**Summary:**

The problem involves creating a matrix and counting the number of "walls" between 0s and 1s in both the original matrix and its transpose. The program takes user input to construct the matrix, computes the transpose matrix, counts the walls in both matrices, and outputs the total count of walls needed to stop the virus.

**Solution:**

The program follows these steps to solve the problem:

1. Accepts user input for the dimensions of the matrix, represented by n (number of rows) and m (number of columns).

2. Constructs the matrix by calling the create\_matrix() function, which iterates n times and takes input for each row. It validates the input and creates the matrix accordingly.

3. Calculates the transpose matrix using the zip() function and stores it in transpose\_matrix.

4. Counts the number of walls in the original matrix by calling the build\_wall() function on matrix. The function checks each row for transitions between 0s and 1s and increments the count accordingly.

5. Similarly, counts the number of walls in the transpose matrix by calling build\_wall() on transpose\_matrix.

6. Prints the sum of row\_walls and column\_walls, representing the total count of walls needed to stop the virus.

**Outputs:**

Here are the outputs of the program for the given sample inputs:

Input:

9 8

0 0 0 0 0 0 0 0

0 1 1 1 1 1 1 0

0 0 1 1 0 1 0

0 0 0 1 1 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 0 1

1 1 1 1 0 0 0 0

0 0 1 0 1 0 0 1 0

0 0 0 0 1 1 1 1

Output:

30

Input:

4 8

0 1 0 0 0 0 1 1

0 1 0 0 0 0 1 1

0 0 0 0 0 0 1 1

0 0 0 0 0 0 0 1

Output:

10

Input:

3 3

0 0 0

0 1 0

0 0 0

Output:

4