Report File

By Mayank Anand

Grade: XII-A

Submitted to: Ms. Kajal

S.NO	Name of Practical
1	Write a python program to input any number from the user and calculate factorial of a number.
2	Write a python program to read a line and calculate the total number of uppercase letters, lowercase letters, spaces, and digits in the given string.
3	Write a python program to search for a given word in a string.
4	Write a python program to input integer values in a list and calculate the sum of them.
5	Write a python program to input a list of elements and search the given element in the list.
6	Write a python program to input integer values in a tuple and find the largest and smallest number.
7	Write a python program to create a dictionary with roll number, name and marks of n students in a class and display the name of the students who have scored marks above 85.
8	Write a python program to perform the basic arithmetic operations in a menu driven program using functions.
9	Write a python program to read a text file line by line and display each word separated by a #.
10	Write a python program to read a text file and display the number of vowels, consonants, uppercase, lowercase characters in a file.
11	Write a python program to remove all the lines that contain the character 'a' in a file and write it to another file.
12	Write a python program to create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message
13	Write a python program to create a binary file with roll number, name and marks. Input a roll number and update the marks.
14	Write a python program to create a random number generator that generates random number between 1 and 6(simulates a dice).

<table>
<tr><td>15</td><td>Write a python program to create a CSV file by entering user-id and password, read and search the password for given user-id.</td></tr>
<tr><td>16</td><td>Write a python program to create a CSV file and store employee number, name and salary. Search employee by employee number and display name and salary, and if not found display an appropriate message.</td></tr>
<tr><td>17</td><td>Write a python program to implement a stack using a list.</td></tr>
<tr><td>18</td><td>Write a query to create a database School and execute following commands:
i. To show database
ii. To use database</td></tr>
<tr><td>19</td><td>Write a query to create Student table using following structure :
<table>
<tr><th>Field Name</th><th>Data Type</th><th></th></tr>
<tr><td>Roll_no</td><td>Char(6)</td><td>Primary Key</td></tr>
<tr><td>Name</td><td>Varchar(20)</td><td>Not Null</td></tr>
<tr><td>Stream</td><td>Varchar(15)</td><td></td></tr>
<tr><td>DOB</td><td>Date</td><td></td></tr>
<tr><td>Marks</td><td>Number</td><td></td></tr>
</table>
</td></tr>
<tr><td>20</td><td>Write a query to perform following operations on table student:
i. Insert data in table student
ii. Display roll_no, name from table student
iii. Display name of student having marks<85 and stream=’medical’</td></tr>
<tr><td>21</td><td>Write a query to ALTER table to add a new attribute/modify data type and drop attribute.</td></tr>
<tr><td>22</td><td>Write a query to UPDATE Student table to modify data.</td></tr>
<tr><td>23</td><td>Write a query to DISPLAY data in ascending/descending order.</td></tr>
<tr><td>24</td><td>Write a query to DELETE the record of students where marks<35.</td></tr>
<tr><td>25</td><td>Write a query to GROUP BY and find min, max, sum, count and average of marks.</td></tr>
</table>

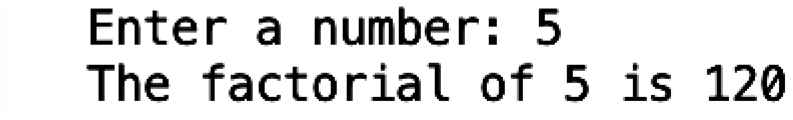
26	Write a query to join two different tables.
27	Write a python program to connect with database and store record of employee and display records.
28	Write a python program to connect with database and search employee number in the table employee and display record, if employee is not found display appropriate message.
29	Write a python program to connect with database and update the record of the entered employee number.
30	Write a python program to connect with database and delete the record of the entered employee number.

1. Write a python program to input any number from the user and calculate factorial of a number.

Code:

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
num = int(input("Enter a number: "))
if num < 0:
    print("Factorial is not defined for negative numbers.")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    result = factorial(num)
    print(f"The factorial of {num} is {result}")
```

Output:



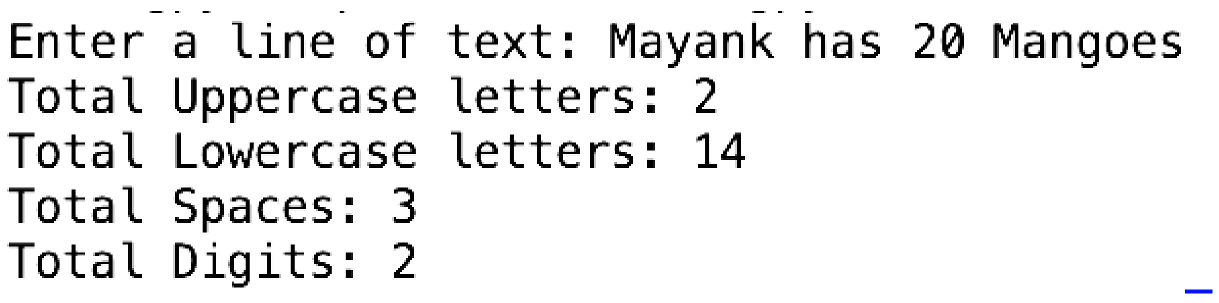
```
Enter a number: 5
The factorial of 5 is 120
```

2. Write a python program to read a line and calculate the total number of uppercase letters, lowercase letters, spaces, and digits in the given string.

Code:

