Date: 07/02/2026

Duration: 1 hour 45

CONTINUOUS ASSESSMENT 1

II YEAR – SEMESTER 4

CSE23AE204 - PCP III

(B.Tech. E01,E02,E03,E05,E06)

SET C

Question 1: Problem Statement

Problem Statement

You are working on a **resource allocation system** used in an enterprise environment.

Every time resources are provisioned, the system generates a **sequential allocation log** where:

- Allocation IDs increase continuously
- Allocations are grouped row-wise
- Each row contains one more allocation than the previous row

Since the validation engine cannot parse console output directly, the log must be returned as a **collection of text rows**, where each row is stored as a string.

Your task is to generate this allocation log based on a given integer N.

Input Format

- A single integer N representing the number of allocation levels.

Output Format

- Return an **array/list of strings**
- Each string represents one row of the allocation log

Example 1

Input

4

Output

```
[
  "1",
  "2 3",
  "4 5 6",
  "7 8 9 10"
]
```

Question 2: Problem Statement

A modern warehouse uses autonomous drones to scan storage shelves arranged in a grid.

The warehouse layout is represented as a **2D matrix**, where:

grid[i][j] = the QR code value stored at row i, column j

Drones are programmed to move in a **diagonal zig-zag pattern**, starting from the **top-left** corner:

The traversal pattern is:

- Move **up-right** until you hit a boundary
- Then switch direction
- Move **down-left**
- Repeat until the entire warehouse grid is scanned

Your task is to simulate the drone's traversal and return the **order of QR codes scanned**.

Input Format

grid → 2D integer matrix (rows × columns)

Output Format

integer array → QR codes in the order they were scanned

Example 1

Input:

```
grid = [
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9]
]
```

Output:

[1,2,4,7,5,3,6,8,9]

Explanation:

The drone path is:

1 → 2 → 4 → 7 → 5 → 3 → 6 → 8 → 9

Date: 07/02/2026
Duration: 1 hour 45

CONTINUOUS ASSESSMENT 1
II YEAR – SEMESTER 4
CSE23AE204 - PCP III
(B.Tech. E01,E02,E03,E05,E06)

SET C

Question 3: Problem Statement

You are developing a **payment processing module** for an online platform.

The system processes **one payment request at a time**, provided as a **space-separated string**.
Each payment is handled by a specific payment mode, and each mode follows its own processing rules.

The system must:

- Identify the payment type from input
- Process the payment based on payment mode
- Deduct applicable charges
- Return the final amount that gets debited
- Use **abstraction and polymorphism** to support multiple payment types

Payment Processing Rules

UPI Payment

- A processing charge is deducted based on a percentage of the amount.
- The remaining amount is the final debited amount.

Card Payment

- A fixed service charge is deducted.
- The remaining amount is the final debited amount.

Net Banking Payment

- No processing charge is applied.
- The full amount is debited.

Input Format

A **single space-separated string**.

UPI Payment

UPI <UserName> <Amount> <ChargePercentage>

Card Payment

CARD <UserName> <Amount> <ServiceCharge>

Net Banking Payment

NETBANKING <UserName> <Amount>

Output Format

Return a **double value** representing the **final debited amount**.

Example 1

Input

{ "payment": "UPI Ravi 1000 2" }

Output

980.0

Explanation

A percentage-based processing charge is deducted from the original amount, and the remaining amount is debited.