# Question

# A class Matrix contains a two dimensional integer array of order [m x n].The maximum value possible for both ‘m’ and ‘n’ is 25 . Design a class Matrix to find the difference of the two matrices . The details of the members of the class are given below:

Classname:Matrix  
Data members/ instance variables:  
arr[ ][ ]:stores the matrix element  
m:integer to store the number of rows  
n:integer to store the number of columns

Member functions:  
Matrix(int mm,int nn): to initialize the size of the matrix m=mm and n=nn  
void fillarray():to enter the elements of the matrix  
Matrix SubMat(Matrix A):subtract the current object from the matrix of parameterized object and return the resulting object

void display():display the matrix elements

Specify the class Matrix giving details of the constructor(int,int ). void fillarray() ,Matrix SubMat(Matrix) and void display() . Also define a main() function to create an object and call the methods accordingly to enable the task.

**Algorithm**

1. Start

2. Define a class `Matrix` with three instance variables:

- `arr` of type `int[][]` to store the matrix elements.

- `m` of type `int` to store the number of rows.

- `n` of type `int` to store the number of columns.

3. Define a parameterized constructor for the class `Matrix`:

- Accept two integers `mm` and `nn` as parameters.

- Initialize `m` with `mm`.

- Initialize `n` with `nn`.

- Initialize `arr` with a new 2D array of size `[m][n]`.

4. Define a method `fillarray()` to enter the elements of the matrix:

- Create a `Scanner` object to read input from the user.

- Prompt the user to enter the elements of the matrix.

- Use nested loops to read and store the elements in the 2D array `arr`.

5. Define a method `SubMat(Matrix A)` to subtract the current object from the parameterized object:

- Create a new `Matrix` object `result` with the same dimensions.

- Use nested loops to iterate through each element of the matrices.

- For each element, subtract the corresponding element of the current object from the parameterized object and store the result in `result`.

- Return the `result` object.

6. Define a method `display()` to display the matrix elements:

- Use nested loops to iterate through each element of the matrix.

- Print each element followed by a space.

- Print a newline character after each row.

7. In the main method:

- Create a `Scanner` object to read input from the user.

- Prompt the user to enter the number of rows and columns (maximum 25 each).

- Read the number of rows and columns into variables `rows` and `cols`.

- If the number of rows or columns exceeds 25, print an error message and exit.

- Create two `Matrix` objects `matrix1` and `matrix2` with the given dimensions.

- Prompt the user to enter elements for the first matrix and call `fillarray()` on `matrix1`.

- Prompt the user to enter elements for the second matrix and call `fillarray()` on `matrix2`.

- Call `SubMat(matrix1)` on `matrix2` to perform the subtraction and store the result in `resultMatrix`.

- Display the result of the subtraction by calling `display()` on `resultMatrix`.

8. End

**Variable description**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data type** | **Purpose** |
| arr[][] | int | To store elements of the matrix |
| m | int | To store number of rows |
| n | int | To store number of columns |
| mm | int | To initialize number of rows |
| nn | int | To initialize number of columns |
| i | int | Loop counter to iterate through rows |
| j | int | Loop counter to iterate through columns |
| rows | int | To store number of rows entered by user |
| cols | int | To store number of columns entered by user |