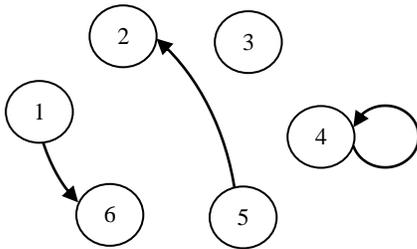


FONDAMENTI DI INFORMATICA I

FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Un Grafo è un tipo di dato astratto che implementa un *grafo orientato*, ossia una struttura matematica caratterizzata da un insieme di  $N$  nodi (che identificheremo **mediante i numeri da 1 ad  $N$** ) ed un insieme di archi orientati  $(u,v)$ , che indica la presenza di una relazione fra i due nodi  $u$  e  $v$ . I grafi orientati sono molto utilizzati in informatica in moltissime applicazioni. Per fissare le idee, vediamoli in azione per descrivere le connessioni stradali (archi) fra un insieme di città (nodi), nel seguente esempio di Grafo composto da 6 nodi e 3 archi orientati:  $(1,6)$ ,  $(5,2)$  e  $(4,4)$ .



Il fatto che nell'esempio preso in considerazione non esista l'arco  $(6,1)$  può stare a significare che, mentre la corsia dalla città 1 a 6 è aperta, quella da 6 a 1 è chiusa per lavori (lo stesso dicasi per la mancanza dell'arco  $(2,5)$ ). Infine si noti che nella nostra implementazione del tipo di dato astratto Grafo sono ammessi gli archi  $(u,u)$  (nell'esempio delle strade potrebbe essere una tangenziale). Infine precisiamo che assumeremo che fra il nodo  $u$  e il nodo  $v$  esista *al più un arco*, ossia che non vi sia più di una strada che collega due città nella stessa direzione.

Utilizzando il linguaggio C++, realizzare il tipo di dato astratto Grafo in maniera tale da permettere le seguenti operazioni:

**(PRIMA PARTE)**

- ✓ **Grafo g(k)**  
Costruttore che inizializza un Grafo di  $k$  nodi e nessun arco. Se  $k \leq 0$  viene costruito un Grafo di 5 nodi.
- ✓ **g.aggiungiArco(u,v)**  
Funzione che aggiunge un arco orientato dal nodo  $u$  al nodo  $v$ . Se l'operazione fallisce, la funzione lascia il Grafo inalterato e restituisce `false`. Restituisce `true` nei casi di successo.  **$u$  e  $v$  assumono valori fra 1 e  $N$ .**
- ✓ **g.eliminaArco(u,v)**  
Funzione che elimina l'arco  $(u,v)$  se presente e restituisce `true`. Altrimenti (arco non presente o valori di  $u$  e  $v$  errati) non altera il grafo e restituisce `false`.
- ✓ **cout << g**  
Operatore di uscita per il tipo Grafo. Mostra a video il grafo  $g$ , nella forma della seguente matrice  $N \times N$ . Se nella riga  $u$  e nella colonna  $v$  è presente una 'x', questo significa che nel grafo  $g$  è presente l'arco  $(u,v)$ .  
Esempio di uscita a video di un grafo:

```

123456
1   x
2
3
4   x
5 x
6
  
```

**(SECONDA PARTE)**

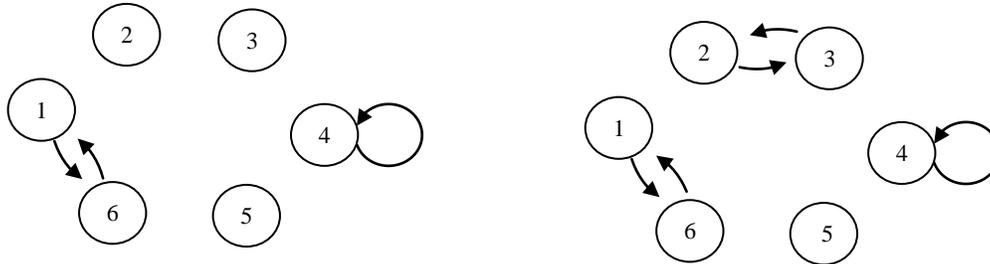
- ✓ **~Grafo**  
Distruttore della classe Grafo.
- ✓ **g%u**  
Operatore che restituisce `true` se il nodo  $u$  nel grafo  $g$  è isolato, `false` negli altri casi. Un nodo si dice isolato se non ha né archi entranti né archi uscenti né .

✓ **g.invertiArchi()**

Funzione che inverte l'orientamento di ciascun arco del grafo (gli archi  $(u,u)$  rimangono invariati).

✓ **!g**

Operatore di negazione logica che restituisce `true` se il grafo è *non orientato*, `false` altrimenti. Un grafo si dice non orientato se per ogni arco  $(u,v)$  esiste anche l'arco  $(v,u)$ . I seguenti sono due esempi di grafi *non orientati* (mentre il grafo illustrato nella pagina precedentemente era *orientato*).



## NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA

### AVVIO E IDENTIFICAZIONE

- Avviare la macchina in modalità diskless, scegliere “Fondamenti di Informatica I” ed effettuare il login:

```
nome: studenti
password: studenti
```

- Aprire un terminale e spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.

- Dare il comando `$ ident`, sempre da dentro la cartella. Lo script richiede i propri dati (cognome, nome, numero di matricola e password (la password **non va dimenticata** in quanto è indispensabile per scaricare da internet il proprio elaborato a consegna avvenuta). Il comando `ident` crea il file `matricola.txt` nella cartella corrente. Lo script può essere lanciato più volte, in tal caso il file `matricola.txt` viene sovrascritto. Per verificare che il file sia stato creato e che il contenuto sia quello giusto dare il comando (la password è codificata):

```
$ cat /home/studenti/elaborato/matricola.txt
```

- A questo punto il docente verifica che tutti gli studenti abbiamo effettuato l'identificazione, dopodichè provvede a inviare i seguenti file nella cartella `elaborato` del proprio PC: `compito.h`, `compito.cpp`, `main.cpp`.

Controllare pertanto che questi file, insieme al file `matricola.txt`, siano presenti sul proprio elaboratore.

### SVOLGIMENTO DELLA PROVA

- Definire ed implementare il tipo di dato astratto richiesto e le relative funzioni nei file `compito.h` e `compito.cpp`. Il file `main.cpp` contiene la funzione principale `main()` ed è utilizzato dallo studente per testare la sua implementazione della classe. Il file `main.cpp` può essere modificato a piacere. In sede di valutazione dell'elaborato verrà considerato **esclusivamente il contenuto dei file `compito.h` e `compito.cpp`** ed è pertanto **vietato cambiare nome a tali file**.

Per compilare e linkare dare il comando:

```
$ g++ main.cpp compito.cpp (eseguibile invocabile tramite $ ./a.out)
```

(utilizzare `g++ -g` per includere le informazioni di debug qualora si intenda debuggare con `ddd`).

### PER CONSEGNARE O RITIRARSI

Recarsi dal docente avendo preso nota dell'identificativo della macchina (g34, s23, ...).

### REGOLA PER LA CORREZIONE

Per il superamento della prova pratica è **obbligatorio** aver definito ed implementato correttamente la classe e le seguenti funzioni:

- costruttore (`Grafo g(k)`)
- aggiungi arco (`g.aggiungiArco(u,v)`)
- elimina arco (`g.eliminaArco(u,v)`)
- stampa a video (`cout<<g`)

Il codice di altre funzioni che impedisca la compilazione deve essere messo come commento. Tali funzioni verranno comunque valutate in sede di correzione della prova.