

NOME:

COGNOME:

MATRICOLA:

Si vuole realizzare un sistema per la gestione di un social network tipo Twitter rappresentato da un *grafo orientato non pesato*. I nodi del grafo sono di due tipi: tipo utente (rappresentato dal carattere 'U') rappresentato dal nickname e tipo tweet (rappresentato dal carattere 'T') rappresentato dal testo del tweet (frase di al più 80 caratteri). Gli archi del grafo rappresentano tre tipi di relazione: da utente a utente rappresentano una relazione di tipo "follow" (U1 follow U2), da tweet a utente rappresentano una relazione di tipo "ownership" (proprietà) (T owner U), da utente a tweet rappresentano una relazione di tipo "like" (U like T).

Nella cartella "19-07-18" sono presenti i file `grafo.h`, `grafo.c`, `coda.c`, `coda.h`, `node.h` che contiene il tipo di dato `node`, `graph` che contiene la struttura del grafo (la prima riga contiene il numero di nodi contenuti nel grafo, a seguire una riga per arco rappresentato dalla coppia sorgente destinazione), `node` che contiene il contenuto dei nodi (per ogni nodo, la prima riga rappresenta il contenuto, la seconda il tipo).

Testo problema	Fatto	Val.	Max
Punto 1: Creare un progetto (e il corrispondente makefile) per l'acquisizione del grafo. Il progetto include il modulo "grafo" e il file <code>compito.cc</code> . Il <code>main</code> nel file <code>compito.cc</code> deve contenere il codice per l'acquisizione del grafo orientato contenuto nel file <code>graph</code> quindi richiamare la procedura <code>stampa(graph)</code> da codificare nel file <code>compito.cc</code> che stampa gli archi del grafo.			18
Punto 2: Estendere il progetto con l'acquisizione del contenuto dei nodi: <ul style="list-style-type: none"> Estendere il file <code>compito.cc</code> aggiungendo nel <code>main</code> il codice per l'acquisizione del contenuto dei nodi dal file <code>node</code>. I nodi devono essere memorizzati in un vettore dinamico (della dimensione corrispondente al numero dei nodi) di tipo <code>node</code>. Gli indici dei nodi nel vettore sono nell'intervallo 0..n-1 per nodi con identificativo 1..n. Estendere la procedura <code>stampa</code> che diventa <code>stampa(graph, node*)</code> dove vengono stampati gli archi nel formato "verbose": <code>contenuto_nodo1 tipo_relazione contenuto_nodo2</code> (ad esempio <i>pippo FOLLOW pluto, pippo LIKE Mi piace il cioccolato,...</i>). Estendere il progetto con l'inclusione di <code>node.h</code>. 			4
Punto 3.a: Estendere il file <code>compito.cc</code> con la stampa dei <u>most influential people</u> fra gli utenti. Un utente è un most influential person se il numero dei like dei suoi tweet è il massimo rispetto a tutti gli utenti. A tale scopo: <ul style="list-style-type: none"> aggiungere al file <code>compito.cc</code> la funzione <code>int* totalLike(graph, node*)</code> che restituisce un vettore dinamico <code>V</code> della dimensione corrispondente al numero di nodi. Per ogni nodo con identificativo <code>i</code>, <code>V[i-1]</code> contiene il numero totale di like ricevuti per i suoi tweet se il nodo è di tipo utente, <code>V[i-1]=0</code> altrimenti. richiamare <code>totalLike</code> dal <code>main</code> quindi trovare il nodo o i nodi con valore massimo di like e stampare il nickname. 			5
Punto 3.b: Estendere il file <code>compito.cc</code> aggiungendo la procedura <code>follow(graph, node*, int)</code> che dati in ingresso il grafo, il vettore dei nodi e l'identificativo di un nodo utente <code>U</code> stampa l'elenco degli utenti che l'utente <code>U</code> segue direttamente o indirettamente attraverso sequenze di follow (se <code>U follow U1</code> , <code>U1 follow U2</code> allora <code>U follow U2</code> e così via). Richiamare <code>follow</code> dal <code>main</code> per un identificativo di nodo inserito da tastiera. Estendere il progetto includendo <code>coda.c</code> e <code>coda.h</code> (utile per l'implementazione di <code>follow</code>).			5
Voto			32

NOME:

COGNOME:

MATRICOLA:

NOTE

La valutazione del codice prodotto avviene al termine della prova e punto per punto (dipendenze tra i punti: 1-2-3.a, 1-2-3.b). Per ogni punto completato, è fondamentale che il codice compili e rispetti le specifiche descritte nel punto stesso.

È VIETATO l'uso di break e di variabili globali.

Si ricorda che è possibile consultare solo il materiale cartaceo/digitale del corso e i libri di testo consigliati.

Al termine della prova, creare un folder nella directory /tmp/esame/risultato etichettato con il numero di matricola. All'interno devono essere presenti tutti i file .h e .cc del progetto e il makefile. Per essere valutati, il makefile deve produrre un eseguibile funzionante.

Dati di prova:

Il **Punto 1** deve stampare l'elenco degli archi visibile nel file *graph*.

Il **Punto 2** deve stampare:

```
fede LIKE Amo la Ferrari  
fede LIKE Mi piace il cioccolato
```

```
Mi piace il cioccolato OWNER ale
```

```
Oggi piscina OWNER fede
```

```
Voglia di vacanze OWNER fede
```

```
ale FOLLOW ricky  
ale FOLLOW fede  
ale LIKE Oggi piscina  
ale LIKE Amo la Ferrari
```

```
ricky FOLLOW fede  
ricky LIKE Voglia di vacanze
```

```
Amo la Ferrari OWNER ricky
```

```
wilma FOLLOW ale
```

Il **Punto 3.a** deve stampare:

```
fede  
ricky
```

Il **Punto 3.b** dato in ingresso il nodo 8 deve stampare:

L'utente wilma segue:

```
fede  
ricky  
ale
```