Reg. No.:

Name :

VIT BHOPAL
www.vitbhopal.ac.in

TERM END EXAMINATIONS (TEE) – May 2023

Programme	**:**	**B.Tech.**	**Semester:**	**Summer Semester 2022-23**
Course Title/ Course Code	**:**	**Introduction to problem solving / CSE1021**	**Slot;**	**A11-A17**
Time	**:**	**1½ hours**	**Max. Marks**	**50**

Answer ALL the Questions

Q. No.		Question Description	Marks

PART - A (30 Marks)

1 (A) In a mall, there are several stores, and people visit these stores for shopping. Write an algorithm to determine the most visited store in the mall based on the number of people who enter each store at different point of time in a day. **10**

OR

(B) In a university, there is a large group of student applications with various CGPA, who have applied for different majors (CSE, EEE, CSE(AI), Mech, …..etc). Write an algorithm determine the most popular major among the students based on their CGPA. The popularity of a major is determined by the number of students enrolled in that major with higher CGPA. **10**

2 (A) Create a Python program that evaluates an arithmetic expression containing variables by substituting their values. The expression should include basic arithmetic operations such as addition, subtraction, multiplication, and division. **10**

The program should follow the following steps:

a. Input Variables and Expression: Prompt the user to input the values of the variables used in the arithmetic expression. Ask the user to enter the arithmetic expression.
b. Variable Substitution: Parse the arithmetic expression to identify the variables present. For each variable found in the expression, substitute its value provided by the user.
c. Arithmetic Expression Evaluation: Evaluate the arithmetic expression using the substituted variable values. Handle the order of operations (PEMDAS/BODMAS) to ensure correct evaluation.
d. Output: Print the substituted expression with the evaluated result.

OR

(B) Create a Python program that calculates the factorial of a positive integer using an iteration statement. The program should handle invalid input gracefully by providing appropriate error messages. **10**

The program should follow the following steps:

a. Prompt the user to input a positive integer for which the factorial will be calculated.
b. Check if the input value is a positive integer. If the input is not a positive integer, display an error message and ask the user to input a valid positive integer. Repeat this step until a valid positive integer is provided.
c. Initialize a variable **factorial** to 1. Use an iteration statement (e.g., **for** loop) to multiply **factorial** by numbers from 1 to the input value. Update the **factorial** value with each multiplication.
d. Print the calculated factorial value.

3 (A) Create a Python program that efficiently computes the Nth Fibonacci number by storing previously computed Fibonacci numbers to avoid redundant calculations and improve performance. **10**

The program should follow the following steps:

a. Prompt the user to input a positive integer **N** representing the position of the Fibonacci number to be computed.
b. Check if the input value is a positive integer. If the input is not a positive integer, display an error message and ask the user to input a valid positive integer. Repeat this step until a valid positive integer is provided.
c. Initialize a list **fibonacci** with the first two Fibonacci numbers, [0, 1]. Use a loop or iteration statement to calculate the remaining Fibonacci numbers up to the Nth position efficiently. Avoid redundant calculations by utilizing the previously computed Fibonacci numbers in the **fibonacci** list. Store each computed Fibonacci number in the **fibonacci** list as you calculate it.
d. Print the Nth Fibonacci number. previously computed Fibonacci numbers in the **fibonacci** list. Store each computed Fibonacci number in the **fibonacci** list as you calculate it.
e. Print the Nth Fibonacci number.

OR

(B) Create a Python program that removes duplicates from the array in-place, without using additional memory to create a new array. The program should modify the original array to contain only unique elements. **10**

The program should follow the following steps:

a. Input an Ordered Array:
b. Prompt the user to input an ordered array of integers. Ensure that the array is sorted in ascending order. Handle invalid input gracefully, such as non-numeric values or unsorted arrays.
c. Duplicate Removal: Implement an algorithm to remove duplicates from the ordered array in-place. Use a loop or iteration statement to traverse the array and compare adjacent elements. Remove duplicate elements by shifting the remaining elements to the left. Update the length of the array accordingly after removing duplicates.
d. Output: Print the modified array after removing duplicates.

PART-B (20 MARKS)

4 Create a program that determines the type of a triangle based on its side lengths. The program should utilize Boolean values, operators, and conditional statements to handle various triangle types. **10**

The program should include the following:

a. Define a function named **triangle_type** that takes three side lengths (**a**, **b**, **c**) as input.
b. Check if any of the side lengths (**a**, **b**, **c**) are less than or equal to zero. If so, return the message "Invalid triangle. Side lengths must be positive."
c. Determine the type of the triangle based on the following conditions: If all three sides (**a**, **b**, and **c**) are equal, the triangle is an equilateral triangle. If all three sides are different, the triangle is a scalene triangle. If two sides are equal and the third side is different, the triangle is an isosceles triangle.
d. Return the corresponding triangle type based on the conditions met.

5 Create a program that performs various operations on different collection types and measures the execution time for a time trade-off scenario. **10**

a. List Operations: fruits = ['apple', 'banana', 'cherry'] Append the element 'orange' to the list of fruits. Insert the element 'grape' at index 1 in the list of fruits. Remove the element 'banana' from the list of fruits.
b. Tuple Operations: colors = ('red', 'green', 'blue') Access and print the first element of the tuple of colors.
c. Set Operations: vegetables = {'carrot', 'potato', 'spinach'} Add the element 'broccoli' to the set of vegetables. Remove the element 'potato' from the set of vegetables.
d. Dictionary Operations: student = {'name': 'John', 'age': 20, grade': 'A'}
e. Print the age of the student from the dictionary. Update the grade of the student to 'B' in the dictionary. Delete the 'age' key from the dictionary.
f. Time Trade-off: Write a code to Measure the execution time for a time-consuming operation by iterating a large number of times. Write a code Print the execution time in seconds.

⇔⇔⇔