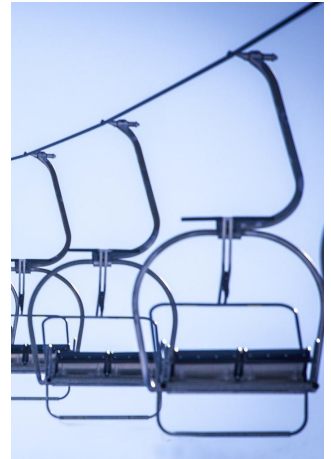


## Ejercicio 1. Algoritmos voraces (2 puntos)

### Telesillas

La estación de esquí *Es Nou Parc*, con dos pequeñas pistas, un telearrastre e incluso un telesilla, acaba de abrir. Por desgracia, la falta de experiencia de la constructora ha hecho que el telesilla instalado tenga una inquietante limitación de peso. Aunque cada silla tiene espacio para dos personas, sus usuarios no pueden superar (entre los dos) un peso máximo, o se corre el riesgo de que la barra que une la silla con el cable se parta.

A primera hora, el número de usuarios esperando para subir a la cima es enorme y el tiempo de espera en el acceso al telesilla es un cuello de botella para el disfrute de la nieve. Para solucionarlo, hay que mejorar el uso de los recursos, minimizando el número de sillas (dobles) que deben usarse para que todos los usuarios suban hasta la cima. Como ayuda, conocemos el peso de cada uno de ellos y están dispuestos a reordenarse para que se consiga reducir el tiempo de espera total.



### Entrada

La entrada del programa contendrá múltiples casos de prueba. Cada uno comienza con una línea con dos números mayores que cero. El primero indica el máximo peso que soporta cada silla (de dos plazas), y el segundo la cantidad de usuarios que desean utilizarlo.

A continuación, vendrá una línea con los pesos de todos esos usuarios, separados por un espacio. Se garantiza que el peso soportado por cada silla será como mínimo el mayor de tales pesos.

### Salida

Para cada caso de prueba, el programa escribirá el número mínimo de viajes (sillas dobles) que se necesitan para que todos los usuarios del telesilla suban hasta la cima.

### Entrada de ejemplo

```
20 4
10 10 10 10
30 5
12 20 10 16 8
40 6
30 30 30 30 30 30
```

### Salida de ejemplo

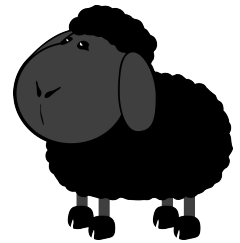
```
2
3
6
```

## Ejercicio 2. Grafos y estructuras de partición (3 puntos)

### Ovejas negras

Parece ser que en cierta ocasión estaban de viaje por Escocia un abogado, un físico y un matemático. Por la ventanilla del tren en el que viajaban vieron un campo con ovejas negras. Ninguno de los tres había visto ovejas negras nunca, por lo que se estableció un curioso diálogo entre ellos:

— ¡Vaya! ¡En Escocia las ovejas son negras! —dijo el abogado.  
— Querrás decir que en Escocia *algunas* ovejas son negras... —corrigió el físico.  
— Bueno, —no pudo evitar decir el matemático— con lo que hemos visto lo único que podemos decir es que en Escocia algunas ovejas son negras... *¡por un lado!*



Para comprobar si el abogado tenía razón o no, se han tomado fotografías a todas las ovejas de Escocia, y ahora hay que analizarlas para ver si aparecen ovejas blancas (al menos por un lado).

### Entrada

La entrada estará compuesta por distintos casos de prueba, cada uno siendo una instantánea de una o más ovejas escocesas. Cada foto comienza con una línea con dos números indicando el ancho y el alto (en píxeles) de la imagen. A continuación viene la imagen en blanco y negro en donde el carácter ‘.’ representa el color blanco y ‘X’ el negro.

Se puede asumir que: el fondo de la imagen es siempre blanco; todas las ovejas tienen la silueta negra; las ovejas blancas tienen todo blanco en su interior; las ovejas nunca se solapan (es decir, en las fotos las ovejas nunca se tocan); ninguna oveja entra en contacto con los bordes de la foto (es decir, en todas las fotos la primera y última fila y la primera y última columna son ‘.’); y en la foto solo aparecen ovejas.

### Salida

Para cada caso de prueba se escribirá el número de ovejas blancas que aparecen en la foto.

### Entrada de ejemplo

```
16 8
.....
.....XXX...
.....XXXXXXXXX...
.....XXXXXXXXX.....
.....XXXXX.....
.....X...X.....
.....X...X.....
.....
22 8
.....
.....XXX..XXX.....
..XXXXXXXXX..XXXXXXXXX..
.X....X.....XXXXXXXXX.
..XXXXX.....XXXXX..
..X...X.....X...X..
..X...X.....X...X..
.....
```

## Salida de ejemplo

```
0
1
```

## Ejercicio 3. Programación dinámica (3 puntos)

### Aibofobia

La *aibofobia* es el miedo o fobia a los palíndromos, esas palabras que se leen igual de izquierda a derecha que de derecha a izquierda. Una forma de combatirla es enfrentarse, de manera gradual y progresiva, a los temidos palíndromos.



Para ello, un controvertido ejercicio consiste en transformar cualquier palabra en un palíndromo, a base de añadir las letras que hagan falta en cualquier posición. Para medir la idoneidad de una palabra para el tratamiento nos fijamos en el menor número de letras que hacen falta añadir para convertirla en un palíndromo.

Por ejemplo, para convertir la palabra **roda** en un palíndromo podemos añadir tres letras al final, para formar el palíndromo **rodador**; para convertir **amad** solamente hace falta añadir una letra al principio para conseguir el palíndromo **damad**; y para convertir **reconor** hace falta insertar dos letras para conseguir el palíndromo **reconocer**.

### Entrada

La entrada estará formada por una serie de casos. Cada caso consiste en una línea con una palabra con caracteres en minúsculas del alfabeto inglés.

### Salida

Para cada caso se escribirá el menor número de letras que hace falta añadir (en cualquier posición y no necesariamente consecutivas) para convertir la palabra dada en un palíndromo.

### Entrada de ejemplo

```
roda
amad
reconor
reconocer
```

### Salida de ejemplo

```
3
1
2
0
```

# Normas de realización del examen

1. Debes programar soluciones para cada uno de los tres ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc/domjudge/team>.
2. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
3. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas al juez.
4. Puedes descargar el fichero <http://exacrc/DocumSEP.zip> que contiene material que puedes utilizar para la realización del examen (diapositivas de clase, implementación de las estructuras de datos, una plantilla de código fuente y ficheros de texto con los casos de prueba de cada ejercicio del enunciado).
5. Los ficheros con las implementaciones de las estructuras de datos están instalados en el juez, por lo que no es necesario subirlos como parte de tu solución (y conviene no hacerlo).
6. Los ejercicios están identificados con el nombre del tema de la asignatura en el que habrían aparecido si hubieran sido propuestos como parte de los ejercicios de la evaluación continua. Para obtener la máxima puntuación, las soluciones deberán seguir los criterios exigidos a los ejercicios de ese tema durante el curso (en cuanto a encapsulación, eficiencia, simplicidad, análisis de costes, etc.).
7. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.
8. Las notas de los ejercicios suman 8 puntos. La calificación obtenida en este examen será sumada a la obtenida por la evaluación continua (sobre 2 puntos) para obtener la calificación final de la asignatura.