```python
def count_characters(line):
    upper_count = 0
    lower_count = 0
    space_count = 0
    digit_count = 0
    for char in line:
        if 'A' <= char <= 'Z':
            upper_count += 1
        elif 'a' <= char <= 'z':
            lower_count += 1
        elif char == ' ':
            space_count += 1
        elif '0' <= char <= '9':
            digit_count += 1
    return upper_count, lower_count, space_count, digit_count
input_line = input("Enter a line of text: ")
uppercase, lowercase, spaces, digits = count_characters(input_line)
print("Total Uppercase letters:", uppercase)
print("Total Lowercase letters:", lowercase)
print("Total Spaces:", spaces)
print("Total Digits:", digits)
```

Output:



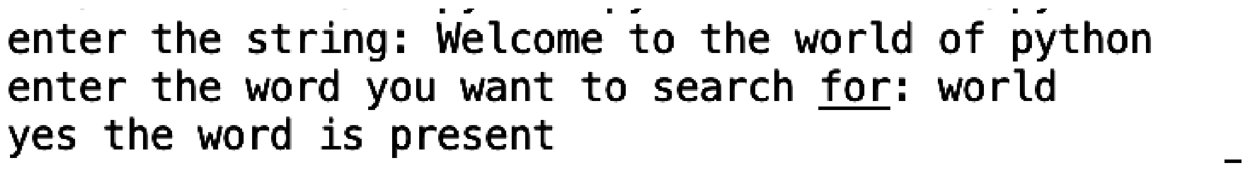
```
Enter a line of text: Mayank has 20 Mangoes
Total Uppercase letters: 2
Total Lowercase letters: 14
Total Spaces: 3
Total Digits: 2
```

3. Write a python program to search for a given word in a string.

Code:

```
string=input("enter the string: ")

n=input("enter the word you want to search for: ")

if n in string:

   print("yes the word is present")

else:

   print("no the word is not present")
```

Output:



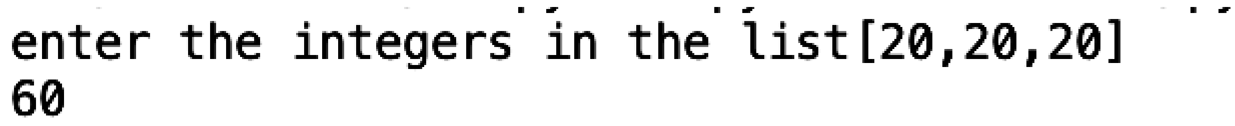
```
enter the string: Welcome to the world of python
enter the word you want to search for: world
yes the word is present
```

4. Write a python program to input integer values in a list and calculate the sum of them.

Code:

```
def suml(var):
    a=0
    for i in l:
        a=a+i
    return a
l=eval(input("enter the integers in the list"))
print(suml(l))
```

Output:



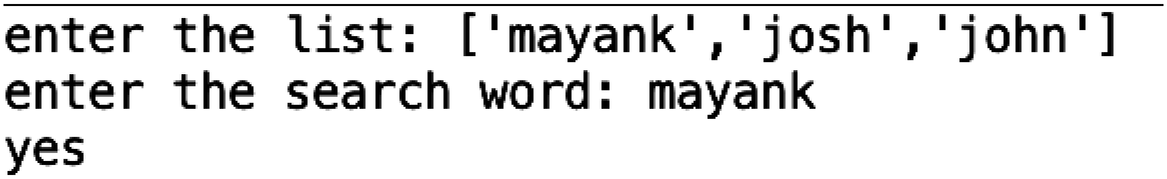
```
enter the integers in the list[20,20,20]
60
```

5. Write a python program to input a list of elements and search the given element in the list.

Code:

```
l=eval(input("enter the list: "))
n=input("enter the search word: ")
if n in l:
   print("yes")
else:
   print("no")
```

Output:



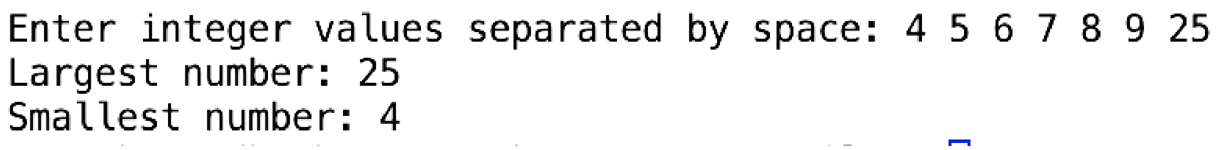
```
enter the list: ['mayank','josh','john']
enter the search word: mayank
yes
```

6. Write a python program to input integer values in a tuple and find the largest and smallest number.

Code:

```
def find_max_min(numbers):
    if len(numbers) == 0:
        return None, None

    max_num = numbers[0]
    min_num = numbers[0]

    for num in numbers:
        if num > max_num:
            max_num = num
        elif num < min_num:
            min_num = num

    return max_num, min_num

values = tuple(map(int, input("Enter integer values separated by space: ").split()))

largest, smallest = find_max_min(values)

if largest is None or smallest is None:
    print("No numbers were entered.")

else:

    print(f"Largest number: {largest}")

    print(f"Smallest number: {smallest}")
```

Output:



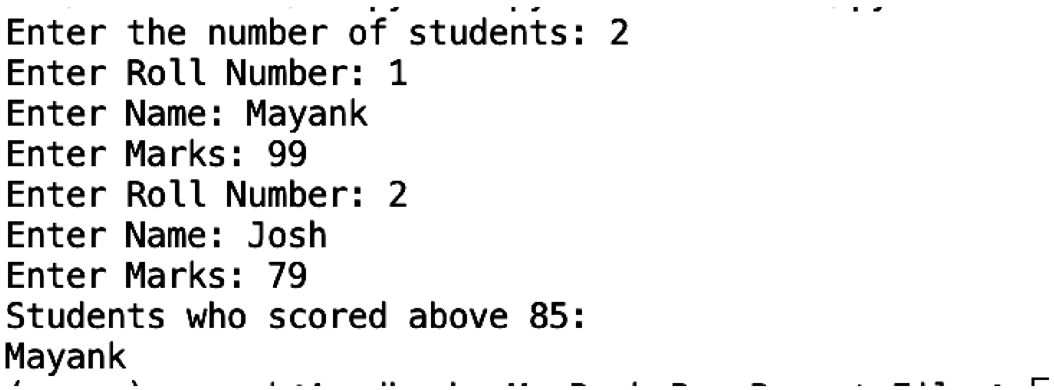
```
Enter integer values separated by space: 4 5 6 7 8 9 25
Largest number: 25
Smallest number: 4
```

7. Write a python program to create a dictionary with roll number, name and marks of n students in a class and display the name of the students who have scored marks above 85.

Code:

