Java-题目描述：

学校组织活动，将学生排成一个矩形方阵。请在矩形方阵中找到最大的位置相连的男生数量。这个相连位置在一个直线上，方向可以是水平的、垂直的、呈对角线的或者反对角线的。

注：学生个数不会超过 10000.

输入描述：

输入的第一行为矩阵的行数和列数，接下来的 n 行为矩阵元素，元素间用“,”分隔。

输出描述：

输出一个整数，表示矩阵中最长的位置相连的男生个数。

补充说明：

示例 1

输入：

```
3,4
F,M,M,F
F,M,M,F
F,F,F,M
```

输出：

```
3
```

说明：

```
import java.util.Scanner;

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        String[] split = s.split(",");
        int n = Integer.parseInt(split[0]);
        int m = Integer.parseInt(split[1]);
        String[][] str = new String[n][m];
```

```
for (int i = 0; i < n; i++) {
	String s1 = sc.nextLine();
	String[] split1 = s1.split(",");
	for (int j = 0; j < m; j++) {
		str[i][j] = split1[j];
	}
}
int res = 0;
for (int i = 0; i < n; i++) {
	for (int j = 0; j < m; j++) {
		if ("M".equals(str[i][j]) ) {
			int left = 1;//
			int leftLine = 1;
			int up = 1;//
			int upLine = 1;
			int leftUp = 1;//
			int leftUpLine = 1;
			int rightUp = 1;//
			int rightUpLine = 1;
			//左边情况
			while (j - leftLine >= 0 && "M".equals(str[i][j - leftLine])) {
				left++;
				leftLine++;
			}
			//上边情况
			while (i - upLine >= 0 && "M".equals(str[i - upLine][j])) {
				up++;
				upLine++;
			}
			//左上
			while (j - leftUpLine >= 0 && i - leftUpLine >= 0 &&
			"M".equals(str[i - leftUpLine][j - leftUpLine]) ){
			leftUp++;
			leftUpLine++;
		}
		//右上
		while (j + rightUpLine < m && i - rightUpLine >= 0 &&
				"M".equals(str[i - rightUpLine][j + rightUpLine])) {
			rightUp++;
			rightUpLine++;
		}
		int max1 = Math.max(left, up);
				int max2 = Math.max(leftUp, rightUp);
				int max = Math.max(max1, max2);
```

```
                                    res = Math.max(max, res);
                }
            }
        }
        System.out.println(res);
        }
}
```