

Due barre

Il problema

Una barra è una sequenza orizzontale o verticale di almeno 2 celle consecutive in una griglia. Due barre, una orizzontale e una verticale, sono disposte su una griglia N per N . In figura 1 le due barre sono indicate dalle **X**. Le barre possono avere o non avere la stessa lunghezza; inoltre possono avere una cella in comune. Se, in una situazione come quella di figura 1, è possibile interpretare una cella (nell'esempio, la cella (4, 4)), come appartenente o a una sola barra o a entrambe, assumeremo sempre che appartenga ad entrambe. Quindi, la cella più in alto della barra verticale in figura è la (4, 4), e non la (5, 4).

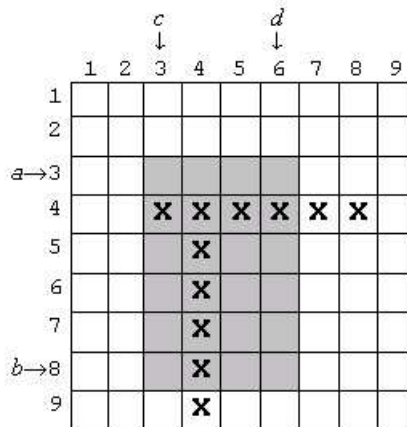


Figura 1.

Inizialmente non sappiamo dove giacciono le due barre, e quindi il vostro compito è di scrivere un programma che lo determini. Chiamiamo ROD1 la barra orizzontale, e ROD2 la barra verticale. Ogni cella della griglia è rappresentata da una coppia riga/colonna (r, c) , dove (1, 1) rappresenta l'angolo in alto a sinistra della griglia. Ogni barra è rappresentata da una coppia di celle $\langle (r_1, c_1), (r_2, c_2) \rangle$. In figura 1, ROD1 è $\langle (4, 3), (4, 8) \rangle$ mentre ROD2 è $\langle (4, 4), (9, 4) \rangle$.

Il vostro compito richiede l'uso di un insieme di funzioni di libreria per leggere l'input, per determinare la soluzione e per emettere l'output. La lunghezza del lato della griglia quadrata è fornita dalla funzione di libreria `gridsize`, che il vostro programma invocherà tipicamente all'inizio di ogni esecuzione. Per determinare

la collocazione delle barre, potete usare la funzione di libreria `rect(a, b, c, d)`, che esamina la regione rettangolare $[a, b] \times [c, d]$ (la zona ombreggiata in figura 1), dove $a \leq b$ e $c \leq d$. [Osservate attentamente in che ordine compaiono i parametri!] Se c'è almeno una cella appartenente a qualche barra dentro il rettangolo $[a, b] \times [c, d]$, la chiamata alla funzione `rect` restituirà 1; altrimenti, restituirà 0. Quindi, nell'esempio, la chiamata `rect(3, 8, 3, 6)` restituisce 1. Il vostro compito è di scrivere un programma che scopra la collocazione esatta delle barre usando un numero limitato di chiamate alla funzione `rect`.

Per emettere l'output, dovete chiamare `report(r1, c1, r2, c2, p1, q1, p2, q2)`, un'altra funzione di libreria, intendendo che ROD1 è $\langle (r1, c1), (r2, c2) \rangle$ e che ROD2 è $\langle (p1, q1), (p2, q2) \rangle$. La chiamata a `report` termina il vostro programma. Ricordate che ROD1 è la barra orizzontale e ROD2 è quella verticale, e che $(r1, c1)$ è la cella all'estremità sinistra di ROD1, mentre $(p1, q1)$ è la cella all'estremità superiore di ROD2. Quindi $r1 = r2$, $c1 \leq c2$, $p1 \leq p2$ e $q1 = q2$. Se i parametri che passate chiamando la funzione `report` non soddisfano questi vincoli, verrà segnalato un messaggio di errore sullo standard output.

