Sistemi Operativi – a.a. 2019/2020
prova di laboratorio
– 29 gennaio 2020 –

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

```
morra-cinese <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 **G** e un processo tabellone **T**. I tre processi useranno un segmento di memoria condiviso e un certo numero di semafori (minimo a scelta dello studente) da usare opportunamente.

Il segmento condiviso dovrà contenere i dati relativi ad una singola partita con i seguenti specifici campi:

- **mossa_p1**: S(asso) / C(arta) / F(orbici)
- **mossa_p2**: S(asso) / C(arta) / F(orbici)
- **vincitore**: 1 (giocatore 1) / 2 (giocatore 2)
- (eventuali flag ausiliari)

Iniziata una partita, P1 e P2 popolano (contemporaneamente e senza vincolo di mutua-esclusione) i relativi campi con una propria mossa casuale; ognuno, fatta la mossa, deve segnalare la cosa al processo G che valuterà chi ha vinto, se c'è un vincitore allora G popolerà il campo apposito e segnalerà la disponibilità di un nuovo esito al processo T; se invece la partita è patta (stessa mossa), allora lancerà direttamente una nuova partita. Il processo T, in caso di vittoria, aggiornerà la propria classifica interna e avvierà, se necessario, una nuova partita. Sempre T, alla fine della serie di partite, decreterà l'eventuale processo vincitore.

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-cinese 9

P1: mossa 'carta'
P2: mossa 'forbice'
G: partita n.1 vita da P2
T: classifica temporanea: P1=0 P2=1
P2: mossa 'sasso'
P1: mossa 'sasso'
G: partita n.2 passa e quindi da ripetere
P1: mossa 'sasso'
P2: mossa 'forbice'
G: partita n.2 vinta da P1
T: classifica temporanea: P1=1 P2=1

[…]

P2: mossa 'carta'
P1: mossa 'sasso'
G: partita n.10 vinta da P2
T: classifica finale: P1=4 P2=5
T: vincitore del torneo: P2
```

Suggerimenti:

- 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 diverse ad ogni avviso del programma si può scegliere un seme diverso usando: srand(time(NULL)).

Tempo: 2 ore e 15 minuti

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

Verrà valutata anche l'efficienza computazionale delle soluzioni algoritmiche utilizzate.

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).