



**Competencia Ibero-Americana de Informática por Correspondencia**

**CIIC 2014**

**Tareas**

Junio de 2014

LA PRUEBA TIENE UNA DURACION DE 5 HORAS

## Resumen

Problema	Selfies	Contraseñas	Ciclorutas	Sumas
Límite de memoria	256MB	64MB	64MB	64MB
Límite de tiempo (Intel Core i5 @ 2,5GHZ)	35 segundos	5 segundos	3 segundos	0.5 segundos
Puntaje Máximo	100	100	100	100

- Los programas siempre deben leer de la entrada estándar (*stdin*), y escribir en la salida estándar (*stdout*). Todas las líneas de la entrada y de la salida deben terminar con un único carácter de fin de línea (ver código ejemplo).
- Los límites dados son para cada ejecución individual.
- Las soluciones se pondrán a prueba en Linux.

## Lenguajes de Programación

### Lenguaje C

Extensión: .C

Compilador: GCC

Versión de Compilador: 4.3.2

Línea de comando: “gcc -02 -lm prog.c”

Código ejemplo (leer dos números separados por un espacio e imprimir su suma)

```
#include <stdio.h>
int main(){
    int a, b;
    scanf ("%d %d", %a, %b);
    printf("%d\n", a+b);
    return 0;
}
```

### Lenguaje C++

Extensión: .C

Compilador: GCC

Versión de Compilador: 4.3.2

Línea de comando: “g++ -02 -lm prog.c”

Código ejemplo (leer dos números separados por un espacio e imprimir su suma)

```
#include <stdio.h>
int main(){
    int a, b;
    std::cin >> a >> b;
    st::cout << a+b << std::endl;
    return 0;
}
```

## Lenguaje Pascal

Extensión: .pas

Compilador: FreePascal

Versión de Compilador: 2.2.4

Línea de comando: “fpc -02 prog.pas”

Código ejemplo (leer dos números separados por un espacio e imprimir su suma)

```
program sum;
var
  a, b: integer;
begin
  read(a, b);
  writeln(a+b);
end.
```

**Nota Importante:** Teniendo en cuenta que la competencia se realizará en diversas fechas en diversos países, está estrictamente prohibido hacer público de ninguna manera cualquier tipo de información sobre los problemas. Se informará cuando todos los países ya hayan participado en la competencia, y sólo entonces se podrá publicar la información.

# Selfies

Nombre del archivo fuente: `selfies.c`, `selfies.cpp`, o `selfies.pas`

Mario está en el Parque Nacional del Café. Este parque tiene varias atracciones mecánicas y varias lagunas. Aunque Mario no se anima a montarse en las atracciones, quiere tomarse fotos con algunas de ellas. Para cada una de las atracciones que él ha elegido, si se toma una foto muy cerca de ella, la atracción no sale completa y si la toma muy lejos sale muy pequeña. Mario es perezoso y quiere caminar lo menos posible dentro del parque para tomarse las fotos. En su desplazamiento él puede entrar a las atracciones aunque no se suba a ellas, pero no puede atravesar las lagunas.

## Tarea

Dada una descripción del parque y de de las atracciones en el parque con las que Mario quiere fotografiarse, hacer un programa que determine el desplazamiento mínimo que debe hacer Mario. Mario siempre empieza su recorrido en la casilla  $(1, 1)$  y termina una vez haya tomado todas las fotos. No es necesario que vuelva a la posición inicial.

## Entrada

- Línea 1: Tres enteros  $n$ ,  $h$  y  $w$ ,  $n$  es el número de atracciones que quiere fotografiar Mario,  $h$  y  $w$  representan respectivamente el número de filas y de columnas que tiene un arreglo que representa el parque.
- Líneas  $2 \dots h + 1$ :  $w$  caracteres que describen una fila del arreglo ('X' es una casilla de laguna y '.' es una casilla por donde Mario puede transitar).
- Líneas  $h + 2 \dots h + n + 1$ : Cuatro enteros  $x_i$ ,  $y_i$ ,  $min_i$ ,  $max_i$ . ( $x_i$ ,  $y_i$  es la posición de la atracción  $i$ ,  $min_i$  y  $max_i$  representan respectivamente la distancia mínima y máxima a las que debe estar Mario para tomar la foto a la atracción  $i$ ).

