

## Sistemi Operativi – a.a. 2019/2020

prova di laboratorio  
– 21 febbraio 2020 –

Creare un programma `morra-cinese2.c` in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

```
morra-cinese2 <numero-partite>
```

Il programma gestisce una serie di partite tra due giocatori virtuali (processi) P1 e P2 che giocano alla Morra Cinese. Il Programma creerà: due processi **P1** e **P2** che rappresenteranno i giocatori, un processo giudice **J** e un processo tabellone **T**. I primi tre processi useranno una unica coda di messaggi per comunicare tra loro; il giudice e il tabellone opereranno usando una coda FIFO.

I quattro processi opereranno secondo le seguenti regole:

- il giudice all'inizio di ogni partita chiede ai processi giocatori la propria mossa;
- ogni giocatore risponde inviando al giudice la propria mossa: S(asso) / C(arta) / F(orbici);
- ricevute entrambe le mosse, il giudice stabilisce se si tratta di una partita patta o se c'è un vincitore: nel primo caso reitera la partita, nel secondo invia attraverso la coda FIFO un carattere '1' o '2' ad indicare il vincitore della stessa;
- quando il giudice determina che sono state svolte le partite (con vincitore) richieste, indica a tutti gli altri processi, con gli specifici strumenti di IPC, di auto-terminarsi.
- il tabellone tiene traccia del numero di partite vinte dai rispettivi giocatori e, ricevuto l'indicazione di terminare, determina l'identità del vincitore o l'eventuale patta del torneo.

I processi dovranno tutti terminare spontaneamente alla fine del torneo, rilasciando correttamente le risorse di IPC persistenti.

Un possibile output del programma potrebbe essere il seguente:

```
$ morra-cinese2 9
G: inizio partita n.1
P1: mossa 'carta'
P2: mossa 'forbice'
G: partita n.1 vita da P2
T: classifica temporanea: P1=0 P2=1
G: inizio partita n.2
P2: mossa 'sasso'
P1: mossa 'sasso'
G: partita n.2 patta e quindi da ripetere
G: inizio partita n.2
P1: mossa 'sasso'
P2: mossa 'forbice'
G: partita n.2 vinta da P1
T: classifica temporanea: P1=1 P2=1

[...]

G: inizio partita n.9
P2: mossa 'carta'
P1: mossa 'sasso'
G: partita n.9 vinta da P2
T: classifica finale: P1=4 P2=5
T: vincitore del torneo: P2
```

Suggerimenti:

- usare il campo numerico tipo nei messaggi per indirizzarli al processo preposto;
- per generare una mossa casuale si può usare la chiamata `rand()`;
- la sequenza pseudo-random è deterministica e determinata dall'eventuale seme/seed impostato con `srand()`;
- per avere sequenze distinte ad ogni avviso del programma si può scegliere un seme diverso usando: `srand(time(NULL))`.

**Tempo:** 2 ore

Ricordarsi di inserire i propri dati (nome, cognome, matricola) nei commenti preliminari del codice sorgente.

Per inviare il proprio elaborato sul server è necessario utilizzare il comando **exam-box-sync**. Verrà richiesta la password associata al proprio account e verrà data una conferma all'avvenuto caricamento. E' possibile, e fortemente consigliato, inviare il proprio elaborato più volte e periodicamente come copia di riserva (l'ambiente di lavoro degli esami risiede in memoria RAM e è pertanto di tipo non-persistente).