

Following our great technical interview session, as a next step we would like to see your code writing skills. The solution should be sent back to us by an email.

Here is the problem:

- Having a matrix of **n** by **m** elements. Each of them can be **marked or not**. In the example shown this is expressed by white (not marked) and red (marked) colors. The numbers inside are just for clearance, and will not be given as an input condition.

|   |   |   |   |   |
|---|---|---|---|---|
| 1 |   |   |   |   |
| 1 |   | 2 | 2 |   |
|   | 2 |   | 2 |   |
|   | 2 | 2 | 2 |   |
|   |   |   |   | 3 |

- The goal is to find the **number of all individual figures**, by one figure we mean all colored elements (squares in the matrix) that touch each other by one of their sides. If two cells touch only a vertex (diagonally) and not a common side, they are not considered one figure.

For example, for the matrix shown above, the program must give us a **"3 figures"** solution. The figures themselves are marked with numbers in each square, but only for clarity. It is not necessary for the program to mark the affiliation of the fields as in this case if this will slow down the execution of the algorithm.

Requirements for the provided solution:

- Used programming language - C ++.
- It is important that the code is written professionally as a commercial software **meeting all quality requirements** as such. It would be good to consider a scalable solution that would work with large matrices of millions of elements.
- It is of no use to anyone if the algorithm is taken from the Internet, but on the other hand, it is not forbidden to seek knowledge in the problem area if you think it is necessary.

The task does not aim to take too much of your time, the goal is to see how you write production ready code. There is no need (still not forbidden) to further complicate the solution, such as the use of external databases, multi-threaded execution, etc.