GLS UNIVERSITY

FACULTY OF COMPUTER APPLICATIONS & IT IMScIT

SEM 5

221601505 PRACTICALS ON PYTHON

UNIT– 2 LAB TASK

Sr. No.	Question
1	Write a Python program to accept marks for 5 subjects, calculate average, and assign grade (A/B/C/D/Fail). If any subject has marks <35, display “Fail due to subject back”.

2	Print the following hollow triangle pattern using nested `for` loops:
	
	    *
	   * *
	  * *
	 * *
	******

3	Accept 10 integers. Remove duplicates, sort in descending order, print the second highest and second lowest unique value, and average of top 5 values.
4	Accept a tuple of 10 integers. Count even and odd numbers. Extract prime numbers into a new tuple. Display max, min, and sum of the new tuple.
5	Create a dictionary with employee names and salaries. Increase salary by 10% if it's <50,000. Remove employees whose updated salary exceeds 1,00,000.
6	Take 10 numbers from the user. Use `continue` to skip numbers divisible by both 3 and 5. Use `break` if number is negative. If loop completes, use `else` to print “Input complete”.
7	Create two sets: even and prime numbers between 1–20. Find union, intersection, difference, and symmetric difference. Convert one to frozenset and try modifying it (handle the error).
8	Given a list of numbers, use `enumerate()` to print index and value. Use `all()` to check if all elements are positive, and `any()` to check if any is a multiple of 7.
9	Build a student record system using a dictionary. Include add, update, delete, and sort by marks using `lambda`.
10	Demonstrate list operations: `append()`, `extend()`, `insert()`, `remove()`, `pop()`, `reverse()`, `sort()` in descending order.
11	Create a nested loop to generate a multiplication table from 1 to 10 (in matrix format).
12	Accept a string and create a dictionary where each key is a character and value is its frequency. Ignore spaces and make it case-insensitive.

13	Continuously accept numbers until user enters 0. Store in a list. Print max, min, sum, and count of even numbers.
14	Accept a tuple of integers. Print elements whose square is > 50 using list comprehension.
15	Create two lists: one with keys and one with values. Combine them into a dictionary using `zip()` and update one of the values.
16	Accept 10 numbers. Create a second list with even numbers only. Count frequency of each even number using `count()` method.
17	Accept a sentence and create a dictionary where each word is a key and value is its length. Print the word(s) with the maximum length.
18	Write a login system using a dictionary. Allow up to 3 login attempts and show success or failure with proper message.
19	Accept a number and check whether it is a palindrome using logic (no string conversion allowed).
20	Accept a tuple of integers. Without converting to list, find the 3rd highest and 2nd lowest values.
21	Create a program that checks whether a given list of brackets (like `'([{}])'`) is balanced using stack logic (using list as stack).
22	Write a program that reads a list of words and returns a dictionary where keys are word lengths and values are lists of words of that length.
23	Create a nested dictionary that stores student names as keys, and their subjects and marks as sub-dictionaries. Add, update and delete subjects or marks.
24	Write a Python program to flatten a nested list (e.g., `[[1, 2], [3, 4], [5]]`) into a single list `[1, 2, 3, 4, 5]` without using built-in `sum()`.
25	Create a number guessing game where the program randomly picks a number from 1 to 100, and the user has to guess it in limited attempts (use loop + conditions).