CLASS XII RECORD CODES:

1. SUMMING THE SERIES

AIM: To compute the sum of the following series :

i) $x^1/2! - x^3/4! + x^5/6! - x^7/8! + \ldots$ 'n' terms

ii) $x^1/3! - x^3/5! + x^5/7! - x^7/9! + \ldots$ 'n' terms

METHODOLOGY : The values for x and n are read as inputs and the sum of the 2 series are computed. The program makes use of one user defined function to compute the factorial and another to compute the powers of x.

Note: Include 2 UDFs Factorial() and Power() to compute the sum of the series

2. STATISTICAL FUNCTIONS

AIM : To compute the mean, mode and median of a list of integers.

METHODOLOGY: A list of integers are read as input and passes as argument to an user defined function which computes the mean, mode, median and returns all the three values which is printed in the calling function.

Note: The udf should return all 3 values which are to be printed in the calling function(main)

3. MENU DRIVEN CODE- NUMERIC

AIM: To write user defined functions and test them with a menu driven code

METHODOLOGY: The factorial computation and generation of Fibonacci series is performed using 2 user defined functions. A menu is displayed and the user's choice is accepted. Based on the choice the appropriate function is executed.

Note: The menu to be displayed :

MENU

1. Factorial of a number
2. Generate 'N' numbers of Fibonacci series
3. Exit

4. MENU DRIVEN CODE- LIST/STRING

AIM: To write user defined functions and test them with a menu driven code

METHODOLOGY: The Binary Search operation on a list of integers in ascending order and Reversing a String operation is performed using 2 user defined functions. A menu is displayed and the user's choice is accepted. Based on the choice the appropriate function is executed.

Note: The menu to be displayed:

MENU

1. Binary Search in a list of integers
2. Reversing a string
3. Exit

5. LARGEST, SECOND LARGEST, PRIME NUMBERS FROM A RANDOM GENERATION

AIM: To find the largest, second largest and the prime numbers from a list of randomly generated integers

METHODOLOGY: The lower and upper limits are read as inputs. Three user defined functions are written to do the required operations namely i)to generate the random numbers between the given limits ii) to find the largest, second largest number in the generated list and iii) to store the prime numbers in a local list and print them else print a message stating 'No prime numbers in the generated list'.

Note: Include 3 UDFs Genreate(), Max_SecMax(), Prime() to do the above.

6. MENU DRIVEN CODE – TUPLES

AIM : To perform the menu options on a tuple of values read as input.

METHODOLOGY: The menu options are displayed. As per the user's choice, within each of the menu options, a tuple is read as input and the appropriate functions are called. The first menu option, reads a tuple of 'n' words, calls a function that takes the tuple as parameter which checks if each word has all the 5 vowels in any case and returns a tuple of such words. The second menu option reads a nested tuple having height, weight of 'n' persons, calls a function that takes as parameter, each element of the nested tuple, which calculates the BMI and returns both the BMI and the result as OBESE /OVERWEIGHT/NORMAL/UNDERWEIGHT. The returned values are displayed in the main function for each person.

Note: The displayed menu should be as follows:

MENU

i) Display words with all vowels
ii) Check BMI
iii) Exit

To check BMI(computed as Kg/m^2) and return OBESE /OVERWEIGHT/NORMAL/UNDERWEIGHT, use the following data:

BMI >= 30 Obese;
>25 Overweight
18.5 – 25 Normal.
<18.5 – Underweight

7. MENU DRIVEN CODE – DICTIONARY

AIM: to create a dictionary to store details of 'n' subscribers and perform the menu operations.

METHODOLOGY: The details of 'n' subscribers namely name and phone number are read and stored in a dictionary. Using a menu and user's choice, the operations like Add a subscriber detail, View all subscribers, Modify name of a given subscriber, Delete an existing subscriber are done using user defined functions.

Note: The menu to be display should be:

MENU
1. ADD
2. VIEW
3. MODIFY NAME
4. DELETE
5. EXIT

Creating the dictionary should be done only by calling option 1)ADD repeatedly.
In Modify and Delete, the input taken is the phone number. If the number does not exist in the list, appropriate message should be shown

8. NESTED DICTIONARY

AIM: To create a nested dictionary and manipulate the same.