```
n = int(input("Enter the number of students: "))

student_data = {}

for i in range(n):

    roll_no = input("Enter Roll Number: ")

    name = input("Enter Name: ")

    marks = int(input("Enter Marks: "))

    student_data[roll_no] = {"Name": name, "Marks": marks}

print("Students who scored above 85:")

for roll_no, i in student_data.items():

    if i["Marks"] > 85:

        print(i["Name"])
```

Output:



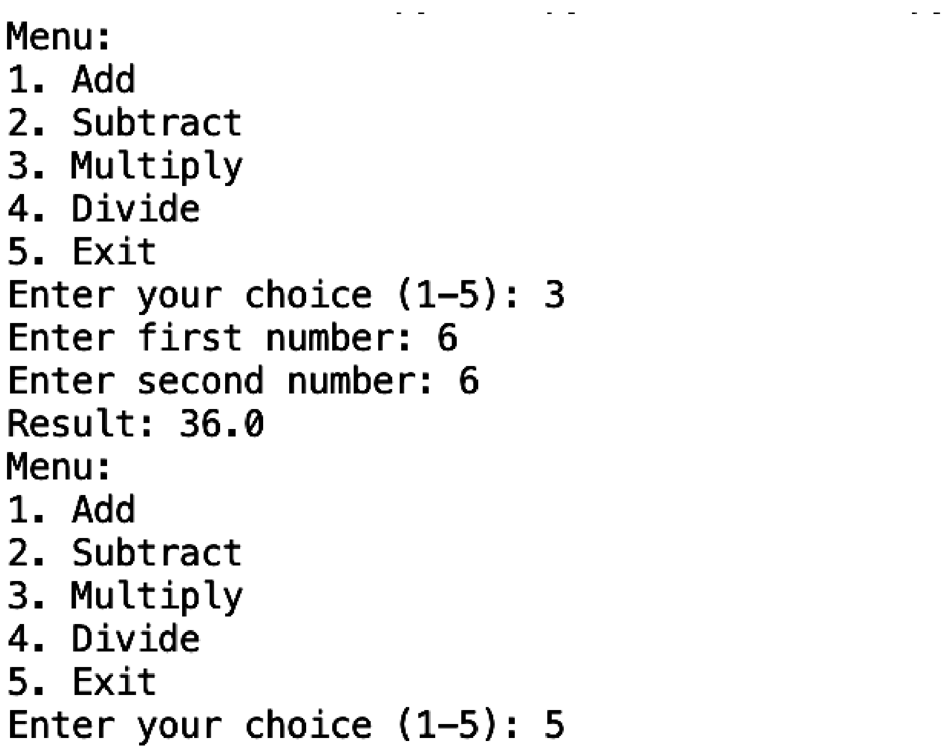
```
Enter the number of students: 2
Enter Roll Number: 1
Enter Name: Mayank
Enter Marks: 99
Enter Roll Number: 2
Enter Name: Josh
Enter Marks: 79
Students who scored above 85:
Mayank
```

8. Write a python program to perform the basic arithmetic operations in a menu driven program using functions.

Code:

```
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    if y != 0:
        return x / y
    else:
        return "Cannot divide by zero"

while True:
    print("Menu:")
    print("1. Add")
    print("2. Subtract")
    print("3. Multiply")
    print("4. Divide")
    print("5. Exit")

    choice = input("Enter your choice (1-5): ")

    if choice == "5":
        break

    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))

    if choice == "1":
        print("Result:", add(num1, num2))
    elif choice == "2":
        print("Result:", subtract(num1, num2))
    elif choice == "3":
        print("Result:", multiply(num1, num2))
    elif choice == "4":
        print("Result:", divide(num1, num2))
    else:
        print("Invalid choice")
```

Output:



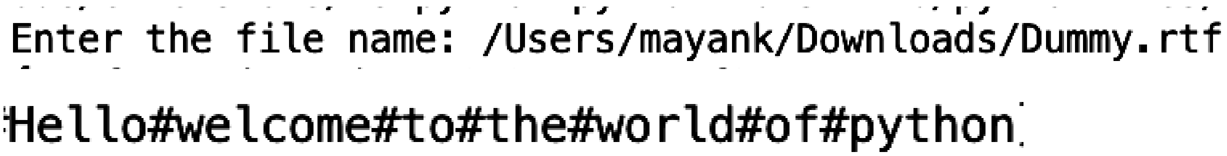
```
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice (1-5): 3
Enter first number: 6
Enter second number: 6
Result: 36.0
Menu:
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit
Enter your choice (1-5): 5
```

9. Write a python program to read a text file line by line and display each word separated by a #.

Code:

