

# Prova di Laboratorio di Algoritmica

Corso A e B

Simulazione 26/05/2011

## Istruzioni

Risolvete il seguente esercizio, prestando particolare attenzione alla formattazione dell'input e dell'output, in quanto la correzione avverrà in maniera automatica.

Per consegnare un elaborato dovete fornire il codice sorgente attraverso il comando `./consegna` che avete nella vostra home directory e che provvede ad inviare il vostro esercizio al server di valutazione. Il comando deve essere utilizzato dal vostro terminale nel seguente modo:

```
./consegna sorgente.c numEx
```

dove:

- `sorgente.c` è il nome del file che contiene la soluzione che avete elaborato, ricordando che il percorso deve essere specificato a partire dalla vostra home directory
- `numEx` è l'identificativo numerico dell'esercizio a cui fa riferimento la soluzione (in questa simulazione l'esercizio è solo il numero 1, per cui al posto di `numEx` dovete mettere sempre 1).

Il comando `consegna` può essere utilizzato molteplici volte, per cui è possibile sovrascrivere la propria soluzione per un dato esercizio. Di tutte le consegne per un dato esercizio, viene corretta soltanto l'ultima. Il file da consegnare deve contenere nelle prime righe un commento `C` che specifica il vostro Nome, Cognome e Numero di Matricola. Per esempio:

```
/*  
Nome: Alan  
Cognome: Turing  
Matricola: 193700  
*/
```

File non contenenti tali informazioni NON saranno ritenuti validi. Lo script di consegna, prima di inviare la vostra soluzione al server, proverà ad eseguire il vostro codice utilizzando alcuni file di input predefiniti e controllerà la correttezza dell'output prodotto. Tali file di input e output per la verifica del codice sono stati collocati nella cartella `dati` della vostra home directory. Dentro la cartella `dati` a loro volta i file di input e output per i test sono suddivisi in tante cartelle quanti sono gli esercizi. In questa simulazione abbiamo solamente l'esercizio 1, per cui dentro la cartella `dati` trovate una sola altra cartella nominata `1` dove trovate i file di input e output per effettuare il test dell'esercizio 1. Questi file sono nominati secondo lo schema: `input0.txt output0.txt input1.txt output1.txt input2.txt output2.txt ...`. Questi file possono essere usati anche da voi per testare il vostro codice e verificare poi manualmente la correttezza dell'output prodotto. Per effettuare le vostre prove potete infatti utilizzare il comando del terminale per la redirectione dell'input. Ad esempio:

```
./compilato < dati/1/input0.txt
```

effettua il test del vostro codice sui dati contenuti nel primo file di input, assumendo che `compilato` contenga la compilazione della vostra soluzione e che si trovi nella vostra home directory. Dovete aspettarvi che l'output corrisponda a quanto contenuto nel file `dati/1/output0.txt`. Per effettuare un controllo in automatico sul primo file input `input0.txt` potete eseguire i comandi:

```
./compilato < dati/1/input0.txt > res0  
diff res0 dati/1/ouput0.txt
```

Il primo comando infatti esegue la vostra soluzione e stampa l'output prodotto nel file `res0`, il secondo infine controlla le differenze fra l'output prodotto da voi e quello corretto.

La consegna andrà a buon fine solo se il vostro codice riesce a superare tutti i test contenuti nella cartella `dati`. Eventualmente lo script di consegna vi informa per quali di questi test il vostro codice non risponde correttamente. Nel caso invece la vostra soluzione passi i test, vi verrà chiesto di specificare il vostro numero di matricola e la soluzione sarà consegnata. Una volta effettuata la prima consegna, il numero di matricola verrà associato alla macchina e tramite quella macchina non sarà possibile consegnare altri elaborati se non per quello specifico numero di matricola.

Una volta consegnata, la vostra soluzione verrà testata nel server di consegna su altri file di test, da voi non accessibili. Il consiglio è quindi quello di non provare ciecamente e ripetutamente la vostra soluzione con i file di test che vi sono forniti ma cercare di ragionare sul codice che avete scritto perché il fatto che il vostro codice passi i test di consegna non significa necessariamente che questo sia corretto in assoluto.

## Esercizio 1

Si consideri il quadrante positivo del piano cartesiano. Un *punto colorato* sul piano è caratterizzato da una tripla  $(x, y, c)$ , dove  $x, y$  e  $c$  sono valori interi non-negativi. Il primo intero della tripla caratterizza l'ascissa del punto, il secondo intero l'ordinata, il terzo intero è il colore assegnato al punto.

Sia  $A$  un insieme di  $N$  punti colorati. Lo scopo del programma è quello di rispondere a una sequenza di interrogazioni sui punti di  $A$ . Un'interrogazione è definita da due coppie  $\langle (x_1, y_1); (x_2, y_2) \rangle$ , dove  $x_1 < x_2$  e  $y_1 < y_2$ , che identificano un rettangolo  $R$ :

$$R = \{(u, v) \in \mathbb{N}^2 \mid x_1 \leq u \leq x_2 \wedge y_1 \leq v \leq y_2\}$$

Data un'interrogazione  $R$  si vuole calcolare il numero di colori **distinti** dei punti di  $A$  che ricadono in  $R$  (i punti sul perimetro del rettangolo devono essere considerati nel conteggio).

Scrivere un programma che legga da tastiera una sequenza  $A$  di  $N$  punti colorati e un insieme  $Q$  di  $M$  interrogazioni, e stampi la risposta a ciascuna interrogazione su una riga distinta. Nel caso non vi siano punti all'interno del rettangolo stampare 0.

L'input è formattato nel seguente modo: le prime due righe contengono i due interi  $N$  e  $M$ , rispettivamente. Si assuma che  $N > 0$  ed  $M > 0$ . Seguono  $N + M$  righe. Le prime  $N$  righe contengono i punti colorati, uno per riga. Ogni punto è definito da 3 interi, separati da uno spazio, che rappresentano, nell'ordine, i valori  $x, y$  e  $c$ . Le ultime  $M$  righe contengono le interrogazioni, disposte una per riga. Ogni interrogazione è definita da 4 interi, separati da uno spazio, che rappresentano, nell'ordine, i valori  $x_1, y_1, x_2, y_2$ . Si assuma che  $x_1 < x_2$  e  $y_1 < y_2$ .

L'output **deve** contenere **solo e soltanto** gli interi di risposta alle interrogazioni.

**Nota:** La `scanf` considera lo "spazio" allo stesso modo del "ritorno a capo", per cui se i numeri sono contenuti in una sola linea separati dagli spazi, il comportamento della `scanf` è identico a quello che si otterrebbe se i numeri fossero disposti uno per linea.

**Nota:** È possibile consultare i manuali (in inglese) contenenti la spiegazione di funzionamento e la sintassi di funzioni di libreria utilizzando il comando `man`, ad esempio:

```
man qsort
```

.

## Esempio

### Input

```
6
4
0 0 1
6 0 1
6 1 13
1 3 8
4 4 9
4 6 137000
2 2 9 7
0 0 7 7
6 2 7 8
0 2 5 5
```

### Output

```
2
5
0
2
```