**Question**

Write a Java program to enter a m x n matrix and find the sum of right diagonal .

The specifications are as follows :

Classname: Matrix

Instance variables:

Int arr[][] : to store input array

Int m : to store row number

Int n: to store column number

Member methods:

Matrix (int mm, int nn) – to initialize the array size with mm and nn

void input() - to take input from the user in the array

int rdigsum() – to calculate and return the sum of the right diagonals

void display() – display the original matrix and the sum of right diagonal

Implement the main function to create object and call the above methods.

**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 constructor for the class that takes two integers `mm` and `nn` as input:

- Set `m` to `mm`.

- Set `n` to `nn`.

- Initialize `arr` with size `[m][n]`.

4. Define a method `input`:

1. Create a Scanner object `scanner`.

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

3. For each row `i` from 0 to `m-1`:

- For each column `j` from 0 to `n-1`:

- Read an integer from the user and store it in `arr[i][j]`.

5. Define a method `rdigsum`:

1. Initialize `sum` to 0.

2. For each index `i` from 0 to the minimum of `m` and `n`:

- Add `arr[i][n-1-i]` to `sum`.

3. Return `sum`.

6. Define a method `display`:

1. Print the original matrix:

- For each row `i` from 0 to `m-1`:

- For each column `j` from 0 to `n-1`:

- Print the element `arr[i][j]`.

- Move to the next line.

2. Call the `rdigsum` method to get the sum of the right diagonal.

3. Print the sum of the right diagonal.

7. In the main method:

1. Create a Scanner object `scanner`.

2. Prompt the user to enter the number of rows `m`.

3. Read an integer from the user and store it in `m`.

4. Prompt the user to enter the number of columns `n`.

5. Read an integer from the user and store it in `n`.

6. Create a `Matrix` object `matrix` with `m` and `n`.

7. Call the `input` method of `matrix`.

8. Call the `display` method of `matrix`.

8. End

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| arr | int[][] | A 2D array to store the elements of the matrix. |
| m | int | Represents the number of rows in the matrix. |
| n | int | Represents the number of columns in the matrix. |
| mm | int | Parameter for the constructor to initialize the number of rows. |
| nn | int | Parameter for the constructor to initialize the number of columns. |
| Scanner | Scanner | A Scanner object used to read input from the user. |
| i | int | Loop control variable for iterating over rows. |
| j | int | Loop control variable for iterating over columns. |
| sum | int | Used to store the sum of the right diagonal elements of the matrix. |