```
file_name = input("Enter the file name: ")

try:

   with open(file_name, 'r') as file:

       for line in file:

           words = line.split()

           formatted_line = "#".join(words)

           print(formatted_line)

except FileNotFoundError:

   print("File not found.")
```

Output:



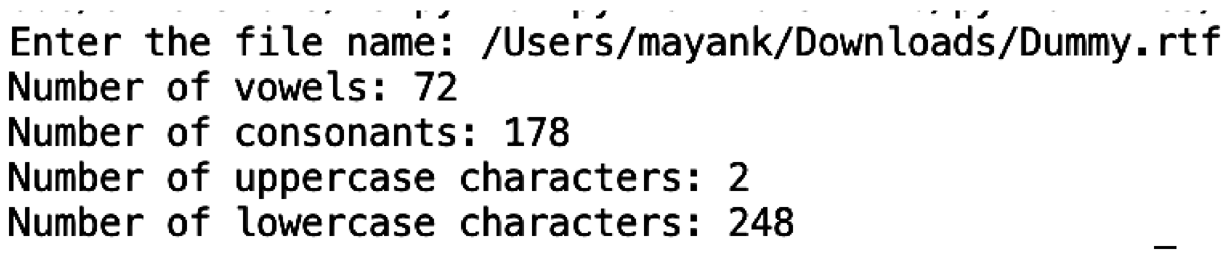
```
Enter the file name: /Users/mayank/Downloads/Dummy.rtf

Hello#welcome#to#the#world#of#python
```

10. Write a python program to read a text file and display the number of vowels, consonants, uppercase, lowercase characters in a file.

Code:

```
file_name = input("Enter the file name: ")

try:
   with open(file_name, 'r') as file:
       content = file.read()

       vowels = 0
       consonants = 0
       uppercase = 0
       lowercase = 0

       for char in content:
           if char.isalpha():
               if char.lower() in 'aeiou':
                   vowels += 1
               else:
                   consonants += 1
               if char.islower():
                   lowercase += 1
               else:
                   uppercase += 1

       print("Number of vowels:", vowels)

       print("Number of consonants:", consonants)

       print("Number of uppercase characters:", uppercase)

       print("Number of lowercase characters:", lowercase)

except FileNotFoundError:

   print("File not found.")
```

Output:



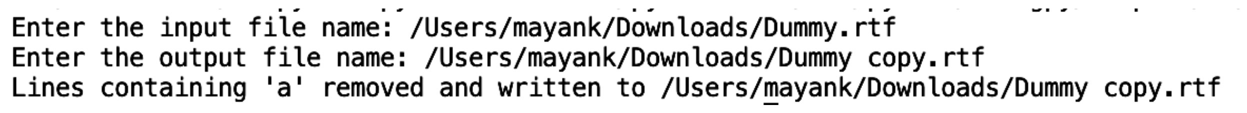
```
Enter the file name: /Users/mayank/Downloads/Dummy.rtf
Number of vowels: 72
Number of consonants: 178
Number of uppercase characters: 2
Number of lowercase characters: 248
```

11. Write a python program to remove all the lines that contain the character ‘a’ in a file and write it to another file.

Code:

```
input_file_name = input("Enter the input file name: ")

output_file_name = input("Enter the output file name: ")



try:

   with open(input_file_name, 'r') as input_file:

       with open(output_file_name, 'w') as output_file:

           for line in input_file:

               if 'a' not in line:

                   output_file.write(line)

   print("Lines containing 'a' removed and written to", output_file_name)

except FileNotFoundError:

   print("File not found.")
```

Output:



```
Enter the input file name: /Users/mayank/Downloads/Dummy.rtf
Enter the output file name: /Users/mayank/Downloads/Dummy copy.rtf
Lines containing 'a' removed and written to /Users/mayank/Downloads/Dummy copy.rtf
```

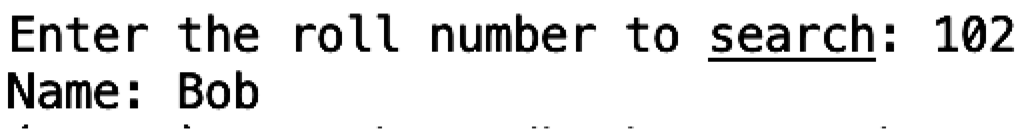
12. Write a python program to create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.

Code:

```
import pickle


def write_records(file_name, records):

    with open(file_name, 'wb') as file:

        pickle.dump(records, file)


def read_records(file_name):

    with open(file_name, 'rb') as file:

        records = pickle.load(file)

        return records


def search_roll_number(records, roll_number):

    for record in records:

        if record["roll_number"] == roll_number:

            return record["name"]

    return None


file_name = "student_records.dat"
```

```
records = [
   {"roll_number": 101, "name": "Alice"},
   {"roll_number": 102, "name": "Bob"},
   {"roll_number": 103, "name": "Charlie"}
]
write_records(file_name, records)


search_roll = int(input("Enter the roll number to search: "))
found_name = search_roll_number(read_records(file_name), search_roll)


if found_name:
   print("Name:", found_name)
else:
   print("Roll number not found.")
```

Output:



```
Enter the roll number to search: 102
Name: Bob
```

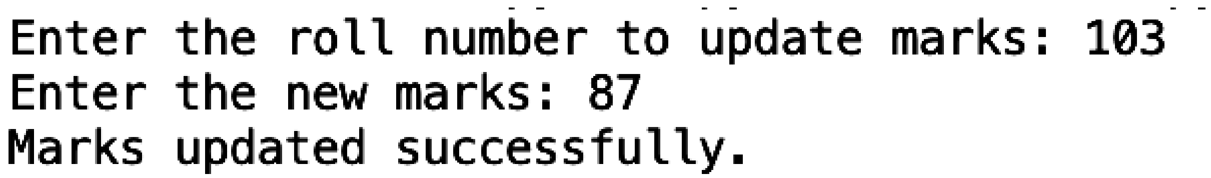
13. Write a python program to create a binary file with roll number, name and marks. Input a roll number and update the marks.

Code:

