**Problem 5 – Paint Ball**

You are given a painting canvas of size 10 x 10, divided into 100 cells. Initially, the canvas is white (all cells have a value of **1**). You shoot black and white paint balls with different sizes at the canvas. White is represented by **1**s and black is represented by **0**s. You alternate between black and white paint after each shot; the first shot is always with black paint (**0**s), the second is white (**1**s), the third is black again and so on. You will be given each shot's impact row and column coordinates as well as the ball's radius. The impact area is a square, its center is the impact cell; all cells in the impact area change values to either **0** or **1**, depending on the color of the paint.

After you run out of ammo (when you receive the string "**End**" from the console) the canvas will be some combination of 1s and 0s. Each row of the canvas represents a binary integer number. Your task is to find the **sum of the 10 numbers** and print it to the console. An example where a single shot with parameters "4 5 2" was fired is shown below. The impact cell is shaded black, the splashed cells in the impact area are shaded grey.

**Input**

The input data is read from the console.

* It consists of a **random number of lines**. The input **ends with the string "End"**.
* Each line will hold **three numbers** – the **row and column** of the cell where the ball lands and the **radius of the ball**, all separated from each other by a single space.

The input data will always be valid and in the format described. There is no need to check it explicitly.

**Output**

The output data must be printed on the console.

* On the only output line you must print the **sum of the 10 rows of the canvas in decimal format.**

**Constraints**

* The **number of shots** will be in the range [1…25].
* The **rows** and **columns** are integer numbers in the range [0…9].
* The **radius of the ball** will be an integer between 0 (single cell) and 10 (large splash area damage).
* Time limit: 0.25 seconds. Allowed memory: 16 MB.

**Examples**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Number |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1023 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1023 |
| 2 | 1 | 1 | **0** | **0** | **0** | **0** | **0** | 1 | 1 | 1 | 775 |
| 3 | 1 | 1 | **0** | **0** | **0** | **0** | **0** | 1 | 1 | 1 | 775 |
| 4 | 1 | 1 | **0** | **0** | **0** | **0** | **0** | 1 | 1 | 1 | 775 |
| 5 | 1 | 1 | **0** | **0** | **0** | **0** | **0** | 1 | 1 | 1 | 775 |
| 6 | 1 | 1 | **0** | **0** | **0** | **0** | **0** | 1 | 1 | 1 | 775 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1023 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1023 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1023 |
|  |  |  |  |  |  |  |  | **sum =** | | | **8990** |

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 5 2  End | 8990 |
|  |  |
| **Input** | **Output** |
| 1 2 5 | 5118 |
| 3 3 1 |  |
| 0 6 4 |  |
| 0 0 0 |  |
| 8 9 2 |  |
| 1 7 2 |  |
| End |  |