

### Esercizio 1 del 3 Aprile 2017 (La mezza maratona)

In molte gare di corsa i concorrenti indossano un dispositivo RFID nelle scarpe o nel pettorale. Quando il corridore attraversa uno dei sensori posizionati lungo il percorso, un computer registra il numero di gara e l'orario del passaggio. In questo modo gli organizzatori possono tener traccia in modo preciso dei tempi parziali e totali dei partecipanti.

Si consideri un sistema RFID usato per una gara di mezza maratona, cioè di 21,0975 km. Lungo il percorso sono posizionati 3 sensori: alla partenza, a 11 km dalla partenza, e al traguardo. I dati della gara sono organizzati come segue. La prima riga dell'input contiene il numero di partecipanti `n_part` (compreso tra 1 e 100) seguito dall'ora di partenza della gara rappresentata come tre `int` nel formato `HH MM SS` (ore, minuti, secondi). Le righe successive rappresentano i dati dei sensori come sequenze di `int` e contengono l'ID del sensore (0 = partenza, 1 = metà gara, 2 = traguardo), seguito dal numero di gara del partecipante (compreso tra 0 e `n_part - 1`) e l'orario di passaggio in formato `HH MM SS`. L'orario di passaggio al sensore 0 può essere diverso dall'orario ufficiale di partenza della gara. L'input termina con una riga che contiene il valore sentinella `-1`. Ad esempio, una gara con 3 partecipanti che inizia alle 8 del mattino può essere rappresentata come segue:

```
3 08 00 00
0 0 08 00 00
0 1 08 00 03
0 2 08 00 00
1 0 08 50 46
1 2 08 51 15
1 1 08 51 18
2 1 09 34 16
2 0 09 35 10
2 2 09 45 15
-1
```

Scrivere un programma che usi i dati dei sensori per identificare i partecipanti che imbrogliano, considerando i casi seguenti:

1. un partecipante prende una scorciatoia e salta un sensore;
2. un partecipante utilizza un veicolo per percorrere una parte della gara. In questo caso un segmento percorso in meno di 2:45 per km si considera sospetto.

Il programma deve stampare l'elenco dei sospetti ordinati per numero di gara, assieme al motivo. Se il partecipante `n` è sospetto, nel caso 1. deve scrivere "`n: salto sensore`" mentre nel caso 2. deve scrivere "`n: tempo sospetto`". Se non ci sono sospetti, il programma scrive "`gara regolare`".

Rappresentare i dati in maniera adatta a risolvere il problema, ed utilizzare funzioni ausiliarie per scomporre la soluzione in sottoproblemi più semplici, come ad esempio leggere l'input, stampare il risultato, trovare i sospetti, eccetera. *Dare nomi significativi alle funzioni e ai loro parametri.*

**Correttezza:** scrivere precondizioni e postcondizioni per tutte le funzioni ausiliarie. Trovare gli invarianti dei cicli di una funzione (non banale) a piacere e dimostrarne la correttezza.