("MySQL connection is closed")*

**OUTPUT:**



## PROGRAM 20

WRITE A PROGRAM TO CONNECT PYTHON WITH MYSQL USING DATABASE CONNECTIVITY AND PERFORM THE FOLLOWING OPERATIONS ON DATA IN DATABASE: FETCH, UPDATE AND DELETE THE DATA**.**

**1)TO FETCH DATA:**

*import MySQLdb*

*# Open database connection db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )*

*# prepare a cursor object using cursor() method cursor = db.cursor()*

*sql = "SELECT \* FROM EMPLOYEE \ WHERE INCOME > '%d'" % (1000)*

*try:*

*# Execute the SQL command*

*cursor.execute(sql)*

*# Fetch all the rows in a list of lists.*

*results = cursor.fetchall() for row in results: fname = row[0] lname = row[1] age = row[2] sex = row[3] income = row[4]*

*# Now print fetched result*

*print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" % \*

*(fname, lname, age, sex, income ) except:*

*print "Error: unable to fecth data"*

*# disconnect from server db.close()*

**OUTPUT:**



2**)TO UPDATE THE DATA:**

*The following procedure updates all the records having SEX as* ***'M'****.*

*import MySQLdb*

*# Open database connection db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )*

*# prepare a cursor object using cursor() method cursor = db.cursor()*

*# Prepare SQL query to UPDATE required records sql = "UPDATE EMPLOYEE SET AGE = AGE + 1*

*WHERE SEX = '%c'" % ('M') try:*

*# Execute the SQL command*

*cursor.execute(sql)*

*# Commit your changes in the database db.commit() except:*

*# Rollback in case there is any error db.rollback()*

*# disconnect from server db.close()*

**3)TO DELETE THE DATA:**

*Following is the procedure to delete all the records from EMPLOYEE where AGE is more than 20 − import MySQLdb*

*# Open database connection db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )*

*# prepare a cursor object using cursor() method cursor = db.cursor()*

*# Prepare SQL query to DELETE required records sql = "DELETE FROM EMPLOYEE WHERE AGE > '%d'" % (20)*

*try:*

*# Execute the SQL command*

*cursor.execute(sql)*

*# Commit your changes in the database db.commit() except:*

*# Rollback in case there is any error db.rollback()*

*# disconnect from server db.close()*

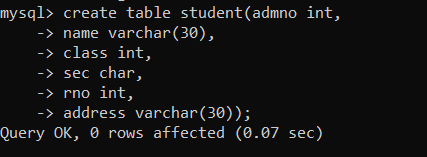
*1.Command for creating a database:*

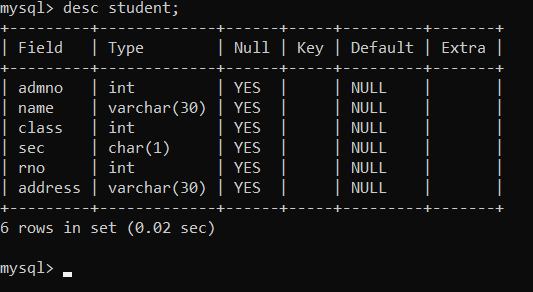


*2.Command for using a database:*

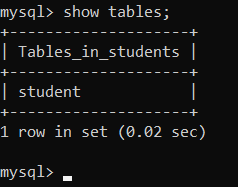
**

*3.Command for creating a table:*

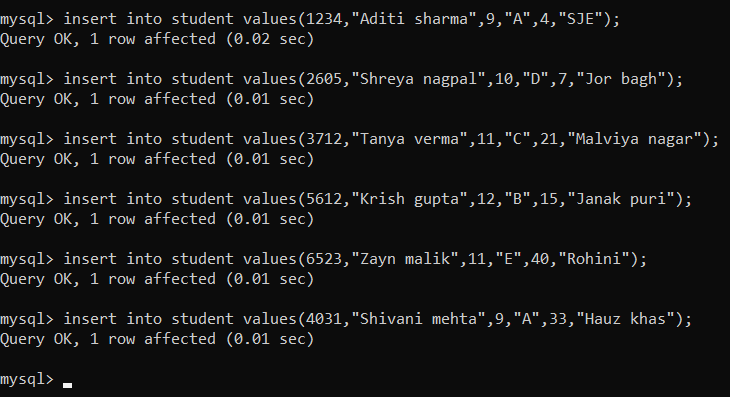
**

*4.Command for showing the structure of table:*

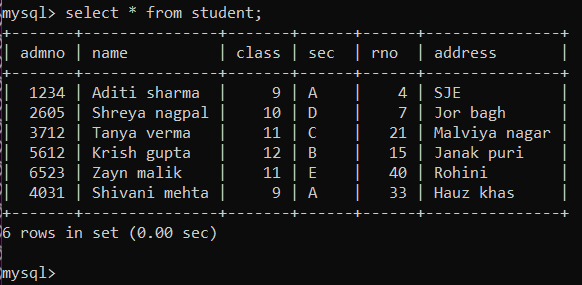
*5.Command to show tables in database:*

**

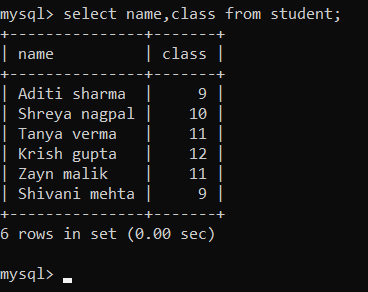
*6.Command for inserting data into table:*

