PROGRAMMING FOR PROBLEM SOLVING LABORATORY

II Semester: AERO / MECH / CIVIL

Course Code	Category	Hours / Week			Credits	Maximum Marks		
ACSC03	Foundation	L	T	P	C	CIA	SEE	Total
		0	0	3	1.5	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 36				Total Classes:36		

Prerequisite: Knowledge of Python programming

I.COURSE OVERVIEW:

This course introduces students to writing computer programs. This course presents the principles of structured programming using the Python language, one of the most increasingly preferred languages for programming today. Because of its ease of use, it is ideal as a first programming language and runs on both the PC and Macintosh platforms. However, the knowledge gained in the course can be applied later to other languages such as C and Java. The course uses iPython Notebook to afford a more interactive experience. Topics include fundamentals of computer programming in Python, object-oriented programming and graphical user interfaces.

II.COURSE OBJECTIVES:

The students will try to learn:

I Acquire programming skills in core Python.

II Acquire Object-oriented programming skills in Python.

III Develop the skill of designing graphical-user interfaces (GUI) in Python.

IV Develop the ability to write database applications in Python.

IV Acquire Python programming skills to move into specific branches - Internet of Things (IoT), Data Science, Machine Learning (ML), Artificial Intelligence (AI) etc.

III. COURSE OUTCOMES:

After successful completion of the course, students should be able to:

CO 1	**Demonstrate** the basic concepts of python programming with the help of data types, operators and expressions, console input/output.	Apply
CO 2	**Make use of** f control statements for altering the sequential execution of programss in solving problems.	Analyze
CO 3	**Demonstrate** operations on built-in container data types (list, tuple, set, dictionary) and strings.	Apply
CO 4	**Make use of** operations and applications on strings with the help of builtin functions.	Analyze
CO 5	**Solve** the problems by using modular programming concepts throughfunctions.	Apply
CO 6	**Identify** object-oriented programming constructs for developing large, modular and reusable real-time programs.	Apply

IV. SYLLABUS

Week – 1: STUDENTS IN A COLLEGE

1. There are D departments in a college and each department has A_i number of students. Your task is to find the total number of students in the college.

   Input Format:
   The first line of input contains an integer D , the number of departments
   The second line of input contains D space-separated integers denoting number of students in each department.

   Output Format:
   A single integer(the total number of students in the college)

   Example:
   Input:
   3

1 2 3

Output:
6

2. In Statistics, range is defined as the difference between highest and lowest values. Given marks of students in a class, find the range.

Input Format:
The first line of input contains an integer N, the number of students
The second line of input contains N space-separated integers denoting the marks of each student in the class.

Output Format:
A single integer(the range)

Example:
Input:
5
10 20 40 20 30

Output:
30

Week – 2: TRIANGLES

1. What is the maximum number of squares of size **2x2** that can be fit in a right angled isosceles triangle of base **B**. One side of the square must be parallel to the base of the isosceles triangle. Base is the shortest side of the triangle.

Input Format:
The first line of input contains an integer T , denoting number of test cases.
Each of the next T lines contain a single integer , B (base)

Output Format:
For each test case , print a single integer, the number of squares.

Example:
Input:
3
1
1
4
11
Output:
0
0
1
10

2. Given 3 sides of a triangle, check whether the given three sides form a triangle and if so , check if it is an equilateral , isosceles or scalene triangle, also print its area.

Input Format:
The first line of input contains an integer T , denoting number of test cases.
Each of the next T lines contain 3 space separated integers , the 3 sides

Output Format:
For each test case , if the given 3 sides form a triangle ,
Print “EQUILATERAL” / “ISOSCELES” /”SCALENE” followed by the area (up to 2 decimal places)
If they do not form a triangle , print “NOT A TRIANGLE”

Example:

Input:
```
5
3 3 3
3 5 3
3 7 3
4 2 3
1 2 3
```

Output:
```
EQUILATERAL 3.89
ISOSCELES 4.14
NOT A TRIANGLE
SCALENE 2.90
NOT A TRIANGLE
```

Week – 3: MAGIC SQUARE

1. A magic square of size N is a square matrix of order NxN that satisfies these conditions.
   a. It should contain all elements from 1 to N^2 without repetitions.
   b. The sum of the numbers in any row,column or diagonal should be equal.

   Write a Python program to check whether a given matrix is a magic square or not

   Input Format:
   The first line of input contains an integer N, the order of the square matrix
   Each of the next N lines contain N-space separated integers denoting the elements of the matrix

   Output Format:
   Print “YES” if it is a magic square, else print “NO”.

   Example:

   Input:
   ```
   3
   8 1 6
   3 5 7
   4 9 2
   ```

   Output:
   ```
   YES
   ```

Week – 4: RUNNING RACE

1. The scores of participants in a running race are given, find the runner up.

   Input Format:
   The first line of input contains an integer T, the number of test cases
   Each of the next T lines contain some space separated integers denoting the participant’s scores

   Output Format:
   For each test case, print a single integer denoting the score of the runner up. If there is no runner up, print “NONE”.

   Example:

   Input:
   ```
   5
   1 2 3 4 5
   5 5 5 5 2 5 5 2
   5 5 5 5 5 5
   10 20 30 40 50
   ```

19 76 89 12 34 78 90 90 76 89 90
Output:
4
2
NONE
40
89

2. The scores of participants in a running race were recorded but the person recording the scores made some errors and added some duplicate entries. Remove all duplicate entries and print the count of the errors made.

