```
/*
Given a boolean matrix (contains only 0 and 1) of size m X n
where each row is sorted, write an
algorithm and a program to find which row has maximum number
of 1's.

TIME COMPLEXITY -> O(N)

INPUT FORMAT:
  - first line contains m and n (the size of matrix).
  - for next m input lines, each line contains space-separated
n integers describing each row of the
     matrix.

OUTPUT FORMAT:
  - output will be row number which has maximum number of 1's.
If all the elements of matrix is 0
     then print "Not Present".
*/


// LIBS
#include <stdio.h>

// MAIN FUNCTION
int main() {
  int m = 4, n = 3;

  int matrix[m][n];

  for (int r_index = 0; r_index < m; r_index++) {
     for (int c_index = 0; c_index < n; c_index++) {
     scanf("%d", &matrix[r_index][c_index]);
     }
  }

  int r_index = 0, c_index = n - 1;
  int row = -1;

  while(r_index < m && c_index >= 0) {
     if (matrix[r_index][c_index] == 1) {
     row = r_index;
     c_index--;
     }
```

```
        else {
        r_index++;
        }
    }

    printf("%d\n", row);

    return 0;
}
```