METHODOLOGY: A nested dictionary is created having the main keys to be 3 categories namely SENIORS, JUNIORS, SUBJUNIORS. For each of these keys, an inner dictionary is created with keys as BHARATHI, TAGORE, SHIVAJI, PRATAP and values as the score for that house. The code makes use of an user defined function MAX_SCORE() that takes the dictionary as parameter and displays the house having maximum score for each category.

Note: Strictly follow the uppercase for all the keys as mentioned above.

9. USER DEFINED MODULE – 1

AIM: To create an user defined module NUMBER to include 2 user defined functions PALINDROME(), SPECIAL() and import this module in another python code and execute the functions

METHODOLOGY: A module NUMBER is created to include the 2 functions namely PALINDROME() which takes a number as parameter and returns 1 if it's a palindrome and -1 otherwise; SPECIAL() which takes a number as parameter and returns 1 if it's a special number and -1 otherwise.

This module is imported in another python code and both the functions are executed. For the PALINDROME() function, a tuple of integers are read and the code displays all the palindrome numbers in the tuple. If there weren't any palindrome numbers appropriate message is shown. For the SPECIAL() function 2 limits are read and all the special numbers between these limits are generated. If there were no special numbers, appropriate message is shown.

Note: Create a module NUMBER to include the functions, with each function to include docstrings to describe the function. Also write a python code to import this module and use the two functions for inputs from the user.
i) palindrome() to take as parameter a number and returns 1 if the number is a palindrome and –1 otherwise. This function to be tested with a tuple of integers
ii) special() – takes as parameter a number and returns 1 if it's a special number and -1 otherwise. [A special number is a number equal to the sum of the factorial of the individual digits of the number] This function to be tested to generate list of special numbers between two limits accepted from the user.

10. USER DEFINED MODULE 2

AIM: To create an user defined module MATRIX to include 2 user defined functions SWAP(),

SORTROWCOL() and import this module in another python code and execute the functions
METHODOLOGY: A module MATRIX is created to include the 2 functions namely SWAP() which takes a nested list of integers as parameter and swaps the main and secondary diagonal elements; SORTROWCOL() which takes a nested list of integers as parameter and sorts each row in ascending order using Bubble sort followed by each column in ascending order using Insertion sort.
This module is imported in another python code and both the functions are executed with the outputs displayed in the main function.
Note: Create a module MATRIX to include the functions, with each function to include docstrings to describe the function. Also write a python code to import this module and use the two functions for inputs from the user.
i) swap() to take as parameter a nested list and swaps the left and right diagonal elements. This function to be tested with a nested list of integers
ii) sortrowcol() – takes as parameter a nested list of integers and sorts the rows using bubble sort in ascending order and sorts columns using Insertion sort in ascending order. This function to be tested with a nested list of integers.
Both the outputs to be shown in the main function

11. TEXT FILE 1
AIM: To create a text file with contents entered by the user as long as the user wishes to and perform the operations in the menu according to the user's choice
METHODOLOGY: A text file is created by reading one line at a time and asking is the user wants to input more lines with a Y/N choice. A menu is then displayed and the related operations are performed on the text file, depending on user's choice.
Note: The menu options to be as follows:
MENU
1. Display number of lines
2. Copy the words containing 'U' into another file and display the new file
3. Convert the case of the letters(lower to upper and vice versa) in the original text file and display the contents
4. Exit

12. TEXT FILE 2
AIM: To create a text file with 'N' lines of text and perform the operations in the menu according to the user's choice
METHODOLOGY: A text file is created by 'N' lines of text. A menu is then displayed and the related operations are performed on the text file, depending on user's choice.
Note: The menu options to be as follows:
MENU
1. Count the number of words
2. Display Palindrome words
3. Display words starting and ending with a vowel
4. Exit

13. BINARY FILE 1
AIM : To create a Binary file STUDENT.DAT/.BIN/.TXT with each record having details of rollnumber ,name, stream, total and perform the operations using a menu driven code.
METHODOLOGY: The binary file is created to store details of 'N' students, with input read in the form of a list. The menu is displayed and the related operations are performed on the binary file depending on user's choice.
Note: The menu options to be as follows:
MENU
1. Display Stream wise topper detail
2. Increment the total by 3 marks for students in the biology stream and decrement by 2for students in EG stream.
3. Exit

14.BINARY FILE 2
AIM : To create a Binary file EMPLOYEE.DAT with each record having fields Name, Age, Department , Designation, Salary and perform the operations using a menu driven code.

