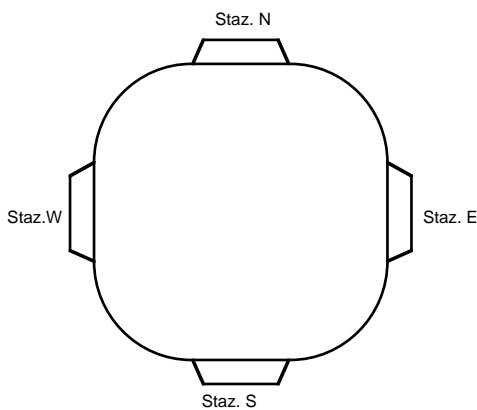


CORSO DI SISTEMI OPERATIVI  
 CORSO DI LAUREA IN INFORMATICA - UNIVERSITA' DI BOLOGNA  
 SESSIONE ESTIVA 1998/99 - PRIMO APPELLO  
 6 GIUGNO 1999

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

La città di Monopoli ha quattro stazioni ferroviarie chiamate Nord, Sud, Est, Ovest (per brevità da ora in poi le chiameremo N,S,E,W).

La linea ferroviaria è circolare a binario semplice e solo alle stazioni esiste un secondo binario per consentire sorpassi o incroci tra treni. Per motivi di sicurezza anche se due treni procedono nella stessa direzione uno solo di essi può occupare la tratta ferroviaria fra due stazioni, l'altro deve aspettare che il primo abbia raggiunto la stazione successiva per poter entrare nella tratta.



A monopoli esistono tre treni: il locale che ferma in tutte le stazioni, il treno espresso N/S e in treno espresso E/W.

Tutti i treni viaggiano nella direzione N->E->S->W (senso orario).

I treni espresso hanno la precedenza sul locale: se il locale è fermo in stazione e un espresso o è fermo nella stessa stazione o è sulla tratta ferroviaria immediatamente precedente allora il locale deve attendere che l'espresso abbia superato la stazione prima di poter ripartire.

I treni espresso si possono superare nelle stazioni (a patto che il treno locale non occupi già un binario della stazione).

Il programma che implementa i tre treni è il seguente:

```
#define LOCALE 0
#define EXPR_NS 1
#define EXPR_EW 1
```

```
locale: process
  staz.init(LOCALE,N);
  while(true)
    ... sosta in stazione...
    staz.exit(LOCALE, N);
    staz.enter(LOCALE, E);
    ... sosta in stazione...
    staz.exit(LOCALE, E);
    staz.enter(LOCALE, S);
    ... sosta in stazione...
    staz.exit(LOCALE, S);
    staz.enter(LOCALE, W);
    ... sosta in stazione...
    staz.exit(LOCALE, W);
    staz.enter(LOCALE, N);
  end while
end
```

```
expr_ns: process
  staz.init(EXPR_NS,S);
  while(true)
    ... sosta in stazione...
    staz.exit(EXPR_NS, S);
    staz.enter(EXPR_NS, W);
    staz.exit(EXPR_NS, W);
    staz.enter(EXPR_NS, N);
    ... sosta in stazione...
    staz.exit(EXPR_NS, N);
    staz.enter(EXPR_NS, E);
    staz.exit(EXPR_NS, E);
    staz.enter(EXPR_NS, S);
  end while
end
```

```
expr_ew: process
  staz.init(EXPR_EW,E);
  while(true)
    ... sosta in stazione...
    staz.exit(EXPR_EW, E);
    staz.enter(EXPR_EW, S);
    staz.exit(EXPR_EW, S);
    staz.enter(EXPR_EW, W);
    ... sosta in stazione...
    staz.exit(EXPR_EW, W);
    staz.enter(EXPR_EW, N);
    staz.exit(EXPR_EW, N);
    staz.enter(EXPR_EW, E);
  end while
end
```

1a) Sono possibili casi di deadlock? PERCHÉ?

1b) Sono possibili casi di starvation? PERCHÉ?

2) Scrivere il monitor staz.

nome e cognome \_\_\_\_\_ numero di matricola 1674 \_\_\_\_\_

3) Scrivere le funzioni staz.\* facendo uso di semafori

4a) Se il treno locale percorresse il tragitto in senso antiorario cambierebbe la situazione in merito ai casi di deadlock? PERCHÉ?

4b) E in merito ai casi di starvation? PERCHÉ?

4c) (facoltativo) Scrivere un monitor staz che risolva il problema (nel miglior modo possibile) in questo caso (con il treno locale che percorre il tragitto in senso antiorario e i treni espresso in senso orario).