# C++ OOP – Exam Retake (11 September 2022)

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++11 standard.

Submit your solutions here: <https://judge.softuni.org/Contests/3598/CPlusPlus-OOP-Retake-Exam-11-September-2022>

Only source code will be accepted as solution for each task.

# Bomberman

Remember the “Nintendo” and “Terminator” TV games? Bomberman was one of them.   
Your task is to play a similar game where you will lay out bombs and score points to win.

The game is played on a 2D grid map of characters.  
The map is filled with **ASCII digit characters** representing how many points does each tile award if you successfully bomb it.  
The game is played on **rounds**. Each round you are given a command to execute.  
After you execute the command you should print to the standard output the outcome of the command.  
  
The possible commands are:

* **power up –** increases the power of your bomb with 1.
* **power down** – decreases the power of your bomb with 1. (minimal bomb power is 0)
* **bomb** [**row**] [**col**], where ‘row’ and ‘col’ will be **indexes** from the 2D grid map -place a bomb at the given location. You are assured that the coordinates will always be valid.  
  The bomb shockwave destroys tiles only in **horizontal** and **vertical** direction (up, down, left and right).  
  The bomb has a “power” – used to determine how many tiles in each direction should the shockwave span to.  
   X X  
   **B** X **B** X X  
  (**Bomb** with power 0) X X X **B** X X  
   (**Bomb** with power 1) X  
   X  
   (**Bomb** with power 2)

After a tile has been destroyed it can **NO** longer award score points if it’s bombed another time.

The output for the commands should be:

* “Increased bomb power to N” (where N is the bomb power after the increase)
* “Decreased bomb power to N” (where N is the bomb power after the decrease)
* ”Score points” (where Score describes how many points were achieved after placing a bomb for the current round)

### Input

First you should read 2 **integers** from the console (rows) and (cols).

After than you should read “rows” lines of data. Each row will contain “cols” number of ASCII characters.

Following up read another **integer** (N) representing the numbers of commands (game rounds), which you should play.

Lastly read N rows of data, containing the commands for each individual round for the game.

### Restrictions

Time limit: 250ms (0.25s)  
Memory limit: 16 MB

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 4  1472  0019  5  bomb 0 1  power up  power up  bomb 0 2  power down | 4 points  Increased bomb power to 1  Increased bomb power to 2  11 points  Decreased bomb power to 1 |
| 3 4  0123  4567  8901  6  power up  bomb 0 0  power up  bomb 1 1  power down  bomb 1 3 | Increased bomb power to 1  5 points  Increased bomb power to 2  27 points  Decreased bomb power to 1  4 points |
| 1 1  5  4  power up  power up  power up  bomb 0 0 | Increased bomb power to 1  Increased bomb power to 2  Increased bomb power to 3  5 points |