La distancia que usa Mario es la distancia Manhattan, esto es  $d((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2|$ . La longitud del recorrido de Mario es el número de celdas que él visita. Siendo coherente con su gusto con la distancia Manhattan él únicamente se desplaza de una celda a otra vertical u horizontalmente, no en diagonal. Mario no puede salir del parque para tomar una foto.

## Salida

- Línea 1: El mínimo número de movimientos que Mario debe hacer para tomarse todas las fotos o -1 si Mario no puede tomarse tomase las fotos.

## Restricciones

- $1 \leq n \leq 10$ .
- $1 \leq w \times h \leq 10^5$ .
- Para toda atracción  $i$ ,  $1 \leq x_i \leq h$ ,  $1 \leq y_i \leq w$ ,  $0 \leq min_i \leq max_i \leq 10^5$ .
- la casilla  $(1, 1)$  nunca será parte de una laguna.

## Calificación

- Para un conjunto de pruebas totalizando 20% del puntaje posible,  $n = 1$ .
- Para un conjunto de pruebas totalizando 20% adicional del puntaje posible, no hay lagunas y para toda atracción  $i$ ,  $\min_i = \max_i = 0$ .
- Para un conjunto de pruebas totalizando un 20% adicional del puntaje posible  $h \times w \leq 100$ .

## Entrada Ejemplo 1

```
2 3 5
.....
.XXXX
.....
1 5 0 0
3 5 0 2
```

## Salida Ejemplo 1

4

## Explicación de la Salida del Ejemplo 1

Basta que Mario se mueva a la casilla (1, 5) para tomar ambas fotos que quiere tomar.

## Entrada Ejemplo 2

```
2 8 15
...X.....
.XXX.....
.XX.....
.XX.....
.XX.....
.....XX...
.....
.....
5 5 2 3
6 13 1 1
```

## Salida Ejemplo 2

18

# Contraseñas

*Nombre del archivo fuente:* `contrase.c`, `contrase.cpp`, o `contrase.pas`

Rafael está administrando usuarios y contraseñas de la CIIC, él asigna una contraseña a cada nuevo usuario. El genera una cadena de longitud  $l$  usando letras minúsculas del alfabeto inglés. Algunos de los usuarios podrían sentirse ofendidos con algunas subcadenas que aparezcan en la contraseña que ellos consideren vulgaridades.

## Tarea

Ayude a Rafael a contar cuántas contraseñas puede formar que no contengan ninguna de las subcadenas catalogadas como ofensivas. Por ejemplo, la palabra 'gtabc' contiene la subcadena 'ab', pero no la subcadena 'ac'. Todas las palabras están en el alfabeto inglés que contiene 26 letras de la 'a' a la 'z'. Como el número de palabras que se pueden formar puede ser muy grande calcúlelo módulo  $10^9 + 7$ .

## Entrada

- Línea 1:  $l$  y  $n$ ,  $l$  es la longitud de las contraseñas y  $n$  es el número de subcadenas a evitar.
- Líneas 2... $n+1$ : La línea  $i+1$  contiene la subcadena  $i$  a evitar. Las subcadenas están en minúsculas y solo contienen letras en inglés de la 'a' a la 'z'.

## Salida

- Línea 1: Número de contraseñas de tamaño  $l$  que se pueden formar sin tener como subcadena a ninguna subcadena a evitar módulo  $10^9 + 7$ .

## Restricciones

- $1 \leq l \leq 1000$ .
- $1 \leq n \leq 100$ .
- $1 \leq \text{longitud de una subcadena a evitar} \leq 100$ .

## Calificación

- Para un conjunto de pruebas totalizando 25% del puntaje posible  $n = 1$ .
- Para un conjunto de pruebas totalizando un 30% adicional del puntaje posible la longitud de cada subcadena a evitar va ser a lo más 2.

## Entrada Ejemplo 1

```
10 26
ab
ba
c
d
e
f
```

g  
h  
i  
j  
k  
l  
m  
n  
o  
p  
q  
r  
s  
t  
u  
v  
w  
x  
y  
z

### Salida Ejemplo 1

2

#### Explicación de la Salida del Ejemplo 1

En este caso las únicas contraseñas posibles son 'aaaaaaaaa' y 'bbbbbbbbbb'.

