

Simulazione Esame Fondamenti di Informatica - Modulo A

Testo tratto dall'esame del 17 gennaio 2011

Esercizio 1 (punti 16)

Ad un casello autostradale ci devono essere sempre almeno 4 sportelli aperti; per questo gli orari del personale sono organizzati a turni.

I turni sono riportati in un file binario [casello.bin](#). Ci sono al più 100 turni riportati. Per ciascun turno, si hanno le seguenti informazioni:

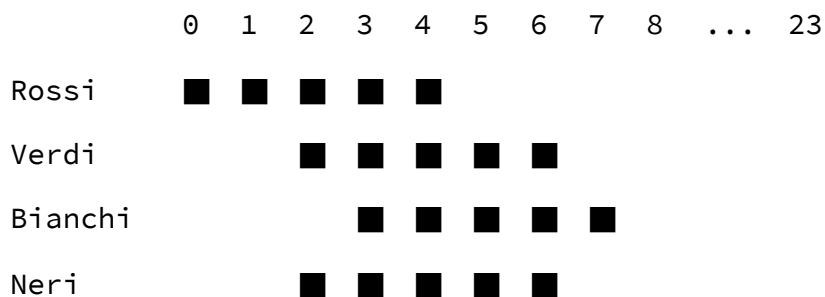
- **Nome.** nome del casellante. Stringa contenente al più 20 caratteri, compreso il terminatore
- **Inizio.** Ora in cui il casellante inizia il turno. Di tipo int, può andare da 0 a 23
- **Fine.** Ora in cui il casellante termina il turno. Di tipo int, può andare da 1 a 24

Si desidera verificare che, ad ogni ora, ci siano almeno 4 persone al lavoro. Si scriva un programma C che legge il file [casello.bin](#) e visualizzi le fasce orarie in cui ci sono meno di 4 casellanti.

Ad esempio, se il file [casello.bin](#) contiene:

```
Rossi    0 5
Verdi    2 7
Bianchi  3 8
Neri     2 7
```

vuol dire che la situazione è quella rappresentata in figura:



quindi il programma dovrà visualizzare:

```
Ore 0 numero casellanti 1
Ore 1 numero casellanti 1
Ore 2 numero casellanti 3
Ore 5 numero casellanti 3
Ore 6 numero casellanti 3
Ore 7 numero casellanti 1
Ore 8 numero casellanti 0
...
```

Per semplicità, si supponga che non ci siano turni che iniziano in un giorno e finiscono il successivo (a cavallo della mezzanotte). Si usi il seguente algoritmo:

1. Nel main, si invochi la funzione di lettura (punto 2), poi una procedura che visualizza gli orari in cui ci sono meno di 4 casellanti (punto 3).
2. Si scriva una procedura o funzione di lettura, che legge il file [casello.bin](#) e ne porta il contenuto in un opportuno array di strutture. Si visualizzi a video il contenuto dell'array.
3. Si scriva una funzione che mostra gli orari in cui non ci sono sufficienti casellanti. Tale funzione prende come parametro l'array letto al punto 2 più, eventualmente, altri parametri. Per ciascun orario del giorno (da 0 a 23) si invochi una funzione che conta il numero di casellanti presenti in tale orario (punto 4)
4. Si implementi una funzione che prende in ingresso
 - l'array letto al punto 2
 - un orario (int)

più eventualmente altri parametri e fornisce come valore di ritorno il numero di casellanti in servizio in tale orario.

Esercizio 2 (4 punti)

Si supponga ora che il numero di casellanti necessari non sia uguale in tutte le fasce orarie, ma che dipenda dall'orario. In particolare, il file binario [casellanti.bin](#) contiene una sequenza di 24 valori di tipo unsigned char; il primo elemento rappresenta il numero di casellanti necessari alle ore 0, il secondo alle ore 1, ecc.

Si modifichi il programma in modo che visualizzi gli orari in cui il numero di casellanti è insufficiente, considerando come numero minimo richiesto quello indicato nel file [casellanti.bin](#). Il programma deve funzionare correttamente anche nel caso in cui un turno inizi in un giorno e termini il giorno successivo.

Qualora si svolga solo l'esercizio 1, si può svolgere tutto l'esercizio su un unico file.

Qualora si svolga anche l'esercizio 2, si consegnino i file:

- un file COGNOME.c (dove COGNOME va sostituito col cognome dello studente) che contiene il main e le funzioni usate solo nell'esercizio 1
- un file facoltativo.c che contiene il main e le funzioni usate solo nell'esercizio 2
- un file funzioni.c che contiene le funzioni comuni

più tutti i file header ritenuti necessari.

Nella correzione, il docente creerà due progetti:

- in uno, inserirà i file COGNOME.c e funzioni.c . L'eseguibile che viene creato dovrà risolvere l'esercizio 1.
- Nell'altro, inserirà i file facoltativo.c e funzioni.c . L'eseguibile che viene creato dovrà risolvere l'esercizio 2.