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 E

Question 1: Problem Statement

You are part of a **resource planning team** responsible for validating symmetric allocation grids used by internal tools.

The system represents allocations in a **uniform, slanted layout** where:

- Each row contains the **same number of allocation units**
- Alignment is achieved through **leading spaces**
- The grid must appear symmetric when rendered

To support automated validation, the grid must be returned as an **array/list of strings**, where each string represents one row.

Your task is to generate the allocation grid based on a given integer N.

Input Format

- A single integer N representing the size of the grid.

Output Format

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

Example 1

Input

4

Output

```
[
  "   1 2 3 4",
  "  1 2 3 4",
  " 1 2 3 4",
  "1 2 3 4"
]
```

Question 2: Problem Statement

A company generates **employee badge codes** using characters from a given pool of characters.

For visual symmetry and design standards, the badge code must be a **palindrome**.

You are given a string S containing **uppercase and lowercase English letters**.

Each character can be used **only as many times as it appears** in the string.

You may **rearrange the characters in any order**.

Your Task

Determine the **maximum possible length** of a palindromic badge code that can be generated.

Input Format

A single string S

Output Format

Print an integer representing the **maximum length** of the palindrome that can be formed.

Example 1

Input

"abccccdd"

Output

7

Explanation

Palindrome can be "dccaccd"

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 E

Question 3: Problem Statement

You are developing a **premium calculation module** for an insurance company. The system processes **one insurance policy at a time**, provided as a **space-separated string**. Each policy belongs to a specific category and follows different premium calculation rules.

The system must:

- Identify the policy type from input
- Calculate the final premium amount
- Use **abstraction and polymorphism** to support multiple policy types
- Assign a unique policy ID using a **static variable**
- Return the calculated premium

Premium Rules

- A base premium is charged to all the policy.

Health Insurance Policy

- An age-based risk amount is added.
- The total premium is the sum of base premium and risk amount.

Vehicle Insurance Policy

- A percentage-based vehicle value charge is added.
- The total premium is the sum of base premium and value-based charge.

Life Insurance Policy

- A fixed coverage amount is added.
- The total premium is the sum of base premium and coverage amount.

Input Format

A **single space-separated string**.

Health Policy: HEALTH <CustomerName> <BasePremium> <AgeRiskAmount>

Vehicle Policy: VEHICLE <CustomerName> <BasePremium> <VehicleValuePercentage>

Life Policy: LIFE <CustomerName> <BasePremium> <CoverageAmount>

Output Format

Return a **double value** representing the **final insurance premium**.

Example 1

Input

{ "policy": "HEALTH Ravi 5000 1500" }

Output

6500.0

Explanation

The health policy premium is calculated by adding the age-based risk amount to the base premium.