### Entrada Ejemplo 2

7 1  
ciic

### Salida Ejemplo 2

31739817

# Ciclorutas

*Nombre del archivo fuente: ciclas.c, ciclas.cpp, o ciclas.pas*

Jaime salió elegido como gobernador de del departamento de Cundinamarca. Anteriormente él había sido alcalde de Bogotá. Siendo alcalde convirtió varias calles en peatonales. Ahora como gobernador desea convertir algunas carreteras en ciclorutas para que los ciclistas las usen sin ser molestados por los automóviles. Jaime piensa que con esta medida va a ganar votos para su próxima re-elección, pero teme que pierda los votos de aquellos automovilistas a los cuales se les alargue el tiempo mínimo que pueden gastar para ir de alguna ciudad a otra. El quiere saber cuántas carreteras puede convertir en ciclorutas sin perder electores.

## Tarea

Ayude a Jaime a contar cuántas carreteras puede convertir en ciclorutas sin disminuir ningún tiempo mínimo para ir de alguna ciudad a otra.

## Entrada

- Línea 1: Dos enteros  $n$  y  $m$ .  $n$  es el número de ciudades de Cundinamarca,  $m$  es el número de carreteras. Las ciudades están convenientemente numeradas de 1 a  $n$ .
- Línea 2.. $m + 1$ : Tres números  $x$ ,  $y$  y  $d$ , indicando que hay una carretera de  $x$  a  $y$  con longitud  $d$ . Las carreteras tienen una sola dirección.

## Salida

- Línea 1: La cantidad de carreteras que se puede convertir en ciclorutas sin aumentar ningún tiempo mínimo para ir de alguna ciudad a otra.

## Restricciones

- $2 \leq n \leq 400$ .
- $0 \leq m \leq n \times (n - 1)$ .
- La longitud de cada carretera será un entero entre 1 y  $10^6$  inclusive.
- Entre cada dos ciudades hay a lo más una carretera en cada dirección y no hay ninguna carretera de una ciudad a si misma.

## Calificación

- Para un conjunto de pruebas totalizando 15% del puntaje posible,  $n \leq 20$ .
- Para un conjunto de pruebas totalizando 40% del puntaje posible,  $n \leq 40$ .
- Para un conjunto de pruebas totalizando 60% del puntaje posible,  $n \leq 100$ .

### Entrada Ejemplo 1

```
7 8
1 2 1
1 3 100
3 2 42
4 5 2
4 6 50
5 6 2
5 7 10
6 7 5
```

### Salida Ejemplo 1

```
2
```

### Explicación de la Salida del Ejemplo 1

En este caso podemos convertir en ciclorruta las carreteras que van de 4 a 6 y de 5 a 7.

### Entrada Ejemplo 2

```
3 6
1 2 1
1 3 1
2 3 1
2 1 2
3 1 2
3 2 2
```

### Salida Ejemplo 2

```
0
```

# Sumas

*Nombre del archivo fuente:* `sumas.c`, `sumas.cpp`, o `sumas.pas`

Diego es muy bueno programando, pero más bien malito para hacer sumas a mano. Él se olvida de los acarreos al hacer la sumas a mano. Mario le ha puesto a Diego como tarea una suma de  $m$  sumandos de números de  $n$  dígitos. Mario quiere saber cuántas secciones contiguas de esta suma le dan a Diego un resultado dado

## Tarea

Ayude a Mario a contar cuántas secciones contiguas de la suma que le dió a Diego le dan un resultado dado.

## Entrada

- Línea 1: Dos enteros  $m$  y  $n$ .
- Líneas  $2 \dots m + 1$ : Cada línea tiene un número con  $n$  dígitos, representando uno los números de la suma propuesta por Mario. Puede haber números con ceros a la izquierda, pero siempre tendrán  $n$  dígitos.
- Línea  $m + 2$ : un número de  $n$  dígitos que es la suma que se quiere obtener.

## Salida

- Línea 1: El número de segmentos contiguos cuya suma es el número deseado.

## Restricciones

- $1 \leq m, n \leq 1000$ .

## Calificación

- Para un conjunto de pruebas totalizando 15% del puntaje posible,  $m \leq 100$ .

## Entrada Ejemplo 2

```
5 2
11
13
28
93
99
24
```

## Salida Ejemplo 1

```
3
```

## Explicación de la Salida del Ejemplo 1

Para Diego  $24=11+13=13+28+93=11+13+28+93+99$ . Entonces él puede sumar 24 sumando los cinco números, los primeros dos, o del segundo al cuarto número.

## Entrada Ejemplo 2

6 5  
12345  
47817  
42010  
15943  
02209  
87025  
59152

## Salida Ejemplo 2

2