```python
import pickle


def write_records(file_name, records):
    with open(file_name, 'wb') as file:
        pickle.dump(records, file)


def read_records(file_name):
    with open(file_name, 'rb') as file:
        records = pickle.load(file)
        return records


def update_marks(records, roll_number, new_marks):
    for record in records:
        if record["roll_number"] == roll_number:
            record["marks"] = new_marks
            return True
    return False


file_name = "student_records.dat"
```

```
records = [
   {"roll_number": 101, "name": "Alice", "marks": 85},
   {"roll_number": 102, "name": "Bob", "marks": 92},
   {"roll_number": 103, "name": "Charlie", "marks": 78}
]
write_records(file_name, records)

search_roll = int(input("Enter the roll number to update marks: "))
new_marks = int(input("Enter the new marks: "))

updated = update_marks(read_records(file_name), search_roll, new_marks)

if updated:
   write_records(file_name, records)
   print("Marks updated successfully.")
else:
   print("Roll number not found.")
```

Output:



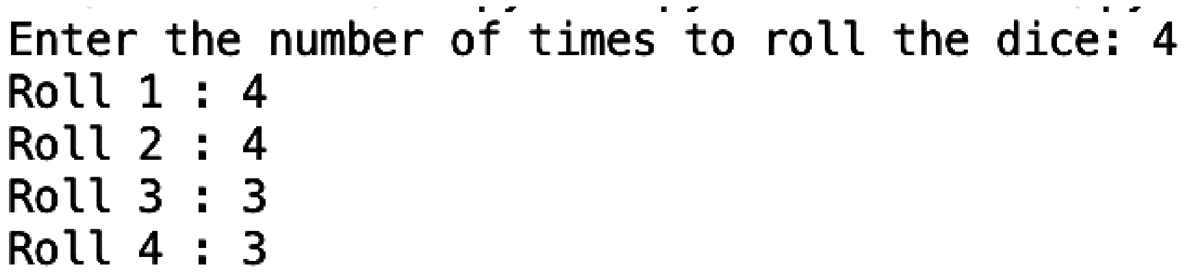
```
Enter the roll number to update marks: 103
Enter the new marks: 87
Marks updated successfully.
```

14. Write a python program to create a random number generator that generates a random number between 1 and 6(simulates a dice).

Code:

```python
import random


def roll_dice():

   return random.randint(1, 6)


num_rolls = int(input("Enter the number of times to roll the dice: "))


for i in range(num_rolls):

   result = roll_dice()

   print("Roll", i+1, ":", result)
```

Output:



```
Enter the number of times to roll the dice: 4
Roll 1 : 4
Roll 2 : 4
Roll 3 : 3
Roll 4 : 3
```

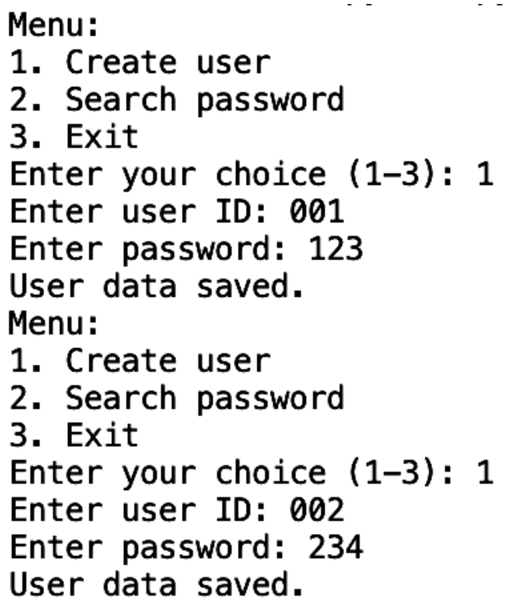
15. Write a python program to create a CSV file by entering user-id and password, read and search the password for given user-id.

Code:

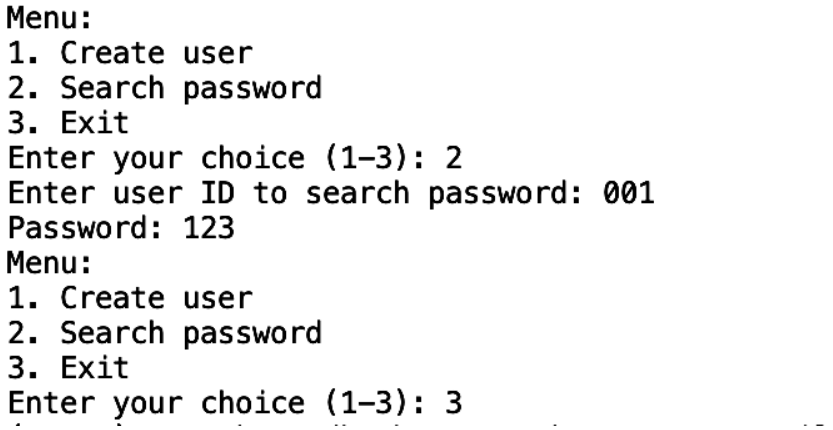
```
import csv


def write_to_csv(file_name, data):
   with open(file_name, 'a', newline='') as file:
       writer = csv.writer(file)
       writer.writerow(data)


def read_csv(file_name):
   records = []
   with open(file_name, 'r') as file:
       reader = csv.reader(file)
       for row in reader:
           records.append(row)
   return records


def search_password(records, user_id):
   for record in records:
       if record[0] == user_id:
           return record[1]
   return None
```

```python
file_name = "user_data.csv"

while True:
    print("Menu:")
    print("1. Create user")
    print("2. Search password")
    print("3. Exit")

    choice = input("Enter your choice (1-3): ")

    if choice == "1":
        user_id = input("Enter user ID: ")
        password = input("Enter password: ")
        write_to_csv(file_name, [user_id, password])
        print("User data saved.")
    elif choice == "2":
        user_id = input("Enter user ID to search password: ")
        records = read_csv(file_name)
        found_password = search_password(records, user_id)
        if found_password:
            print("Password:", found_password)
        else:
            print("User ID not found.")
    elif choice == "3":

        break

    else:

        print("Invalid choice.")
```

Output:



