

Hallando el recorrido

Contribución de Natalia Pérez

Descripción del problema

La Organización Argentina de Ciclismo desea organizar una competencia en una ciudad caracterizada por su apoyo al deporte. El encargado tiene mucha experiencia en la organización de carreras y sabe que muchas veces algunos competidores tienden a reducir el recorrido para llegar antes a la meta. Es por esto que, para evitar que algún ciclista acorte tramposamente el camino, decidió poner comisarios deportivos en cada giro para registrar el paso de todos los competidores por allí.

El público sabe que el giro a velocidad es causa de caídas, por lo tanto prefiere ubicarse en estos lugares para no perderse ninguna. Por este motivo, el encargado decidió instalar allí pequeñas tribunas a fin de organizar a los espectadores. Para economizar costos se redujo el número de giros al mínimo por lo cual se elige un recorrido rectangular debido a que las calles ya tienen suficientes desniveles para hacer reñida la carrera.

No todas las esquinas son aptas para poner tribunas y el encargado recorrió la ciudad registrando aquellas válidas para ser esquina del rectángulo. El problema con el cual se enfrenta ahora es que no sabe cómo elegir un rectángulo de máximo perímetro. No duda de que ello es factible porque sin la exigencia de máximo alguna vez se organizó una carrera similar.

Se te pide que escribas una función **circuito(EO, NS, E, Esquinas)** que devuelva en un *entero* el largo del recorrido. Sus parámetros son:

EO: un *entero* que indica la cantidad de calles con sentido este-oeste de la ciudad ($2 \leq \mathbf{EO} \leq 5.000$).

NS: un *entero* que indica la cantidad de calles con sentido norte-sur de la ciudad ($2 \leq \mathbf{NS} \leq 5.000$).

E: un *entero* que indica la cantidad de esquinas aptas para colocar tribunas ($4 \leq \mathbf{E} \leq 100.000$).

Esquinas: una matriz de *enteros* de tamaño $\mathbf{E} \times 2$ con las ubicaciones (eo, ns) de las esquinas aptas. La coordenada eo

indica la intersección de la calle este-oeste ($1 \leq eo \leq \mathbf{EO}$) y la ns la intersección de la calle norte-sur ($1 \leq ns \leq \mathbf{NS}$).

Ejemplo

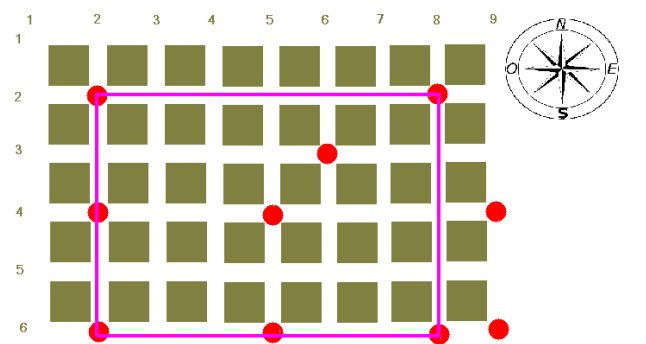
Si se presenta el siguiente caso:

EO=6; NS=9; E=10

Esquinas:

3	4	6	2	4	6	6	2	4	6
6	9	2	8	2	9	5	2	5	8

La solución deberá ser 20.



Evaluador local

El evaluador local lee la entrada por `stdin` en el siguiente formato:

- Una línea con los *enteros* EO, NS y E
- E líneas con 2 *enteros* cada una indicando las coordenadas eo y ns respectivamente de una de las esquinas aptas.

El evaluador reproducirá en una línea la cantidad de esquinas leídas y el valor retornado por la función.

El archivo **circuito.in** suministrado contiene:

```
# este es el ejemplo
6 9 10
    3    6
    4    9
    6    2
    2    8
    4    2
    6    9
    6    5
    2    2
    4    5
    6    8
```

De aportarlo como dato a un programa correcto devolvería en pantalla:

```
Archivo con 10 esquinas da largo 20.
```

Subtareas

En un número de casos por valor de puntos la cantidad de esquinas no superará a