

# Programmazione I

## Prova di Programmazione – 14 giugno 2011 – 2 ore 15 min

Considerare un sistema di irrigazione automatico costituito da tre sezioni. Un controllore programmabile attiva e disattiva ciascuna sezione in funzione dell'ora corrente. All'avvio del sistema, l'ora corrente è la 0 e tutte le sezioni sono spente (inattive). Partendo dal frammento di codice fornito, scrivere un programma di controllo che fornisca le seguenti funzionalità:

1. **aggiungi\_attivazione\_unica(num\_sez, ora\_activ, ora\_spegn)** Elimina ogni eventuale programmazione preesistente relativa alla sezione **num\_sez**, dopodiché aggiunge l'accensione della sezione **num\_sez** all'ora **ora\_activ**, ed il successivo spegnimento all'ora **ora\_spegn**. Tale accensione e spegnimento saranno da effettuarsi ogni giorno. **ora\_spegn** deve essere strettamente maggiore di **ora\_activ**, ed entrambe le ore devono essere numeri interi. Si può inserire ad esempio **(2, 10, 15)**, poi magari **(1, 6, 15)**.
2. **stampa\_programmazione** Stampa la programmazione corrente, scrivendo, solo per ciascuna delle ore in cui vi sono attivazioni o disattivazioni da effettuare, l'azione da compiere per ciascuna sezione. Per ciascuna di queste ore, tali azioni vanno scritte su una riga separata.
3. **salva\_programmazione** Salva la programmazione corrente in un file dal nome predefinito.
4. **carica\_programmazione** Ricarica lo stato dal file. L'eventuale stato precedente è perso.
5. **aggiungi\_attivazione(num\_sez, ora\_activ, ora\_spegn)** Aggiunge alla programmazione l'accensione della sezione **num\_sez** all'ora **ora\_activ**, ed il successivo spegnimento all'ora **ora\_spegn** (stavolta senza eliminare la precedente programmazione per la sezione **num\_sez**). Tale accensione e spegnimento saranno da effettuarsi ogni giorno. **ora\_spegn** deve essere strettamente maggiore di **ora\_activ**, entrambe le ore devono essere numeri interi, e non si può aggiungere un periodo di attivazione sovrapposto o contiguo ad un altro periodo di attivazione già inserito per la stessa sezione. Supponendo che non vi sia alcun periodo di attivazione di alcuna sezione correntemente programmato, si può inserire ad esempio **(2, 10, 15)**, poi magari **(1, 6, 15)**. A causa della presenza di questi due periodi di attivazione non si possono poi inserire, per esempio, **(2, 9, 11)**, **(2, 12, 14)** e neanche **(2, 15, 18)**.
6. **incrementa\_ora** Incrementa l'ora corrente, attivando o disattivando le sezioni in base alla programmazione. Ad esempio, in base alla programmazione riportata nel punto 1, alle ore 6 va attivata la sezione 1, alle ore 10 va attivata la sezione 2, ed alle ore 15 vanno disattivate sia la sezione 1 che la sezione 2. Simulare l'attivazione/disattivazione con la stampa del nuovo stato delle sezioni.

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne gli *overflow* dovuti a valori assoluti troppo elevati e l'inserimento di dati in formato errato da *stdin*.

---

### REGOLE

- Si può utilizzare ogni genere di manuale o di altro materiale didattico
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1 e 2. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
  - a) il programma è perfettamente funzionante in ogni sua parte
  - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati