

Java

```
class Matrix3D {  
  
    public Matrix3D(int n) {  
  
    }  
  
    public void setCell(int x, int y, int z) {  
  
    }  
  
    public void unsetCell(int x, int y, int z) {  
  
    }  
  
    public int largestMatrix() {  
  
    }  
}  
  
/**  
 * Your Matrix3D object will be instantiated and called as such:  
 * Matrix3D obj = new Matrix3D(n);  
 * obj.setCell(x,y,z);  
 * obj.unsetCell(x,y,z);  
 * int param_3 = obj.largestMatrix();  
 */
```

JavaScript

```
/**
 * @param {number} n
 */
var Matrix3D = function(n) {

};

/**
 * @param {number} x
 * @param {number} y
 * @param {number} z
 * @return {void}
 */
Matrix3D.prototype.setCell = function(x, y, z) {

};

/**
 * @param {number} x
 * @param {number} y
 * @param {number} z
 * @return {void}
 */
Matrix3D.prototype.unsetCell = function(x, y, z) {

};

/**
 * @return {number}
 */
Matrix3D.prototype.largestMatrix = function() {
```

```
};

/**
 * Your Matrix3D object will be instantiated and called as such:
 * var obj = new Matrix3D(n)
 * obj.setCell(x,y,z)
 * obj.unsetCell(x,y,z)
 * var param_3 = obj.largestMatrix()
 */
```

TypeScript

```
class Matrix3D {
    constructor(n: number) {

    }

    setCell(x: number, y: number, z: number): void {

    }

    unsetCell(x: number, y: number, z: number): void {

    }

    largestMatrix(): number {

    }
}

/**
```

```
* Your Matrix3D object will be instantiated and called as such:  
* var obj = new Matrix3D(n)  
* obj.setCell(x,y,z)  
* obj.unsetCell(x,y,z)  
* var param_3 = obj.largestMatrix()  
*/
```

C++

```
class Matrix3D {  
public:  
    Matrix3D(int n) {  
  
    }  
  
    void setCell(int x, int y, int z) {  
  
    }  
  
    void unsetCell(int x, int y, int z) {  
  
    }  
  
    int largestMatrix() {  
  
    }  
};
```

```
/**  
* Your Matrix3D object will be instantiated and called as such:  
* Matrix3D* obj = new Matrix3D(n);
```

```
* obj->setCell(x,y,z);
* obj->unsetCell(x,y,z);
* int param_3 = obj->largestMatrix();
*/
```

C#

```
public class Matrix3D {

    public Matrix3D(int n) {

    }

    public void SetCell(int x, int y, int z) {

    }

    public void UnsetCell(int x, int y, int z) {

    }

    public int LargestMatrix() {

    }

}

/**
 * Your Matrix3D object will be instantiated and called as such:
 * Matrix3D obj = new Matrix3D(n);
 * obj.SetCell(x,y,z);
 * obj.UnsetCell(x,y,z);
```

```
* int param_3 = obj.LargestMatrix();  
*/
```

Kotlin

```
class Matrix3D(n: Int) {  
    fun setCell(x: Int, y: Int, z: Int) {  
    }  
    fun unsetCell(x: Int, y: Int, z: Int) {  
    }  
    fun largestMatrix(): Int {  
    }  
}
```

```
/**  
 * Your Matrix3D object will be instantiated and called as such:  
 * var obj = Matrix3D(n)  
 * obj.setCell(x,y,z)  
 * obj.unsetCell(x,y,z)  
 * var param_3 = obj.largestMatrix()  
 */
```

Go

```
type Matrix3D struct {  
  
}  
  
func Constructor(n int) Matrix3D {  
  
}  
  
func (this *Matrix3D) SetCell(x int, y int, z int) {  
  
}  
  
func (this *Matrix3D) UnsetCell(x int, y int, z int) {  
  
}  
  
func (this *Matrix3D) LargestMatrix() int {  
  
}  
  
/**  
 * Your Matrix3D object will be instantiated and called as such:  
 * obj := Constructor(n);  
 * obj.SetCell(x,y,z);  
 * obj.UnsetCell(x,y,z);  
 * param_3 := obj.LargestMatrix();  
 */
```

