

1. Diseñe un **subprograma** que permita determinar el número de cadenas de  $n$  bits que no contienen dos "unos" consecutivos. Por ejemplo hay 5 cadenas de 3 bits que cumplen la condición: 000, 001, 010, 100, 101.

(06 puntos)

2. Diseñe un algoritmo que empleando **subprogramas** permita resolver el siguiente problema: El alcalde de la ciudad contrató a Bob el constructor para que tape los agujeros que hay en una carretera. La carretera tiene un largo  $N$  y un ancho  $M$ . Bob dividió esta carretera en cuadrados tal que el resultado fue una rejilla de  $N \times M$ .

Ayuda a Bob a saber cuántos huecos tendrá que tapar, un hueco es un conjunto de cuadrados imperfectos conectados, se dice que dos cuadros o casillas están conectadas si comparten un lado en común. (Ver el ejemplo).

**Entrada**

$N, M$ , el largo y ancho de la carretera.

Seguidamente de  $N$  filas cada una con  $M$  caracteres. Si el carácter es 'X' significa que la sección está bien. Si el carácter es 'O' significa que hay un problema en la sección.

**Salida**

La cantidad de huecos que tendrá que tapar Bob el constructor.

**Restricciones**

$1 \leq N, M \leq 100$

**Ejemplo**

Entrada  
6 10

XXXXXXX  
XOXOOXXXX  
XOXOOXXXX  
XOXOXOXOX  
XXXXXOOXX  
OOOOXXXX

Salida  
4 huecos

**Huecos en el ejemplo**

X X X X X X X X X  
X O X O O O X X X X  
X O X O O O X X X X  
X O O X X X X O X O  
X X X X X O O O X X  
O O O O O O X X X X

(07 puntos)

3. Diseñe un **subprograma** que reciba una cadena que contiene letras y dígitos, luego el subprograma debe realizar las siguientes tareas:

- Devolver la cadena compuesta solo por letras.
- Devolver los dígitos ordenados en forma ascendente.

**Ejemplo:**

Uno 1 es el primer 1 número 2 y luego viene 2, pero 4 no se puede 5 sumar 3.

**Cadena:** uno es el primer número y luego viene, pero no se puede sumar.

**Dígitos ordenados:** 1122345.

(07 puntos)