Vincoli

- Potete accedere all'input esclusivamente usando le funzioni di libreria `gridsize` e `rect`.
- N , il massimo numero di righe (e colonne) della griglia di input, soddisfa la condizione $5 \leq N \leq 10000$.
- Il numero di chiamate a `rect` deve essere al massimo 400 per ogni esecuzione. Se il vostro programma effettua più di 400 chiamate a `rect`, verrà terminato.
- Il vostro programma deve chiamare `rect` più di una volta, e chiamare `report` esattamente una volta.
- Se una chiamata a `rect` ha parametri non validi (per esempio, specifica un rettangolo al di fuori dello spazio della griglia), la chiamata terminerà il programma.
- Il vostro programma non deve leggere né scrivere file, e non deve usare lo standard input o output.

Librerie

Ecco come usare le funzioni di libreria:

Libreria per il FreePascal (prectlib.ppu, prectlib.o)

```
function gridsize: LongInt;  
function rect(a,b,c,d : LongInt) : LongInt;  
procedure report(r1, c1, r2, c2, p1, q1, p2, q2 : LongInt);
```

Istruzioni: Per compilare il vostro programma `rods.pas`, includete la seguente dichiarazione di importazione:

```
uses prectlib;
```

nel sorgente, e compilate con il comando

```
fpc -So -O2 -XS rods.pas
```

Il programma `prodstool.pas` fornisce un esempio di uso della libreria per il FreePascal.

Libreria per il GNU C/C++ (crectlib.h, crectlib.o)

```
int gridsize();  
int rect(int a, int b, int c, int d);  
void report(int r1, int c1, int r2, int c2, int p1, int q1,  
            int p2, int q2);
```

Istruzioni: Per compilare il vostro programma `rods.c`, usate

```
#include "crectlib.h"
```

nel sorgente, e compilate con uno dei due seguenti comandi

```
gcc -O2 -static rods.c crectlib.o -lm
g++ -O2 -static rods.cpp crectlib.o -lm
```

Il programma `crodstool.c` fornisce un esempio di uso della libreria per il GNU C/C++.

Per il C/C++ sotto RHIDE

Selezionate correttamente `Option->Linker` su `crectlib.o`.

Esperimenti

Per fare esperimenti con la libreria, dovete creare un file di testo di nome `rods.in`. Il file deve contenere tre righe. La prima contiene un intero N , la dimensione della griglia. La seconda riga contiene le coordinate di ROD1, $r_1 c_1 r_2 c_2$, dove (r_1, c_1) è la cella all'estremità sinistra di ROD1. La terza riga contiene le coordinate di ROD2, $p_1 q_1 p_2 q_2$, dove (p_1, q_1) è la cella all'estremità superiore di ROD2.

Dopo avere fatto funzionare il vostro programma che chiama `report`, otterrete il file di output `rods.out`. Il file contiene il numero di chiamate di funzione a `rect` e le coordinate delle estremità delle barre che avete sottoposto tramite la vostra chiamata a `report`. Se ci sono errori o violazioni di vincoli durante le chiamate alla libreria, `rods.out` conterrà i corrispondenti messaggi d'errore.

Il dialogo tra il vostro programma e la libreria è registrato nel file `rods.log`. Il file mostra la sequenza di chiamate di funzioni che il vostro programma ha fatto nella forma “ $k : \text{rect}(a, b, c, d) = ans$ ”, che significa che la k -esima chiamata di funzione `rect( $a, b, c, d$ )` restituisce *ans*.

Esempio di input e output

rods.in	rods.out
9	20
4 3 4 8	4 3 4 8
4 4 9 4	4 4 9 4

Punteggio

Se il vostro programma viola uno dei vincoli (per esempio, più di 400 chiamate a funzione), o se l'output del vostro programma (la posizione delle barre) è sbagliata, otterrete 0 punti.

Se l'output del vostro programma è corretto, il vostro punteggio dipende dal numero di chiamate a `rect` per ogni test. Per ogni test, se il numero di chiamate a `rect` è al più 100, otterrete 5 punti. Se il programma chiama `rect` da 101 a 200 volte, otterrete 3 punti. Se il numero di chiamate a `rect` è tra 201 e 400, otterrete 1 punto.