Set 1

Roll No.: _

Amrita Vishwa Vidyapeetham

Amrita School of Computing, Coimbatore

B.Tech. Mid-term Examinations – Oct 2023

First Semester

Computer Science and Engineering

23CSE101 Computational Problem Solving

Duration: Two hours Maximum: 50Marks

Course Outcomes (COs):

CO#	Course Outcomes
CO01	Apply algorithmic thinking to understand, define and solve problems.
CO02	Design and implement algorithm(s) for a given problem.
CO03	Apply the basic programming constructs for developing solutions and programs.
CO04	Analyze an algorithm by tracing its computational states, identifying bugs and correcting them.

Answer all questions.

Part A **30 Marks**

1. Using a Python program, simulate a simple car race where cars move along a straight track. The cars have different starting positions and speeds, and they move forward based on a series of time steps. Your goal is to determine which car reaches the finish line first.

   The racetrack is straight, and cars move only in one direction (towards the finish line). Time measurement is to be done in discrete time steps. Each car's position is represented as an integer (measured in meters from the start line), and the speed is represented as an integer (measured in meters per time step). The finish line is located at a fixed distance (say, 50 meters) from the start line. You may assume that the speed remains constant for the entire duration of the race. [15 marks] [CO2] [BTL3]

   a. Initialize two cars with their positions and speeds as variable properties. (5 marks) Example:
      - Car 1 starts at position 0 meters and has a speed of 2 meters per time step.
      - Car 2 starts at position 10 meters and has a speed of 3 meters per time step.

   b. Simulate the movement of the two cars until either one car overtakes the other, or one (or both) of the cars reach the finish line. Update the positions of the cars based on their speeds and print the output for each time step. (5 marks)

c. Determine which car wins the race. The car that overtakes the other before the finish line or the one that crosses the finish line first wins. If both cars finish at the same time, consider it a tie. (5 marks)

Sample Input:

Each line of the input contains information about a car, consisting of three numbers: the car number, its position from the starting point, and its speed in meters per second.

1 0 3
2 10 2

Sample Output:

Display the positions of each car until either one car overtakes the other or one (or both) of the cars reach the finish line. Each line of the output starts with the car number and is followed by the positions that they are in during a measurement. The display can stop as soon as a winner is decided, and the next position can be marked as 'Winner' or 'Runner-up'.

1 0 3 6 9 12 15 18 21 24 27 30 33 Winner

2 10 12 14 16 18 20 22 24 26 28 30 32 Runner-Up

2. A company maintains a weekly sales record to assess the profit or loss incurred for the 5 products based on the quantities sold as shown below. Based on the assessment, the quantity of the products can be updated in the forth coming weeks. Weekly sales record data for 2 weeks is entered as total number of quantities sold towards the end of the respective week. [15 marks] [CO2] [BTL3]

Product_ID={AAA,BBB,CCC,DDD,EEE}

Previous_Week_sales= {80,92,36,83,77}

Current_Week_sales= {29,64,69,83,120}

Use tuples to assign the above sales record data. Write a menu-driven python program that does the following based on the option selected.

1. Display the contents of the tuple.
2. Compute the difference in the sales values between Previous and Current Week and add it to a list. Finally, display the list. The difference needs to be calculated as (Current_Week_sales – Previous_Week_sales).
3. Display the counts of products with higher, lower, and equal sales when compared to the previous week in the same line.
4. Based on the analysis, Display whether it is a profit or loss to the company in terms of quantities sold.

5. Display the product IDs, whose difference in sales values is greater than or equal to 30, to increase their sales in future.

Input Specification

- Input is the choice of operation to be performed.
- No product/sales data is read from the user, as the tuples are created based on the data given in the question.

Output Specification

- Print the output as per the choice opted for.