```
Menu:
1. Create user
2. Search password
3. Exit
Enter your choice (1-3): 1
Enter user ID: 001
Enter password: 123
User data saved.
Menu:
1. Create user
2. Search password
3. Exit
Enter your choice (1-3): 1
Enter user ID: 002
Enter password: 234
User data saved.
```



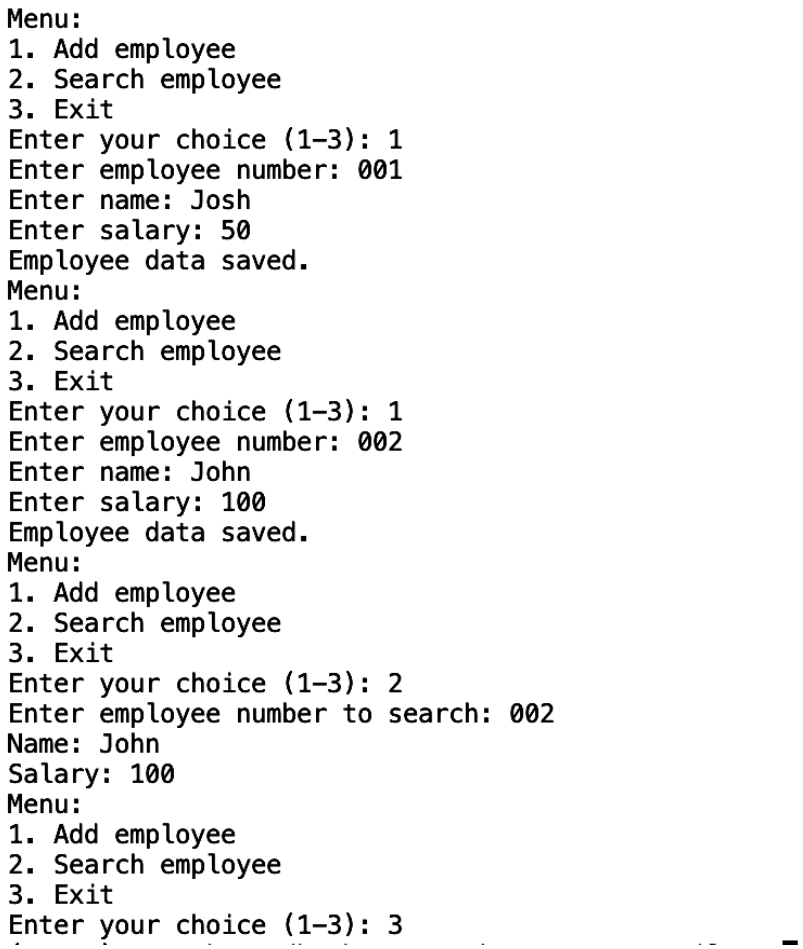
```
Menu:
1. Create user
2. Search password
3. Exit
Enter your choice (1-3): 2
Enter user ID to search password: 001
Password: 123
Menu:
1. Create user
2. Search password
3. Exit
Enter your choice (1-3): 3
```

16. Write a python program to create a CSV file and store employee number, name and salary. Search employee by employee number and display name and salary, and if not found display an appropriate message.

Code:

```
import csv

def write_to_csv(file_name, data):
    with open(file_name, 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(data)

def read_csv(file_name):
    records = []
    with open(file_name, 'r') as file:
        reader = csv.reader(file)
        for row in reader:
            records.append(row)
    return records

def search_employee(records, employee_number):
    for record in records:
        if record[0] == employee_number:
            return record[1], record[2]
    return None, None


file_name = "employee_data.csv"


while True:

    print("Menu:")

    print("1. Add employee")

    print("2. Search employee")

    print("3. Exit")


    choice = input("Enter your choice (1-3): ")


    if choice == "1":

        employee_number = input("Enter employee number: ")

        name = input("Enter name: ")

        salary = input("Enter salary: ")

        write_to_csv(file_name, [employee_number, name, salary])

        print("Employee data saved.")

    elif choice == "2":

        employee_number = input("Enter employee number to search: ")

        records = read_csv(file_name)

        found_name, found_salary = search_employee(records, employee_number)

        if found_name and found_salary:

            print("Name:", found_name)
            print("Salary:", found_salary)
        else:
            print("Employee not found.")
    elif choice == "3":
        break
    else:
        print("Invalid choice.")
```

Output:



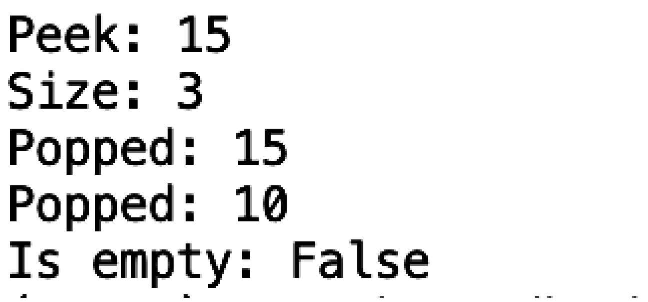
```
Menu:
1. Add employee
2. Search employee
3. Exit
Enter your choice (1-3): 1
Enter employee number: 001
Enter name: Josh
Enter salary: 50
Employee data saved.
Menu:
1. Add employee
2. Search employee
3. Exit
Enter your choice (1-3): 1
Enter employee number: 002
Enter name: John
Enter salary: 100
Employee data saved.
Menu:
1. Add employee
2. Search employee
3. Exit
Enter your choice (1-3): 2
Enter employee number to search: 002
Name: John
Salary: 100
Menu:
1. Add employee
2. Search employee
3. Exit
Enter your choice (1-3): 3
```

17. Write a python program to implement a stack using a list.

Code:

```
class Stack:
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        if not self.is_empty():
            return self.items.pop()
        else:
            return "Stack is empty"

    def peek(self):
        if not self.is_empty():
            return self.items[-1]
        else:
            return "Stack is empty"

    def is_empty(self):
        return len(self.items) == 0

    def size(self):
        return len(self.items)

 stack = Stack()

 stack.push(5)
 stack.push(10)
 stack.push(15)

 print("Peek:", stack.peek())
 print("Size:", stack.size())
 print("Popped:", stack.pop())
 print("Popped:", stack.pop())
 print("Is empty:", stack.is_empty())
```

Output:



```
Peek: 15
Size: 3
Popped: 15
Popped: 10
Is empty: False
```

18. Write a query to create a database School and execute following commands:

i. To show database

ii. To use database

Query:

```
CREATE DATABASE School;

SHOW DATABASES;

USE School;
```

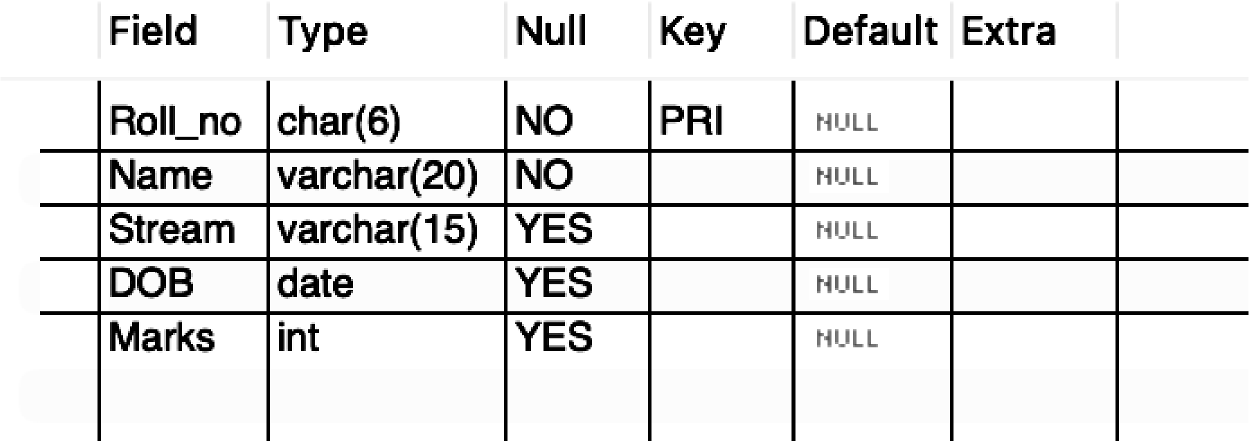
19. Write a query to create Student table using following structure :

Field Name	Data Type	
Roll_no	Char(6)	Primary Key
Name	Varchar(20)	Not Null
Stream	Varchar(15)	
DOB	Date	
Marks	Number	

Query:

```
CREATE TABLE Students (

   Roll_no CHAR(6) PRIMARY KEY,

   Name VARCHAR(20) NOT NULL,

   Stream VARCHAR(15),

   DOB DATE,

   Marks INT );
```

Output:



Field	Type	Null	Key	Default	Extra
Roll_no	char(6)	NO	PRI	NULL	
Name	varchar(20)	NO		NULL	
Stream	varchar(15)	YES		NULL	
DOB	date	YES		NULL	
Marks	int	YES		NULL	

20. Write a query to perform following operations on table student:

i. Insert data in table student

ii. Display roll_no, name from table student

iii. Display name of student having marks<85 and stream='medical'

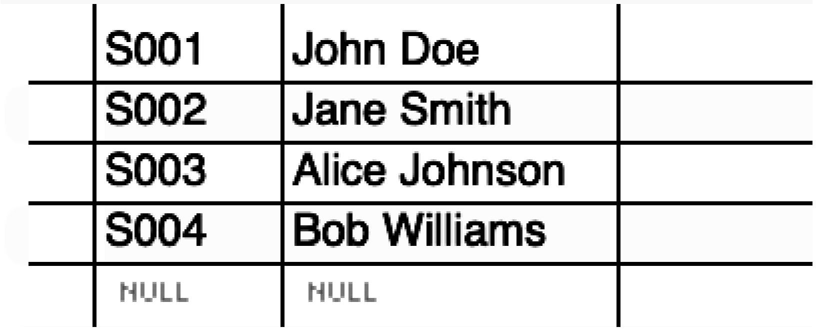
Query:

```
INSERT INTO Students (Roll_no, Name, Stream, DOB, Marks)
VALUES
    ('S001', 'John Doe', 'Science', '2000-05-15', 90),
    ('S002', 'Jane Smith', 'Medical', '2001-02-28', 78),
    ('S003', 'Alice Johnson', 'Arts', '2000-09-10', 72),
    ('S004', 'Bob Williams', 'Medical', '2000-12-03', 79);
```

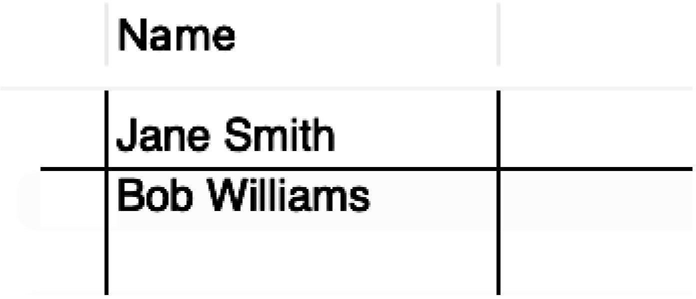
```
SELECT Roll_no, Name

FROM Students;
```

```
SELECT Name

FROM Students

WHERE Marks < 85 AND Stream = 'Medical';
```

Output:



S001	John Doe
S002	Jane Smith
S003	Alice Johnson
S004	Bob Williams
NULL	NULL



Name
Jane Smith
Bob Williams

21. Write a query to ALTER table to add a new attribute/modify data type and drop attribute.

Query:

```
I.    ALTER TABLE Students
      ADD Address VARCHAR(50);
II.   ALTER TABLE Students
      MODIFY Marks FLOAT;
III.  ALTER TABLE Students
      DROP COLUMN Stream;
```

22. Write a query to UPDATE Student table to modify data.

Query:

```
UPDATE Students

SET Marks = 85
```

23. Write a query to DISPLAY data in ascending/descending order

Query:

```
SELECT Name

FROM Students

ORDER BY Name ASC;
```

```
SELECT Name

FROM Students

ORDER BY Name DESC;
```

24. Write a query to DELETE the record of students where marks<35.

Query:

```
DELETE FROM Students

WHERE Marks < 35;
```

25. Write a query to GROUP BY and find min, max, sum, count and average of marks.

Query:

```
SELECT Stream,

    MIN(Marks) AS Min_Marks,

    MAX(Marks) AS Max_Marks,

    SUM(Marks) AS Total_Marks,

    COUNT(*) AS Count_Students,

    AVG(Marks) AS Avg_Marks

FROM Students

GROUP BY Stream;
```

26. Write a query to join two different tables.

Query:

SELECT A.rollno, A.name, B.fee

FROM Student A, Fees

WHERE A.rollno = B.rollno;

27. Write a python program to connect with database and store record of employee and display records.

Code:

```
import mysql.connector

mydb = mysql.connector.connect(

   host="localhost",

   user="root",

   password="root",

   database="test"

)

mycursor = mydb.cursor()

mycursor.execute('''CREATE TABLE employees (

                   id INT AUTO_INCREMENT PRIMARY KEY,

                   name VARCHAR(255),

                   department VARCHAR(255),
                   position VARCHAR(255),
                   salary DECIMAL(10, 2)
               )''')
def insert_employee_record(name, department, position, salary):
   sql = "INSERT INTO employees (name, department, position, salary) VALUES (%s, %s, %s, %s)"
   val = (name, department, position, salary)
   mycursor.execute(sql, val)
   mydb.commit()
   print("Employee record inserted successfully!")
def display_all_employees():
   mycursor.execute("SELECT * FROM employees")
   records = mycursor.fetchall()
   if len(records) > 0:
       print("Employee Records:")
       for record in records:
           print(record)
   else:
       print("No employee records found.")
insert_employee_record('John Doe', 'IT', 'Developer', 60000)
insert_employee_record('Alice Smith', 'HR', 'Manager', 75000)
insert_employee_record('Eva Johnson', 'Marketing', 'Analyst', 55000)
display_all_employees()
mydb.close()
```

28. Write a python program to connect with database and search employee number in the table employee and display record, if employee not found display appropriate message

Code:

```
import mysql.connector
conn = mysql.connector.connect(
host="localhost",
user="root",
password="root",
database="test"
)
cursor = conn.cursor()
def search_employee(employee_id):
   sql = 'SELECT * FROM Employee WHERE ID = %s'
   cursor.execute(sql, (employee_id,))
   record = cursor.fetchone()
   return record
employee_id_to_search = int(input("Enter Employee ID to search: "))
employee_record = search_employee(employee_id_to_search)
if employee_record:
   print("Employee Record:")
   print("ID:", employee_record[0])
```

```
    print("Name:", employee_record[1])

    print("Department:", employee_record[2])

    print("Salary:", employee_record[3])

else:

    print("Employee not found.")

conn.close()
```

29. Write a python program to connect with database and update the record of the entered employee number.

Code:

```
import mysql.connector

conn = mysql.connector.connect(

    host="localhost",

    user="root",

    password="root",

    database="test"

)

cursor = conn.cursor()


def update_employee_record(employee_id, new_name, new_department, new_salary):

    sql = 'UPDATE Employee SET Name = %s, Department = %s, Salary = %s WHERE ID = %s'

    values = (new_name, new_department, new_salary, employee_id)
    cursor.execute(sql, values)

    conn.commit()


employee_id_to_update = int(input("Enter Employee ID to update: "))


cursor.execute('SELECT * FROM Employee WHERE ID = %s', (employee_id_to_update,))

existing_employee = cursor.fetchone()


if existing_employee:

    print("Current Employee Record:")

    print("ID:", existing_employee[0])

    print("Name:", existing_employee[1])

    print("Department:", existing_employee[2])

    print("Salary:", existing_employee[3])

    new_name = input("Enter new name: ")

    new_department = input("Enter new department: ")

    new_salary = float(input("Enter new salary: "))

    update_employee_record(employee_id_to_update, new_name, new_department, new_salary)

    print("Employee record updated.")

else:

    print("Employee ID not found.")


conn.close()
```

30. Write a python program to connect with database and delete the record of the entered employee number.

Code:

```
import mysql.connector

conn = mysql.connector.connect(
   host="localhost",
   user="root",
   password="root",
   database="test"
)

cursor = conn.cursor()

def delete_employee_record(employee_id):
   sql = 'DELETE FROM Employee WHERE ID = %s'
   cursor.execute(sql, (employee_id,))
   conn.commit()

employee_id_to_delete = int(input("Enter Employee ID to delete: "))

cursor.execute('SELECT * FROM Employee WHERE ID = %s', (employee_id_to_delete,))
```

```
existing_employee = cursor.fetchone()

if existing_employee:
   print("Employee Record to be Deleted:")
   print("ID:", existing_employee[0])
   print("Name:", existing_employee[1])
   print("Department:", existing_employee[2])
   print("Salary:", existing_employee[3])

   confirmation = input("Are you sure you want to delete this employee? (yes/no): ")

   if confirmation.lower() == "yes":
       delete_employee_record(employee_id_to_delete)
       print("Employee record deleted.")
   else:
       print("Deletion cancelled.")
else:
   print("Employee not found.")

conn.close()
```