UNITED INTERNATIONAL UNIVERSITY

Department of Computer Science and Engineering (CSE)

Lab Final

60 mins

Set A

Course : DSA II Lab | **Course Code: CSE 2218**

Trimester & Year: Fall 24 | **Section: K**

Name : | **Total Marks: 40**

ID:

SN	Questions	Marks
1	You are given a string `s` consisting of lowercase and uppercase English letters. Your task is to determine whether the string is a palindrome or not. A string is considered a palindrome if it reads the same backward as forward. **To solve this problem, you must use a recursive approach.** The program will not be case-sensitive, meaning that strings like `Abcba` will also be considered palindromes. **Input:** • The input consists of a single line containing the string `s` (1 ≤ |s| ≤ 1000). **Output:** • Print `YES` if the string `s` is a palindrome. Otherwise, print `NO`.	**10**

Sample Input	Sample Output
Abcba	YES
Hello	NO

Marking Criteria

Logic	**3**
Base case	**2**
Proper recursive call	**3**
Overall correctness	**2**

SN	Questions	Marks
2	You are given an integer n.On each step, you may either: ● Subtract one of the digits of the number, or ● Divide the number by 2 (if it is even). What is the minimum number of steps required to make the number equal to 0? **Input:** ● The only input line has an integer **n**. **Output:** ● Print one integer: the minimum number of steps. **Constraints** The time complexity of your solution must not exceed **O(n)**.	**10**

Sample Input	Sample Output
27	4
Explanation: An optimal solution is 27→20→10→5→0.	

Marking Criteria

Logic	**4**
Implementation	**4**
Overall correctness	**2**

SN	Questions	Marks
3	You are given a vector **A** and a series of operations to perform on it. The operations are provided as a single continuous string. Your task is to process these operations and output the results of specific commands. **Input:** ● The first line contains an integer Q (1≤Q≤100), the number of operations. ● The second line contains a string S, where: ○ Each operation and its associated integer (if applicable) are concatenated directly. ○ Example: `a4a6a7bce` means: ■ `a4` → Add 4 to the vector. ■ `a6` → Add 6 to the vector. ■ `a7` → Add 7 to the vector. ■ `b` → Sort the vector in ascending order. ■ `c` → Reverse the vector. ■ `e` → Print the elements of the vector. Each operation can be one of the following types: **1.** `aX` – Add the integer X to the end of the vector A. **2.** `b` – Sort the vector A in ascending order. **3.** `c` – Reverse the vector A. **4.** `d` – Print the size of the vector A. **5.** `e` – Print the space-separated elements of the vector A. **6.** `f` – Sort the vector A in descending order. **Output:** For each output-producing operation (`d` or `e`), print the result on a new line: ● For `d`, print the size of the vector. ● For `e`, print the space-separated elements of the vector. **Sample Input** / **Sample Output** 4 a55a11de 2 55 11 6 a4a6a7bce 7 6 4 **Marking Criteria** **Logic** — **4** **Implementation** — **4** **Overall correctness** — **2**	**10**