

Istruzioni

- Tempo disponibile: 120 minuti.
- Non è permesso l'uso di dispositivi elettronici (a parte il PC della propria postazione).
- Il programma sarà valutato per
 - Identificazione delle strutture dati e degli algoritmi appropriati alle specifiche
 - Corretta implementazione di strutture dati e algoritmi
 - Utilizzo efficiente delle risorse
 - Stile (chiarezza, utilizzo di costrutti appropriati, corretta strutturazione)
- I programmi non compilabili saranno valutati 0 punti.
- Fare l'upload di tutti i file che compongono il programma (elencati nella sezione "Ulteriori specifiche").

Esercizio - Parte 1 (max 17 punti)

Un file binario contiene una registrazione delle ore impiegate in varie giornate da un ricercatore nell'ambito di un progetto di ricerca.

In particolare, ogni record contiene

- giorno (`int`)
- mese (`int`)
- ore impiegate (`float`)
- descrizione dell'attività svolta (stringa di 30 `char` compreso il terminatore)

I record non sono in nessun ordine particolare.

Ad esempio, il file allegato `ore.dat` contiene i dati riportati nella tabella 1.

Scrivere un programma in linguaggio C, da compilare in un eseguibile di nome `giornate`, che

- riceva come argomento della linea di comando il nome di un file del formato sopra indicato;
- stampi in output una riga per ogni data in cui il ricercatore ha lavorato, contenente giorno, mese e numero totale di ore lavorate in quella data.

Ad esempio, se `ore.dat` è il file allegato, l'invocazione `./giornate ore.dat` deve produrre un output simile al seguente:

```
8/10 7.0
8/5 3.6
26/6 9.3
14/3 2.8
6/12 5.7
13/11 7.3
23/10 7.6
6/8 2.8
30/1 1.9
7/3 2.7
```

Per rappresentare i dati relativi alle giornate è obbligatorio utilizzare una lista collegata. Si suggerisce di utilizzare un elemento della lista per ogni giornata.

Esercizio - Parte 2 (max 5 punti)

Fare in modo che l'output della parte 1 sia prodotto

- in ordine crescente di data;
- specificando, per ogni giornata, le ore totali lavorate per ogni attività, oltre a quelle totali della giornata.

Ulteriori specifiche

- La lista collegata deve essere implementata come tipo di dato astratto (in modo cioè che il programma principale acceda alla lista solo attraverso le funzioni definite nell'interfaccia della lista).
- Verificare la correttezza della linea di comando e la corretta apertura dei file; in caso di errore, stampare un messaggio e terminare l'esecuzione.
- Il programma deve essere costituito dai seguenti file:
 - `main.c` contenente (tra eventuali altre) la funzione `main`;
 - `listaGiornate.c` con la definizione delle funzioni su liste (ed eventuali altre);

- `listaGiornate.h` con le definizioni dei tipi di dato e le dichiarazioni delle funzioni definite in `listaGiornate.c` e utilizzate in `main.c`;
- `Makefile` che permetta di costruire l'eseguibile con un singolo comando `make`.

Giorno	Mese	Ore lavorate	Attività
8	10	1.3	Sviluppo
8	5	2.1	Ricerca bibliografica
26	6	0.8	Sviluppo
14	3	1.4	Sviluppo
26	6	2.7	Divulgazione
6	12	2.8	Progettazione
8	10	2.9	Divulgazione
13	11	0.8	Divulgazione
23	10	1.0	Sperimentazione
23	10	2.2	Ricerca bibliografica
13	11	2.9	Progettazione
6	8	0.7	Sperimentazione
26	6	1.1	Progettazione
8	10	2.8	Ricerca bibliografica
23	10	1.6	Progettazione
14	3	0.9	Divulgazione
13	11	1.8	Divulgazione
14	3	0.5	Ricerca bibliografica
23	10	1.0	Divulgazione
23	10	1.2	Ricerca bibliografica
6	12	1.5	Progettazione
30	1	1.9	Divulgazione
13	11	1.8	Divulgazione
8	5	1.5	Ricerca bibliografica
7	3	2.7	Divulgazione
26	6	2.1	Progettazione
23	10	0.6	Ricerca bibliografica
6	8	2.1	Sperimentazione
6	12	1.4	Sviluppo
26	6	2.6	Sviluppo

Tabella 1: Contenuto del file allegato [ore.dat](#)