



Tarea 6

Análisis y Diseño de Algoritmos

Instrucciones para envío en Blackboard

Guarda tus archivos con el nombre según la siguiente regla:

- M<matrícula>pX.cpp, donde <matrícula>corresponde a los 6 dígitos de su matrícula UDEM y X al número de programa.
- Ejemplo: Si tu matrícula es 123456, el archivo para el problema 2, se deberá llamar M123456p2.cpp o .java
- Incluye el Código de Honor en cada programa.
- Si se realizan los programas en pareja, sólo un estudiante deberá subir el archivo nombrado con su matrícula. Importante poner **ambas matrículas y nombres** en el código fuente y/o en los comentarios al subir los programas en Blackboard.
- ****Nota: Sino sigues esta regla, tus programas NO serán calificados.**

Resolver los siguientes problemas de programación. Se sugiere, pero no se limita a resolverlos con la técnica de algoritmos voraces.

Problem 1. *Hormigas Aztecas*

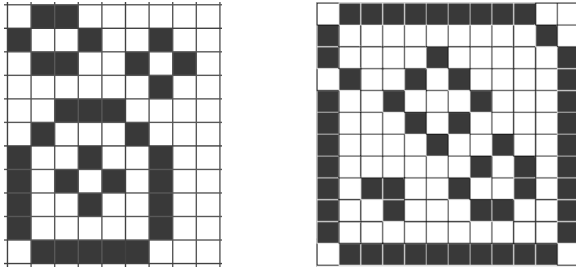
Standard Input

Time limit: 3 seconds

Carlos es un estudiante de biología, y en su proyecto final está estudiando hormigueros que incluyen colonias de una nueva especie de hormigas descubiertas en México, conocidas como hormigas aztecas que tienen la características de formar siempre colonias como ciclos cerrados. Carlos tiene una serie de fotografías que han sido procesadas previamente en imágenes de blanco y negro para mostrar visualmente cada hormiga individual como un pixel negro que se junta con otras para formar una colonia.

Carlos sabe que eres un experto en programación y te ha pedido que le ayudes creando un programa que pueda contar los ciclos cerrados en un arreglo rectangular de píxeles blancos y negros, ya que cada ciclo cerrado cuenta como una colonia de hormigas azteca.

Algo interesante y que te puede servir es que estas hormigas siempre están en contacto con exactamente otras 2 hormigas, pero nunca se tocan ni se superponen con las de otras colonias, pero es posible que debido a comportamientos particulares de estas hormigas, algunas colonias estén dentro de otras. A modo de ejemplo observa las siguientes imágenes, donde cada una contiene exactamente cuatro colonias.



Input

La primera línea de entrada contiene el número de casos k , a continuación, para cada caso, hay dos enteros f y c , ($1 \leq f, c \leq 150$), seguido por f líneas, cada una conteniendo c caracteres, donde '@' denota una hormiga (píxel negro) y '.' denota un píxel blanco. Recuerda que cada hormiga es adyacente a otras 2 hormigas, y nunca hay hormigas abandonadas.

Output

Para cada caso imprime un número entero que represente el número de colonias en cada imagen.

Sample Input

```
2
11 9
.@@.....
@..@..@..
.@@..@.@.
.....@..
..@@@....
.@...@...
@..@..@..
@..@..@..
@..@..@..
@..@..@..
@.....@..
.@@@@@...
12 12
.@@@@@@@@@@..
@.....@.
@....@.....@
.@..@..@...@
@..@...@...@
@...@..@...@
@....@..@...@
@.....@..@..@
@..@@..@..@..@
@..@...@@...@
@.....@
.@@@@@@@@@@@@.
```

Sample Output

```
4
4
```

Problem 2. *Descomponiendo cadenas*

Standard Input

Time limit: 3 seconds

Dado un string s , ¿cuál es el número máximo de subcadenas únicas que se pueden obtener a partir de él?

El string s puede dividirse en cualquier cadena (excepto la vacía), cumpliendo 2 condiciones:

- Las subcadenas deben ser únicas.
- La concatenación de las subcadenas debe formar el original.

Por ejemplo, sea $s = \text{"cocoss"}$, se puede decomponer como "coco" y "ss" y tendríamos 2 subcadenas únicas, pero con un poco de análisis podemos obtener "c", "o", "co", "s", "ss", que son 5 subcadenas únicas.

Observa que si partimos a s como "c", "o", "c", "o", "s", "ss", obtendríamos 6 subcadenas, pero estaríamos violando la regla de que las subcadenas deben ser únicas, estamos repitiendo "o".

Input

La entrada consiste de varios casos de prueba. Cada línea es una cadena s que tendrás que procesar. La longitud de s es entre 1 y 18.

Output

Para cada caso de prueba, imprime el número el número máximo de subcadenas únicas que se pueden obtener

Sample Input

```
cocoss
xx
UDEMxUEM
```

Sample Output

```
5
1
7
```