

20MCA131 PROGRAMMING LAB

End semester examination pattern:

Maximum Marks: 50			
Verification of Daily program record and Git Repository			5 marks
Viva			10 marks
Problem solving (Based on difficulty level, one or more questions may be given)	Flowchart / Algorithm / Structured description of problem to explain how the problem can be solved / Interface Design	15%	35 marks
	Program correctness	50%	
	Code efficiency	15%	
	Formatted output and Pushing to remote Git repository	20%	
Total Marks			50 marks

Course Level Assessment Questions

1. Display future leap years from current year to a final year entered by user.
2. List comprehensions:
 - a. Generate positive list of numbers from a given list of integers
 - b. Square of N numbers
 - c. Form a list of vowels selected from a given word
 - d. List ordinal value of each element of a word (Hint: use ord() to get ordinal values)
3. Count the occurrences of each word in a line of text.
4. Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.
5. Store a list of first names. Count the occurrences of 'a' within the list
6. Enter 2 lists of integers. Check:
 - a. Whether list are of same length
 - b. whether list sums to same value
 - c. whether any value occur in both
7. Get a string from an input string where all occurrences of first character replaced with '\$', except first character. [eg: onion -> oni\$N]

8. Create a string from given string where first and last characters exchanged. [eg:
python -> nythop]
9. Accept the radius from user and find area of circle.
10. Find biggest of 3 numbers entered.
11. Accept a file name from user and print extension of that.
12. Create a list of colors from comma-separated color names entered by user. Display
first and last colors.
13. Accept an integer n and compute $n+nn+nnn$.
14. Print out all colors from color-list1 not contained in color-list2.
15. Create a single string separated with space from two strings by swapping the
character at position 1.
16. Sort dictionary in ascending and descending order.
17. Merge two dictionaries.
18. Find gcd of 2 numbers.
19. From a list of integers, create a list removing even numbers.
20. Program to find the factorial of a number
21. Generate Fibonacci series of N terms
22. Find the sum of all items in a list.
23. Generate a list of four digit numbers in a given range with all their digits even and the
number is a perfect square.
24. Display the given pyramid with step number accepted from user.

Eg: N=4

1

2 4

3 6 9

4 8 12 16

25. Count the number of characters (character frequency) in a string.
26. Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'
27. Accept a list of words and return length of longest word.
28. Construct following pattern using nested loop

```

*
* *
* * *
* * * *
* * *
* *
*

```

29. Generate all factors of a number.
30. Write lambda functions to find area of square, rectangle and triangle.
31. Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements).
32. Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.
33. Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.
34. Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.
35. Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.
36. Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.
37. Write a Python program to read a file line by line and store it into a list.
38. Python program to copy odd lines of one file to other

39. Write a Python program to read each row from a given csv file and print a list of strings.
40. Write a Python program to read specific columns of a given CSV file and print the content of the columns.
41. Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.