

# Laboratorio

11 Maggio 2010

*Risolvete il seguente esercizio, prestando particolare attenzione alla formattazione dell'input e dell'output, in quanto la correzione è automatica. Per consegnare un elaborato dovete fornire il codice sorgente attraverso il comando `./consegna` che avete nella vostra home directory. Il comando deve essere utilizzato nel seguente modo:*

```
/tmp/consegna fileSorgente.c numEx
```

*ricordando che il percorso deve essere specificato a partire dalla vostra home directory e `numEx` deve essere un identificativo numerico (es. 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.*

**Nota:** *E' 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 strlen
```

## Esercizio 1

Si implementi un programma che, sfruttando le nozioni di visita dei grafi vista a teoria, risolva il seguente problema su grafi. Dato  $G = (V, E)$  un grafo orientato connesso, un vertice  $v \in V$  si definisce *punto di articolazione* se la sua rimozione dal grafo aumenta il numero di componenti connesse. In altre parole, si creino  $V' = V \setminus \{v\}$  e  $E' = E \setminus \{(i, j) \text{ t.c. } i = v \vee j = v\}$ . Allora  $v$  è un punto di articolazione se  $G'$  ha più di una componente connessa. Il vostro compito è trovare tutti i punti di articolazione.

Sulla prima riga si trova il numero  $N$  di nodi del grafo seguito da  $M$  il numero di archi. I nodi si suppongono essere numerati da 0 a  $N - 1$  compresi, dove da 0 si sa che è possibile raggiungere qualsiasi altro nodo del grafo attraverso un qualche cammino. Nel file di input seguono  $M$  righe che rappresentano gli  $M$  archi. Ogni arco è denotato da una coppia di interi  $a, b$  che indicano, rispettivamente, il vertice sorgente e il vertice destinazione dell'arco.

L'output consiste di una riga, ove si devono trovare, separati da uno spazio, tutti i vertici che sono punti di articolazioni. Si noti che tali punti **devono** essere elencati in ordine crescente, e che 0 non deve essere considerato come parte dell'output.

## Esempio

### Input

```
10 13
0 1
0 2
1 3
2 3
3 4
3 5
4 4
4 6
5 6
6 7
6 8
7 9
8 5
8 9
```

### Output

## Esercizio 2

Sia  $G = (V, E)$  un grafo *non* orientato connesso. La *distanza minima* tra due vertici  $u$  e  $v$  è il numero di nodi che separano  $u$  e  $v$ . Quindi se esiste un arco  $(u, v)$  nel grafo la distanza tra  $u$  e  $v$  è 0. Scrivete un programma che, dato un grafo, calcoli il suo *diametro*, ovvero la massima distanza minima tra tutte le coppie di nodi.

L'input è uguale al precedente esercizio precedente, con l'unica accortezza che ogni arco viene descritto una volta sola. In altre parole, se trovate  $(u, v)$  nell'input, non troverete  $(v, u)$ .

L'output è costituito da un solo intero: il diametro del grafo.

## Esempio

**Input** *Vedi esercizio precedente*

**Output**

5

*Il percorso minimo più lungo è quello da 0 a 9*