**Question**

A class Shift contains a two-dimensional integer array of order (m×n) where the maximum values of both m and n are 5. Design the class Shift to shuffle the matrix (i.e. the first row becomes the last, the second row becomes the first and so on). The details of the members of the class are given below:   
Class name: Shift  
Data member/instance variable:  
mat[][]: stores the array element  
m: integer to store the number of rows  
n: integer to store the number of columns  
Member functions/methods:  
Shift(int mm, int nn): parameterized constructor to initialize the data members m=mm and n=nn  
void input(): enters the elements of the array  
void cyclic(Shift p): enables the matrix of the object (P) to shift each row upwards in a cyclic manner and store the resultant matrix in the current object  
void display(): displays the matrix elements  
Specify the class Shift giving details of the constructor(), void input(), void cyclic(Shift) and void display(). Define the main() function to create an object and call the methods accordingly to enable the task of shifting the array elements.

**Algorithm**

1. Start

2. Define a class Shift with three instance variables:

- `mat` 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 Shift:

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

- Initialize `m` with `mm`.

- Initialize `n` with `nn`.

- Allocate memory for the 2D array `mat` with dimensions `m` x `n`.

4. Define a method input() to read 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 `mat`.

5. Define a method cyclic(Shift p) to perform a cyclic shift on the matrix:

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

- For each element, if the current row index `i` is equal to `m - 1`, set the element to the corresponding element from the first row of the matrix `p`.

- Otherwise, set the element to the corresponding element from the next row of the matrix `p`.

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 5 each).

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

- If the number of rows or columns exceeds 5, print an error message and exit.

- Create an object `originalMatrix` of the class `Shift` with the given dimensions.

- Call the `input()` method on `originalMatrix` to read the matrix elements.

- Create another object `shiftedMatrix` of the class `Shift` with the same dimensions.

- Call the `cyclic()` method on `shiftedMatrix` with `originalMatrix` as the argument to perform the cyclic shift.

- Call the `display()` method on `originalMatrix` to print the original matrix.

- Call the `display()` method on `shiftedMatrix` to print the shifted matrix.

- Close the `Scanner` object.

8. End

**Variable Description**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data type** | **Purpose** |
| mat | int[][] | 2D array to store matrix elements |
| m | int | Number of rows in the matrix |
| n | int | Number of columns in the matrix |
| mm | int | Parameter for number of rows in the matrix |
| nn | int | Parameter for number of columns in the matrix |
| i | int | Loop variable for row iteration |
| j | int | Loop variable for column iteration |
| rows | int | Number of rows entered by the user |
| cols | int | Number of columns entered by the user |