

# Prova di Laboratorio di Programmazione

## Turno 1: Lunedì 25 Febbraio 2002

Scrivere un programma C `turno1.c` che esegua le seguenti operazioni:

1. Prende in input e carica in una struttura dati in memoria centrale il contenuto di due file di nome `scritto.txt` e `labo.txt`, che contengono rispettivamente i voti ottenuti nella prova scritta e nella prova di laboratorio del corso di Programmazione, sostenute da un certo gruppo di studenti. Ciascun file contiene matricola, cognome, nome e voto di ogni studente per la prova considerata, secondo il seguente formato:

```
<Matr1> <Cognome1> <Nome1> <Voto1>  
<Matr2> <Cognome2> <Nome2> <Voto2>  
<Matr3> <Cognome3> <Nome3> <Voto3>  
...
```

dove: la matricola è un numero naturale; cognome e nome sono stringhe di lettere dell'alfabeto prive di spazi (non si ammettono cognomi tipo "De Marchi"); il voto è un numero reale.

Ogni file è ordinato in modo crescente rispetto al numero di matricola.

Alcuni studenti potrebbero avere sostenuto solo una delle due prove.

2. Costruisce un elenco degli studenti che devono obbligatoriamente sostenere la prova orale di Programmazione. Uno studente deve sostenere obbligatoriamente la prova orale se ha sostenuto entrambe le prove, entrambi i voti sono superiori a 15 e almeno uno dei due voti è minore di 18 e maggiore o uguale a 15. L'elenco deve essere ordinato in modo crescente rispetto al numero di matricola.

I dati contenuti nell'elenco devono avere il seguente formato:

```
<Cognome1> <Nome1> <VotoScritto1> <VotoLabo1>  
<Cognome2> <Nome2> <VotoScritto2> <VotoLabo2>  
<Cognome3> <Nome3> <VotoScritto3> <VotoLabo3>  
...
```

3. Stampa l'elenco generato in un file di nome `Orale.txt` secondo il seguente formato:

```
<Cognome1> <Nome1> Scritto: <VotoScritto1> Labo: <VotoLabo1>  
<Cognome2> <Nome2> Scritto: <VotoScritto2> Labo: <VotoLabo2>  
<Cognome3> <Nome3> Scritto: <VotoScritto3> Labo: <VotoLabo3>  
...
```

dove le parti non comprese tra `<` e `>` sono stringhe di contenuto fisso.

Il programma deve essere realizzato utilizzando le seguenti funzioni/procedure:

1. **leggi-file** (punti 6): procedura che prende come parametro di input una stringa (nome del file) e produce come parametro (o parametri) di output l'elenco studenti/voti nella struttura dati corrispondente. Tale struttura deve essere allocata **staticamente**, scegliendo una dimensione opportuna.
2. **crea** (punti 12): procedura che prende come parametri di input due elenchi ordinati di studenti/voti e restituisce in un parametro di output l'elenco ordinato degli studenti che devono obbligatoriamente sostenere la prova orale, secondo il formato sopra descritto.
3. **scrivi** (punti 6): procedura che prende in input una stringa (nome del file), e un elenco nel formato creato dalla procedura **crea** e lo stampa sul file specificato, secondo il formato specificato sopra.
4. **main** (punti 6, inclusa la definizione delle strutture dati): si occupa solo di mettere insieme le attività richieste, utilizzando le procedure precedenti.

#### **Osservazioni:**

- Il programma non potrà contenere variabili globali. Tutti i passaggi di dati tra il main e le varie procedure e funzioni devono avvenire attraverso parametri.
- Eventuali altri parametri oltre a quelli specificati possono essere inseriti nelle procedure/funzioni se necessario.
- È ammesso utilizzare solamente le due librerie **stdio** e **stdlib**.
- È necessario consegnare solo il file sorgente, che non deve generare errori di compilazione.