D. Слоны и ладьи

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

На шахматной доске стоят слоны и ладьи, необходимо посчитать, сколько клеток не бьется ни одной из фигур.

Шахматная доска имеет размеры 8 на 8. Ладья бьет все клетки горизонтали и вертикали, проходящих через клетку, где она стоит, до первой встретившейся фигуры. Слон бьет все клетки обеих диагоналей, проходящих через клетку, где он стоит, до первой встретившейся фигуры.

Формат ввода

В первых восьми строках ввода описывается шахматная доска. Первые восемь символов каждой из этих строк описывают состояние соответствующей горизонтали: символ B (заглавная латинская буква) означает, что в клетке стоит слон, символ R — ладья, символ \* — что клетка пуста. После описания горизонтали в строке могут идти пробелы, однако длина каждой строки не превышает 250 символов. После описания доски в файле могут быть пустые строки.

Формат вывода

Выведите количество пустых клеток, которые не бьются ни одной из фигур.

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*R\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\* | 49 |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*B\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\* | 54 |

Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| \*\*\*\*\*\*\*\*  \*R\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*B\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\* | 40 |

using System;

class Program

{

static void Main()

{

string[,] array = new string[8, 8];

string buf;

for (int i = 0; i < 8; i++)

{

buf = Console.ReadLine();

for (int y = 0; y < 8; y++)

{

array[i, y] = buf[y].ToString();

}

}

for (int x = 0; x < 8; x++)

{

for (int y = 0; y < 8; y++)

{

buf = array[x, y];

if (buf == "B" || buf == "R")

{

if (buf == "B")

{

for (int ii = 1; ii < 8; ii++)

{

if (x + ii > 7 || y + ii > 7) break;

if (array[x + ii, y + ii] == "B" || array[x + ii, y + ii] == "R") break;

if (array[x + ii, y + ii] == "\*") array[x + ii, y + ii] = "l";

}

for (int ii = 1; ii < 8; ii++)

{

if (x + ii > 7 || y - ii < 0) break;

if (array[x + ii, y - ii] == "B" || array[x + ii, y - ii] == "R") break;

if (array[x + ii, y - ii] == "\*") array[x + ii, y - ii] = "l";

}

for (int ii = 1; ii < 8; ii++)

{

if (x - ii < 0 || y - ii < 0) break;

if (array[x - ii, y - ii] == "B" || array[x - ii, y - ii] == "R") break;

if (array[x - ii, y - ii] == "\*") array[x - ii, y - ii] = "l";

}

for (int ii = 1; ii < 8; ii++)

{

if (y + ii > 7 || x - ii < 0) break;

if (array[x - ii, y + ii] == "B" || array[x - ii, y + ii] == "R") break;

if (array[x - ii, y + ii] == "\*") array[x - ii, y + ii] = "l";

}

}

if (buf == "R")

{

for (int ii = 1; ii < 8; ii++)

{

if (x + ii > 7) break;

if (array[x + ii, y] == "B" || array[x + ii, y] == "R") break;

if (array[x + ii, y] == "\*") array[x + ii, y] = "l";

}

for (int ii = 1; ii < 8; ii++)

{

if (x - ii < 0) break;

if (array[x - ii, y] == "B" || array[x - ii, y] == "R") break;

if (array[x - ii, y] == "\*") array[x - ii, y] = "l";

}

for (int ii = 1; ii < 8; ii++)

{

if (y + ii > 7) break;

if (array[x, y + ii] == "B" || array[x, y + ii] == "R") break;

if (array[x, y + ii] == "\*") array[x, y + ii] = "l";

}

for (int ii = 1; ii < 8; ii++)

{

if (y - ii < 0) break;

if (array[x, y - ii] == "B" || array[x, y - ii] == "R") break;

if (array[x, y - ii] == "\*") array[x, y - ii] = "l";

}

}

}

}

}

int answer = 0;

for (int x = 0; x < 8; x++)

{

for (int y = 0; y < 8; y++)

{

if (array[x, y] == "\*") answer++;

}

}

Console.WriteLine(answer);

}

}