Input Format:
The first line of input contains an integer N, the number of scores that were recorded
The second line of input contains N space-separated integers denoting the recorded scores.

Output Format:
The first line of output should contain the distinct scores after removing duplicate entries.
The second line of output should contain an integer denoting the number of errors made.

Example:
Input:
10
1 2 3 1 1 3 4 2 8 9

Output:
1 2 3 4 8 9
4

Week – 5: PANGRAM

1. Given a string check if it is Pangram or not. A pangram is a sentence containing every letter in the English Alphabet. Ignore case and special characters.

Input Format:
The first line of input contains an integer T, the number of test cases.
Each of the following T lines contain a string

Output Format:
For each test case, print “PANGRAM” or “NOT PANGRAM”.

Example:
Input:
3
The quick brown fox jumps over the lazy dog
$!#@ ABC DEF ghi jkl mnop qrst uvw XYZ @#!$
Institute of Aeronautical Engineering

Output:
PANGRAM
PANGRAM
NOT PANGRAM

Week – 6: FREQUENCY OF LETTERS

1. Given a sentence, print the frequency of each English letter present in the sentence, in alphabetic order. Consider all characters to be lowercase.

Input Format:
A sentence

Output Format:
For every character, print the character followed by a hyphen and then the frequency (in alphabetic order). Ignore digits and special characters and consider uppercase letters also as lowercase.

Example:
Input:
12345 This is a sentence @IARE
Output:
a-2
c-1
e-4
h-1
i-3
n-2
r-1
s-3
t-2

Week – 7: BINARY NUMBERS

1. Write a program to convert a given decimal number into binary.

Input Format:
The first line of input contains an integer T denoting the number of test cases.
Each of the next T lines contains decimal integers.

Output Format:
For each test case, print the binary equivalent.

Example:
Input:
4
1
3
5
10

Output:
1
11
101
1010

2. Write a program to convert a given binary number into decimal form.

Input Format:
The first line of input contains an integer T denoting the number of test cases.
Each of the next T lines contains binary integers.

Output Format:
For each test case, print the decimal equivalent.

Example:
Input:
4
1
11
101
1001

```
Output:
1
3
5
9
```

Week – 8: PATTERNS

1. Write a Python program to print the following pattern.

```
N=5
   *
  ***
 *****
  ***
   *
```

2. Write a Python program to print the following pattern.

```
S= SCHOOL
     IIIIIII
     IAAAAAI
     IARRRAI
     IARERAI
     IARRRAI
     IAAAAAI
        IIIIIII
```

Week – 9: COMBINATIONS

1. Given an array of size n, generate and print all possible combinations of r elements in array.

Input Format:
First line contains Space-separated integers denoting array elements.
Second line contains r , size of each combination

Output Format:
Print each combination in a separate line and every combination should have comma separated integers.

Example:

```
Input:
1 2 3 4
2

Output:
1,2
1,3
1,4
2,3
2,4
3,4
```

Week – 10: CLASS AND OBJECTS

1. Create a Temperature class. Make two methods.
   i. Convert Fahrenheit - It will take Celsius and will print it into Fahrenheit.
   ii. Convert Celsius - It will take Fahrenheit and will convert it into Celsius.

2. Create a Time class and initialize it with hours and minutes.
   i. Make a method add Time which should take two time object and add them. E.g.- (2 hour and 50 min) + (1 hr and 20 min) is (4 hr and 10 min)
   ii. Make a method display Time which should print the time.

iii. Make a method Display Minute which should display the total minutes in the Time. E.g.- (1 hr 2 min) should display 62 minute.

Week – 11: ROMAN NUMERAL

1. Write a Python program to convert a decimal number into its roman numeral form.

Input Format:
The first line of input contains an integer T denoting the number of test cases.
Each of the next T lines contains decimal integers.

Output Format:
For each test case, print the roman numeral equivalent.

Example:
Input:
4
10
100
999
2020

Output:
X
C
CMXCIX
MMXX

2. Write a Python program to convert a roman numeral into its decimal form.

Input Format:
The first line of input contains an integer T denoting the number of test cases.
Each of the next T lines contains roman numbers.

Output Format:
For each test case, print the decimal equivalent.

Example:
Input:
4
XII
C
DXCVII
MMXX

Output:
12
100
597
2020

Week – 12: FILE HANDLING

1. Write a Python program to count the number of characters, words, lines in a file.

Example:

Input File:

```
First line
Second line
Third line
```

Output:

```
Characters:31
Words:6
Lines:3
```

2. Write a Python program to add line numbers to a file.

Example:

Input File:

```
First line
Second line
Third line
```

Output:

```
1.  First line
2.  Second line
3.  Third line
```

V. REFERENCE BOOKS:

1. Michael H Goldwasser, David Letscher, "Object Oriented Programming in Python", Prentice Hall, 1st Edition, 2007.
2. Yashavant Kanetkar, Aditya Kanetkar, "Let us Python", BPB publication, 1st Edition, 2019.
3. Ashok Kamthane, Amit Kamthane, "Programming and Problem Solving with Python", McGraw Hill Education (India) Private Limited, 2018.
4. Taneja Sheetal, Kumar Naveen, "Python Programming – A modular approach", Pearson, 2017.
5. R Nageswara Rao, "Core Python Programming", Dreamtech Press, 2017 Edition.

VI. WEB REFERENCES:

1. https://www.codesdope.com/practice/python-your-class/
2. https://www.geeksforgeeks.org/python-programming-language/
3. https://www.hackerrank.com/
4. https://www.codechef.com/