

ASSIGNMENT _ 1

Solution :

```
1 package College;
2
3 import java.util.Scanner;
4
5 public class Matrix_Patterns {
6
7     static void patternA(int[][] arr, int rows, int cols) { 1 usage
8         int sum = 0;
9         for (int i = 0; i < cols; i++) {
10             sum += arr[0][i];
11             System.out.print(arr[0][i] + " ");
12         }
13         System.out.println();
14         int middleRow = rows / 2;
15         for (int i = 1; i < rows; i++) {
16             sum += arr[i][0];
17             sum += arr[i][cols - 1];
18             for (int j = 0; j < cols; j++) {
19                 if (i == middleRow) {
20                     sum += arr[middleRow][j];
21                     System.out.print(arr[middleRow][j] + " ");
22                 } else if (j == 0 || j == cols - 1) {
23                     System.out.print(arr[i][j] + " ");
24                 } else {
25                     System.out.print(0 + " ");
26                 }
27             }
28             System.out.println();
29         }
30         sum -= arr[middleRow][0] + arr[middleRow][cols - 1];
31         System.out.println("Sum of the pattern is: " + sum);
32     }
```

```
33
34     static void patternX(int[][] arr, int rows, int cols) { 1 usage
35         int sum = 0;
36         for (int i = 0; i < rows; i++) {
37             for (int j = 0; j < cols; j++) {
38                 if (i == j) {
39                     sum += arr[i][j];
40                     System.out.print(arr[i][j] + " ");
41                 } else if (i + j == rows - 1) {
42                     sum += arr[i][j];
43                     System.out.print(arr[i][j] + " ");
44                 } else {
45                     System.out.print(0 + " ");
46                 }
47             }
48             System.out.println();
49         }
50         System.out.println("Sum of the pattern is: " + sum);
51     }
52 }
```

```

53     static void patternZ(int[][] arr, int rows, int cols) { // usage
54         int sum = 0;
55         for (int i = 0; i < cols; i++) {
56             sum += arr[0][i];
57             System.out.print(arr[0][i] + " ");
58         }
59         System.out.println();
60
61         for (int i = 1; i < rows - 1; i++) {
62             for (int j = 0; j < cols; j++) {
63                 if (i + j == rows - 1) {
64                     sum += arr[i][j];
65                     System.out.print(arr[i][j] + " ");
66                 } else {
67                     System.out.print(0 + " ");
68                 }
69             }
70             System.out.println();
71         }
72
73         for (int j = 0; j < cols; j++) {
74             sum += arr[rows - 1][j];
75             System.out.print(arr[rows - 1][j] + " ");
76         }
77         System.out.println();
78
79         System.out.println("Sum of the pattern is: " + sum);
80     }

```

```

82     static void patternD(int[][] arr, int rows, int cols) { // usage
83         int sum = 0;
84         for (int i = 0; i < cols; i++) {
85             sum += arr[0][i];
86             System.out.print(arr[0][i] + " ");
87         }
88         System.out.println();
89
90         for (int i = 1; i < rows - 1; i++) {
91             sum += arr[i][0];
92             sum += arr[i][cols - 1];
93
94             for (int j = 0; j < cols; j++) {
95                 if (j == 0) {
96                     System.out.print(arr[i][j] + " ");
97                 } else if (j == cols - 1) {
98                     System.out.print(arr[i][j] + " ");
99                 } else {
100                     System.out.print(0 + " ");
101                 }
102             }
103             System.out.println();
104         }
105
106         for (int j = 0; j < cols; j++) {
107             sum += arr[rows - 1][j];
108             System.out.print(arr[rows - 1][j] + " ");
109         }
110         System.out.println();
111
112         System.out.println("Sum of the pattern is: " + sum);
113     }

```

```

114
115 > public static void main(String[] args) {
116     Scanner sc = new Scanner(System.in);
117
118     System.out.print("Enter rows: ");
119     int rows = sc.nextInt();
120
121     System.out.print("Enter cols: ");
122     int cols = sc.nextInt();
123
124     int[][] arr = new int[rows][cols];
125     for (int i = 0; i < rows; i++) {
126         for (int j = 0; j < cols; j++) {
127             arr[i][j] = sc.nextInt();
128         }
129     }
130
131     System.out.println("Original Matrix: ");
132     for (int i = 0; i < rows; i++) {
133         for (int j = 0; j < cols; j++) {
134             System.out.print(arr[i][j] + " ");
135         }
136         System.out.println();
137     }
138

```

```

138
139     System.out.print("Enter Pattern: ");
140     char ch = sc.next().charAt(0);
141
142     switch (Character.toLowerCase(ch)) {
143         case 'x':
144             patternX(arr, rows, cols);
145             break;
146
147         case 'a':
148             patternA(arr, rows, cols);
149             break;
150
151         case 'd':
152             patternD(arr, rows, cols);
153             break;
154
155         case 'z':
156             patternZ(arr, rows, cols);
157             break;
158
159         default:
160             System.out.println("choose a correct pattern!!");
161             break;
162     }
163 }
164

```

OUTPUT :

X

```
Enter rows: 5
Enter cols: 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Original Matrix:
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Enter Pattern: x
1 0 0 0 5
0 2 0 4 0
0 0 3 0 0
0 2 0 4 0
1 0 0 0 5
Sum of the pattern is: 27

Process finished with exit code 0
```

A

```
Enter rows: 5
Enter cols: 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Original Matrix:
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Enter Pattern: a
1 2 3 4 5
1 0 0 0 5
1 2 3 4 5
1 0 0 0 5
1 0 0 0 5
Sum of the pattern is: 48

Process finished with exit code 0
```

D

```
Enter rows: 5
Enter cols: 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Original Matrix:
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Enter Pattern: d
1 2 3 4 5
1 0 0 0 5
1 0 0 0 5
1 0 0 0 5
1 2 3 4 5
Sum of the pattern is: 48
```

Z

```
Enter rows: 5
Enter cols: 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Original Matrix:
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
Enter Pattern: z
1 2 3 4 5
0 0 0 4 0
0 0 3 0 0
0 2 0 0 0
1 2 3 4 5
Sum of the pattern is: 39
```