Sample Input

1

Sample Output

('AAA', 'BBB', 'CCC', 'DDD', 'EEE')

(80, 92, 36, 83, 77)

(29, 64, 69, 83, 120)

[**Expected results for the same input for other possible options**

 # for option 2

[-51, -28, 33, 0, 43]

 # for option 3

2 2 1

 # for option 4

Loss (#Display 'Profit' if it is profit to the company)

 # for option 5

CCC

EEE

]

Part B **20 Marks**

3. The following algorithm tries to print the integers between a starting value and an ending value, with the condition that every multiple of 5 is printed as 'Found' instead of the number itself. For example, printing the numbers between 8 and 13 will be **8 9 Found 11 12 13**. [5 marks] [CO01] [BTL2]
   a. Will the algorithm terminate? Justify your answer. **[2 marks]**

b. The given algorithm is incomplete as the output produced is not same as the expected output. Identify the differences. **[3 marks]**

```
Step 1: Start
Step 2: Declare step, start and end
Step 3: Read/Input start and end
Step 4: If start <= end, then
      Step 4.1: Assign step = start
      Step 4.2: While step <= end, do the following sub-steps
            Step 4.2.1: Print step
            Step 4.2.2: Step = Step – 1, go back to step 4.2.
      Step 4.3: Go to Step 6.
Step 5: Else Print "Invalid range"
Step 6: Stop
```

4. You are tasked with Primality checking i.e., figuring out if the number inputted is a prime number or a composite number. [5 marks] [CO01] [BTL3]
    a. Write an algorithm that expresses sequence of steps to be followed to solve the aforesaid problem in natural language (English). Discuss the shortcomings of expressing algorithms in natural language. **(2 marks)**
    b. Draw a flowchart for the above problem to pictorially illustrate various operations to be performed and explain how flowcharts can overcome the limitations of algorithms. **(3 marks)**

5. Run-length encoding (RLE) is a very simple form of data compression in which a string of data is given as the input (i.e. "AAABBCCCC") and the output is a sequence of counts of consecutive data values in a row (i.e. "3A2B4C"). Write a python program to compress a given input string using run-length encoding. [5 marks] [CO03] [BTL3].

    For example, given the input string "cooooooooooooooooolkangaroo" the program should output "1c17o1l1k1a1n1g1a1r2o".

6. Given below is the incomplete python script that creates and manipulates data structures such as list, tuple, and dictionary. Do analyse and suitably fill up the blanks found throughout the code to make it complete. [5 marks] [CO04] [BTL4]

```
# Creating a list of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
# Creating a set of vowels
vow = {‘a’, ‘e’, ‘i’, ‘o’, ‘u’}

# Creating a dictionary of student names and their ages
student_ages = {
    "Alice": 20,
    "Bob": 22,
    "Charlie": 19,
    "David": 21,
    "Eve": 23
}

# Adding a new number to the list
new_number = 11
_____ #Todo 1: Insertion into list (1 mark)
print(f"Added {new_number} to the list: {numbers}")

# Removing a number from the list
remove_number = 5
if remove_number in numbers:
    _____ #Todo 2: Deletion from list (1 marks)
    print(f"Removed {remove_number} from the list: {numbers}")
else:
    print(f"{remove_number} is not in the list.")

# Deleting the vowel ‘e’ from the set
delete_from_set = ‘e’
_____ #Todo 3: Deletion from set (1 mark)
print(f"Deleted {delete_from_set} from the set: {vowels}")

# Adding a new student to the dictionary
new_student = "Frank"
new_age = 18
_____ #Todo 4: Insertion into dictionary (1 mark)
print(f"Added {new_student} to the dictionary: {student_ages}")

# Deleting a student from the dictionary
delete_student = "Charlie"
if delete_student in student_ages:
    _____ #Todo 5: Deletion from dictionary (1 mark)
    print(f"Deleted {delete_student} from the dictionary: {student_ages}")
else:
    print(f"{delete_student} is not in the dictionary.")
```

****

CO	Marks	BTL	Marks
CO01	10	BTL 1	-
CO02	30	BTL 2	5
CO03	5	BTL 3	40
CO04	5	BTL 4	5
-	-	BTL 5	-
-	-	BTL 6	-

Course Outcome /Bloom's Taxonomy Level (BTL) Mark Distribution Table)

Bloom's Taxonomy Levels (attached for reference)

Level 1 – Remember

Level 2 - Understand

Level 3 – Apply

Level 4 - Analyze

Level 5 - Evaluate

Level 6- Create