

	Politecnico di Milano Facoltà di Ingegneria dell'Informazione <b>Sistemi Informatici</b> Appello 10 febbraio 2012	COGNOME E NOME	
	RIGA	COLONNA	MATRICOLA

- Il presente plico pinzato, composto di quattro fogli (fronte/retro), deve essere debitamente compilato con cognome, nome, numero di matricola, posizione durante lo scritto, e deve essere firmato.
- I compiti non compilati, non firmati o con fogli mancanti non saranno considerati validi e quindi non saranno corretti.
- Sarà valutato solo quanto scritto su questi fogli.
- Non è consentito consultare testi né appunti.
- Sul tavolo non devono essere presenti telefoni cellulari, né astucci, né custodie di altro tipo.

FIRMA

**Esercizio 1.a** (6 punti). Si vuole modellare tramite il formalismo degli StateCharts la produzione e la vendita in una delle migliori gelaterie e pasticcerie italiane. Ci sono due persone: Fabio, addetto alla produzione e Cecilia, addetta alle vendite; ci sono due prodotti: gelato e pasticcino.

Per ogni prodotto si deve tenere traccia della scorta in magazzino che può essere da 0 a 3 pezzi.

Se il magazzino è *pieno*, Fabio si trova nello stato *pulisce\_laboratorio*, altrimenti si trova nello stato *produce\_pasticcino* o *produce\_gelato*. Nella produzione viene prediletto l'articolo di cui è presente meno scorta in magazzino, a parità vengono prediletti i pasticcini. Una volta terminata la produzione di un pasticcino (evento *pasticcino\_prodotto*), Fabio torna nello stato *pulisce\_laboratorio*. Analogamente per la produzione del gelato. Qualora stia producendo un prodotto (p.e. gelato) e termina la scorta dell'altro prodotto (p.e. pasticcino), il lavoro viene interrotto e Fabio cambia prodotto in produzione (rispettando le precedenze introdotte prima).

Se non vi sono presenti clienti, Cecilia si trova nello stato *pulisce\_negozio*, ma all'arrivo di un potenziale cliente (evento *cliente\_entra*) si ferma ed entra nello stato *attende\_ordine*. Da qui se si verifica l'evento *ordina\_pasticcino* e in magazzino sono presenti pasticcini, lei si sposta nello stato *serve\_pasticcino*; una volta servito il pasticcino (evento *pasticcino\_servito*) si riporta nello stato *attende\_ordine*. Gli ordini di gelato sono gestiti analogamente. Qualora il cliente faccia un ordine di cui non si dispone merce in magazzino, egli può attendere che venga prodotto in laboratorio o uscire dal negozio (evento *cliente\_esce*).

Non è richiesto di modellare il pagamento della merce (si supponga che dopo essere stato servito il cliente possa uscire dal negozio).

Il modello proposto presenta problemi di ciclo? In caso affermativo si discuta il problema e si introduca una regola anti-ciclo.

**Esercizio 1.b** (4 punti). Si supponga di non poter più disporre degli eventi *pasticcino\_servito* e *gelato\_servito*. Senza introdurre ulteriori eventi, proporre una soluzione alternativa al problema precedente.



**Esercizio 2** (10 punti). Si consideri un problema di schedulazione periodica con i seguenti tre processi:

	$\tau_1$	$\tau_2$	$\tau_3$
$\Phi_i$	2	1	0
$C_i$	1	2	2
$T_i$	6	7	6

Si dimostri che è schedulabile con Rate Monotonic.

Si vuole ora introdurre un deferrable server. Si calcoli il fattore di utilizzazione massimo del server che garantisce la schedulabilità del sistema. Si determini il più piccolo periodo del server garantendo la schedulabilità del problema.

Si faccia lo stesso qualora si utilizzi un server dinamico.

Si supponga di disporre dei seguenti processi aperiodici soft real time.

	$J_1$	$J_2$
$a_i$	1	1
$C_i$	1	3

Si scheduli il problema misto con il server dinamico. Si riportino i tempi di risposta dei due processi aperiodici. Dire se i tempi di risposta sono riducibili in qualche modo.



**Esercizio 3** (6 punti). Dimostrare l'ottimalità di EDF con processi aperiodici senza vincoli di precedenza.

**Esercizio 4** (6 punti). Considerare il codice qui sotto. Assumendo che il codice interno del metodo *main* e del metodo *provastampa* sia corretto, si elenchino gli errori (se presenti) nel restante codice (comprese intestazioni) e si correggano. Si dica poi qual'è l'output a video prodotto dall'esecuzione del codice, discutendo se le ultime due righe del main producono o no stampe a video diverse e motivare la risposta.

```

public class pubblicazione {
    private String autore, titolo;
    private int anno, pagine;

    public void stampa () {
        system.println.out(this.autore);
        system.println.out(this.titolo);
        system.println.out(this.anno);
        system.println.out(this.pagine);
    }
}

public class articolo_rivista {
    static private int classe;
    private String rivista;
    private int volume, numero;

    public void stampa () {
        super.stampa();
        system.println.out(this.rivista);
        system.println.out(this.classe);
    }
}

public class articolo_congresso {
    static private int classe;
    private String congresso, sede, data;
}

public void stampa () {
    super.stampa();
    system.println.out(this.congresso);
    system.println.out(this.sede);
    system.println.out(this.data);
}

}

public void main () {
    articolo_rivista.classe = 0;
    articolo_congresso.classe = 1;
    articolo_rivista RIV1 = new articolo_rivista("gatti", "on
equilibrium computation", 2011, 10, "ACM");
    articolo_rivista RIV2 = new articolo_rivista();
    articolo_congresso CON1 = new articolo_congresso
("gattipanozzo", "new positive results on equilibrium verification",
2012, 10, "AAMAS");
    articolo_congresso articolo_congresso = new
articolo_congresso();
    RIV1.stampa();
    provastampa(RIV1);
    CON1.stampa();
    provastampa(CON1);
}

public void provastampa(pubblicazione x) {
    x.stampa();
}

```

**System.out.println**

**private static int classe**

**private static int classe**

**static**