METHODOLOGY: The binary file is created to store details of 'N' employees, with input read in the form of a dictionary. The menu is displayed and the related operations are performed on the binary file depending on user's choice.

Note: The menu options to be as follows:

MENU

1. Display details of Managers earning more than 50000 in Finance and in Admin Dept.
2. Delete the employee details who have reached retirement age(58 years)
3. Exit

15. CSV FILE - WITH LISTS

AIM: To create CSV files and manipulate its contents

METHODOLOGY: A GST.CSV is created to store 4 records with the following data:

CATEGORY	GST_PERCENTAGE
Automobiles	25
Stationary	12
Chocolates	10
Dairy	3

A second file ITEMS.CSV is created to have fields ItemID, Name, Category, Unitprice, Finalprice. The input for 'N' items are read from the user for the first 4 fields. The Finalprice is computed using the GST.CSV file retrieving the GST% as per the category. The contents of the completed ITEMS.CSV is then displayed

Note: Make sure the Category you input for ITEMS.CSV is one of the Category in GST.CSV

16. STACK OPERATIONS - 1

AIM : To implement a Stack using a list of integers and performing the related operations using user defined functions.

METHODOLOGY: The Stack operations namely Push, Pop, Peek, Display Stack Status are performed on a list of integers using user defined functions

Note: The menu options to be displayed:

MENU

1. Push
2. Pop
3. Display Stack
4. Peek
5. Exit

17. STACK OPERATIONS - 2

AIM : To implement a stack using a list where each element in the list has Country name and Capital. The related operations are performed using user defined functions.

METHODOLOGY: The stack operations namely Insert, Delete, Peek, Display Stack Status are performed on a nested list using user defined functions

Note: The menu options to be displayed:

MENU

1. Insert
2. Delete
3. Display Stack
4. Peek
5. Exit

18. PYTHON MYSQL INTERFACE

AIM: To create a connectivity and provide an interface between Python and MySQL to perform queries using DML commands

METHODOLOGY: The mysql.connector module is imported and the interface between Python and MySQL is created. Using the DDL command CREATE TABLE, a student table structure is created. Using DML commands INSERT, SELECT, UPDATE the operations are performed to obtain the desired outputs.

Note:

Create a table **STUDENT** using python interface having atleast 5 records with the following attributes :

ROLL,NAME,TOTAL,COURSE,DOB. Also apply constraints PRIMARY KEY for ROLL and CHECK for COURSE to take values only from 'CS', 'BIOLOGY', 'COMMERCE', 'EG'. Also write queries to

perform the following operations:
A. Find maximum total marks, minimum total marks and number of students in each course where there at least 2 students.
B. Display details of students born in the month of may 2003 who have scored total between 100 to 200.
C. Sort the student list in the descending order of total.
D. Increase total marks for CS students by 5 %, for those, whose total is less than 180.

19. PYTHON MYSQL INTERFACE- CSV CREATION

AIM: To create a connectivity and provide an interface between Python, MySQL and create a CSV file using the contents of the table in MySQL.

METHODOLOGY: The mysql.connector module is imported and the interface between Python and MySQL is created. The csv module is imported to use the functions for creation and retrieval of the CSV file. The records of the Student table are written into the Student.CSV and the contents are displayed in an appropriate format.

Note:

Using the STUDENT table created in the earlier code, transfer all the records into a Student.CSV and perform the following operations:
i) Display the contents from the CSV file, showing values in each record separated by a comma
ii) Read from the CSV file and display the name of the top scorer.

20. PYTHON- MYSQL CONNECTIVITY FOR JOINS

AIM: To create a connectivity and provide an interface between Python and MySQL to perform join operations on 2 tables

METHODOLOGY: The mysql.connector module is imported and the interface between Python and MySQL is created. Using the DDL, DML commands the Employee and Department tables are created. A query is then executed to join the tables and display the output in the desired format

Note:

Create Employee table (Empno, Name, Desig, Deptno, Salary) and Department table(Deptno, Dname, Location) with appropriate number of records. Execute a query to display details of all Mangers like Empno, Name, Salary, Dname and Location.