

nome e cognome \_\_\_\_\_ numero di matricola \_\_\_\_\_  
(numero di matricola completo: 10 cifre numeriche)

CORSO DI SISTEMI OPERATIVI  
CORSO DI LAUREA IN INFORMATICA - UNIVERSITA' DI BOLOGNA  
PRIMO APPELLO DELLA SESSIONE AUTUNNALE AA 2000/2001  
05 settembre 2001

Esercizio -1. Essersi iscritti per sostenere questa prova.

Esercizio 0. Scrivere correttamente il proprio nome e cognome e numero di matricola in tutti i fogli.

Esercizio 1. In un sistema di elaborazione si è deciso di contraddistinguere le attese in attese maschili e attese femminili. Vengono dette attese con genere.

La classe di attese con genere *g* deve fornire i seguenti metodi:

`g::delay(gender mf)`

`g::request(xgender xmf)`

`delay` viene usato da un processo per sospendere la propria esecuzione; il campo `mf` può assumere valore `m` o `f` (attesa maschile o femminile).

`request` è utile per riattivare un processo in attesa; `xmf` può valere `m` (sblocca un processo in attesa maschile) `f` (riattiva un processo in attesa femminile) `a` (any, sblocca un processo, non importa il genere) `r` (rendez-vous: sblocca un processo `m` e uno `f`).

Il paradigma non prevede memoria: la `request` se non ci sono processi in attesa viene persa (come la `signal` dei monitor).

Solo la `request` di tipo `r` attende che vi sia almeno un processo per genere prima di proseguire.

Esercizio 2. Nel gioco della roulette un croupier accetta le scommesse e lancia una pallina per l'estrazione di uno dei 37 numeri (18 rossi, 18 neri e lo zero). I giocatori entrano nel casinò fanno le numerose scommesse e quando desiderano se ne vanno.

La vita del croupier è schematizzata dal seguente processo:

```
croupier: process
    while (true) {
        casino.mmfvj(); // mesdames et messieurs, faites vos jeux
        casino.ljsf(); // les jeux sont faits rien ne va plus
        // extract the number
        casino.pay();
    }
end.
```

La vita del cliente è la seguente:

```
customer: process
    ...
    casino.enter(name);
    while (!tired)
        casino.newround(name);
        // bet
        casino.betdone(name);
    casino.leave(name);
    ...
end
```

Le sincronizzazioni da implementare sono le seguenti:

- lo scommettitore/cliente deve aspettare che il crupier dichiari "faites vos jeux" per poter scommettere.
- il crupier deve aspettare che tutti i clienti presenti al tavolo abbiano fatto le loro scommesse (o se ne siano andati) prima di tirare la pallina e estrarre il numero.
- il cliente aspetta nella chiamata betdone che vengano pagate tutte le vincite prima di iniziare (se lo desidera) un nuovo ciclo di gioco.

Scrivere il monitor casino.

nome e cognome \_\_\_\_\_ numero di matricola \_\_\_\_\_  
(numero di matricola completo: 10 cifre numeriche)

Esercizio 3a. Avendo a disposizione una chiamata `swapx` che scambia il valore di due byte contigui in memoria in modo atomico è possibile scrivere un supporto di sezioni critiche à la `test&set`. Mostrare come.

Esercizio 3b. Con un processo gestore è possibile estendere il servizio per un numero arbitrario di processi (sapendo solamente che ogni processo ha come identificativo un numero intero compreso fra 0 e `MAXPROC-1`)?  
Se si', come? Se no, perché?