

**UNIVERSITA' DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA**  
**CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2004/2005**  
**COMPITO CONCORRENZA - 13 Gennaio 2005**

**Esercizio -1:** essersi iscritti correttamente per svolgere questa prova.

**Esercizio 0:** Su entrambi i fogli, scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

**Esercizio 1**

Dati i tre processi:

```

stampa(x):(x=A,T,R) process
while (true) {
    synchro(x);
    print(x);
}

```

scrivere la funzione *synchro()* che facendo uso di semafori consenta di avere in output una sequenza infinita di TARATATA (i.e. TARATATATARATATATARATATATARATATATARATATA.....)

**Esercizio 2.** Implementazione dell' algoritmo del Banchiere tramite monitor.

Siano date le funzioni *void allocate(resourceset x)*, *void deallocate(resourceset x)*, and *boolean safe()* che rispettivamente aggiornano la tabella registrando l'allocazione delle risorse (passate come parametro), la aggiornano registrando la deallocazione e controllano la safety della situazione.

Ogni processo per utilizzare le risorse usa il seguente protocollo:

```

banker.use(rs)
/* use rs*/
banker.release(rs)

```

Per l'algoritmo del banchiere l'allocazione che porta il sistema in uno stato unsafe (non safe) deve essere ritardata fino a che altri processi rilasciando risorse non rendono l'operazione safe. Fare attenzione a non introdurre problemi di starvation nella soluzione proposta.

Scrivere quindi il monitor denominato "banker".

(Hint: per "provare" se una allocazione del set di risorse s rende lo stato unsafe chiamate la *allocate(s)* e se *safe* fornisce risultato falso subito la *deallocate(s)*; infatti, la *deallocate()* elimina dalla tabella le modifiche fatte con la *allocate()* e quindi la *allocate()* puo' essere ritentata successivamente).

**Esercizio 3.**

Una agenzia matrimoniale decide di fornire il proprio servizio tramite message passing asincrono. Ogni cliente per trovare il/la proprio/a partner chiama la funzione *partnersearch* cosi' realizzata:

```

partnersearch() {
    asend(love4ever, mydata);
    arecv(love4ever, partnerdata);
}

```

Scrivere il processo denominato *love4ever*. In caso non vi siano persone compatibili fra quelle che hanno fatto richiesta, la richiesta viene memorizzata, se viene trovata una persona compatibile ad entrambi viene spedito il messaggio di ritorno contenete i dati del rispettivo partner.