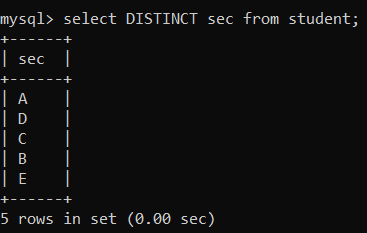
**

*7.Command to view the contents of table:*

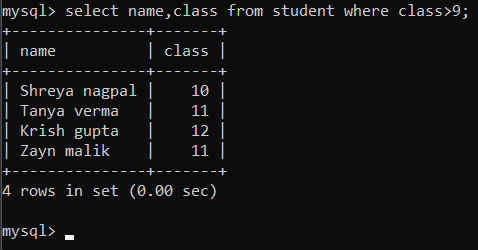
**

*8.Command to retrieve data;*

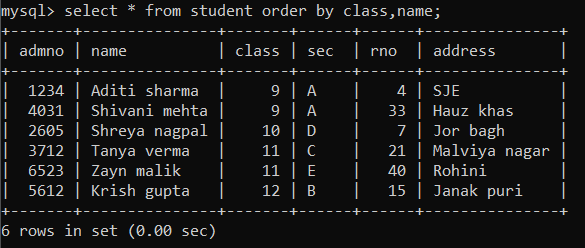
**

*9.Command for using keyword DISTINCT:*

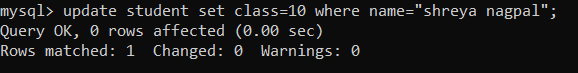
*10.Command for using WHERE clause:*

**

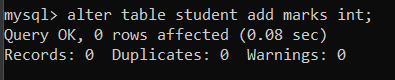
*11.Command for using ORDER BY clause:*

**

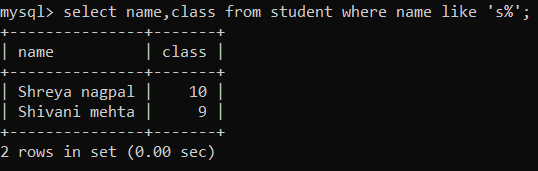
*12.Command for using UPDATE:*

**

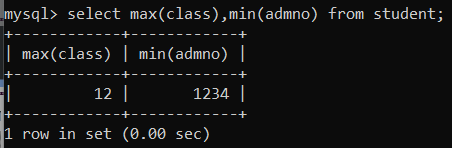
*13.Command for using ALTER (to modify structure of table):*

**

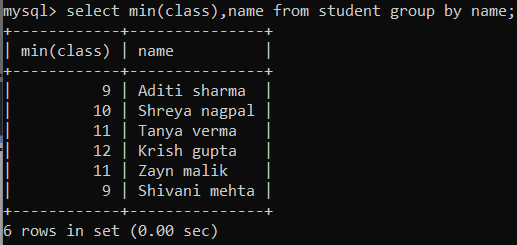
*14.Command for using LIKE operator:*

**

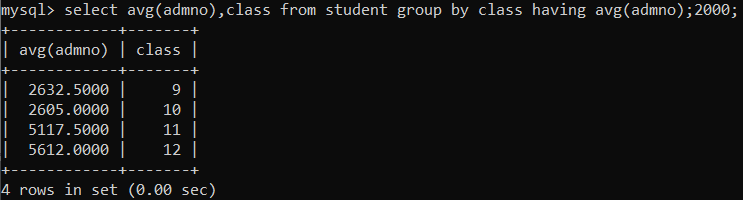
*15.Command for using aggregate functions:*

**

*16.Command for using GROUP BY:*

**

*17.Command for using HAVING clause:*

**

*18.Command for using group by with order by:*

*Text

Description automatically generated*

*19.Command for using group by and having clause with where clause:*

*A picture containing graphical user interface

Description automatically generated*

*20.Command for using where clause and group by:*

*Text

Description automatically generated*

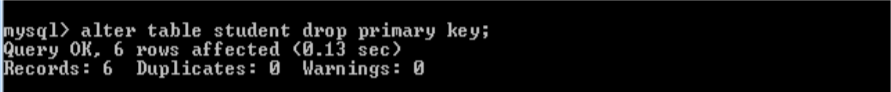
*21.Command for adding primary key;*

**

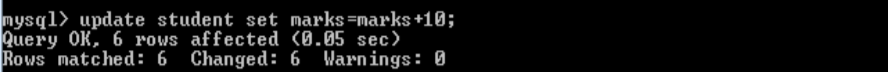
*22.Command for deleting a column:*

**

*23.Command to remove primary key:*

**

*24.Command to increase marks:*

**

*25.Command to delete table:*

**