

Politecnico di Milano
Facoltà di Ingegneria dell'Informazione
Informatica 3
Prof. Giovanni Agosta
Prof. Pier Luca Lanzi
Prof. Marcello Restelli

COGNOME E NOME (IN STAMPATELLO)

MATRICOLA

Appello del 2 Settembre 2008

Risolvere i seguenti esercizi, scrivendo
le risposte ed eventuali tracce di
soluzione negli spazi disponibili.

Non consegnare altri fogli.

Spazio riservato ai docenti

--	--	--	--	--

Esercizio 1.

Determinare l'output del seguente programma Python, motivandolo:

```
def f(x,y):  
    return x*2+y*2
```

```
def g(x,y=['a']):  
    return x>y or type(y)!=type(x)
```

```
a=['a',2]  
b=[1.2,'b']
```

```
print g(f(a,b))  
print f(b,a)  
print g(f(a,b),f(b,a))
```

```
def h(y):  
    def f(x):  
        return x-y  
    return f
```

```
f=h(a)  
print f(b)
```

Esercizio 2.

Il gestore di un campeggio ha installato una struttura docce costituita da due box doccia separati e due diversi ingressi alla struttura (uno maschile ed uno femminile).

Per garantire la massima privacy agli ospiti, è consentito accedere alla struttura docce solo se questa è al momento completamente vuota oppure se solo uno dei due box è occupato e la persona che lo occupa è dello stesso sesso di quella che richiede di accedere alla struttura.

Si modellizzi la struttura docce in Java come una risorsa condivisa che fornisce i seguenti metodi:

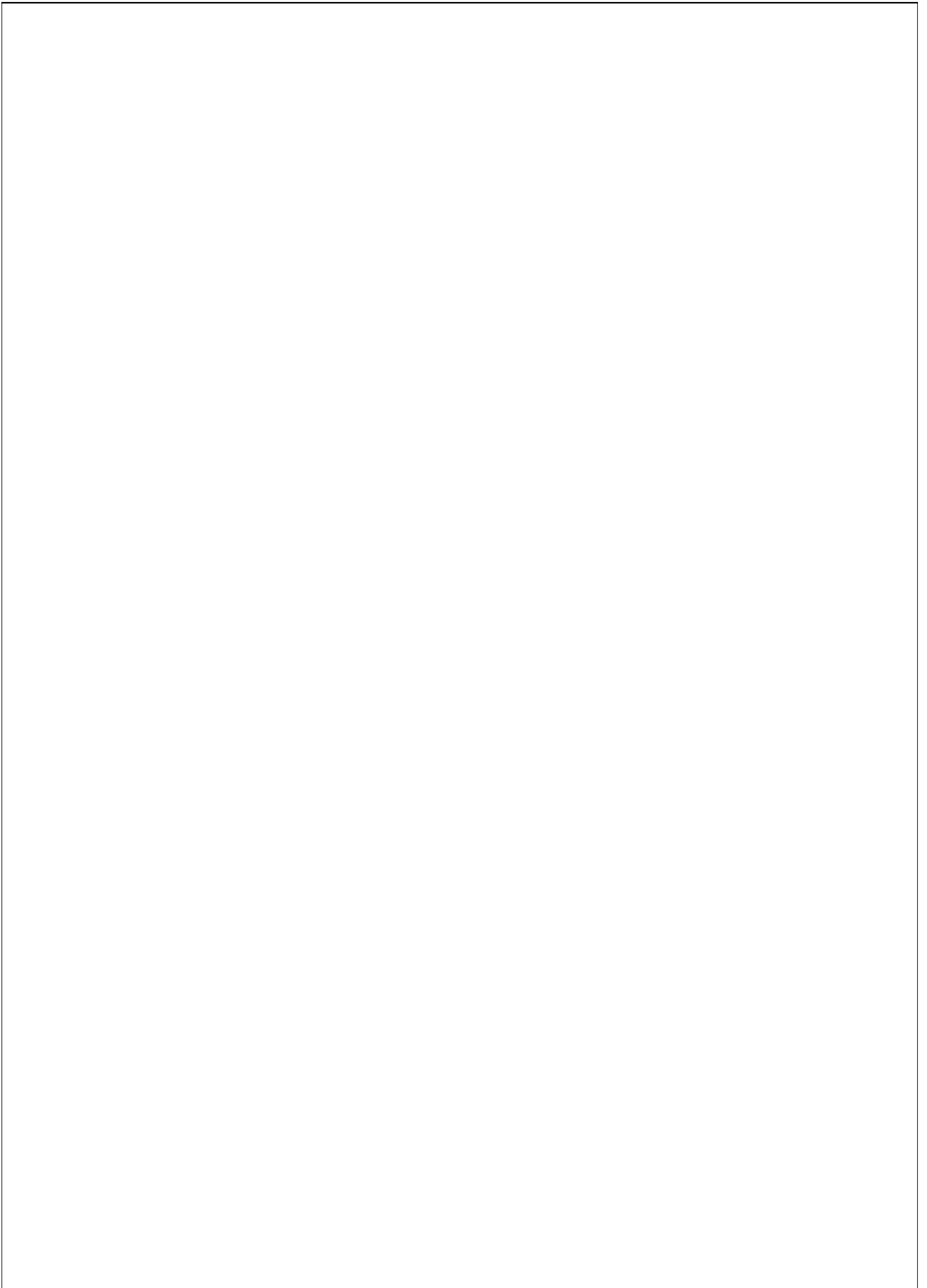
- *entra* (char sesso, string ID)
- *esci* (char sesso, string ID)

dove sesso è un char uguale a 'M' per i maschi ed 'F' per le femmine, mentre ID è un identificativo dell'ospite.

Il metodo *entra* modella la richiesta ad accedere ad un box doccia e la sua terminazione presuppone che la richiesta sia stata accettata e quindi che il box doccia sia stato occupato dal richiedente. Il metodo *esci* modella l'uscita dal box doccia e quindi il rilascio della risorsa.

Gli ospiti del campeggio possono essere considerati come dei thread che in tempo reale invocano i metodi *entra* ed *esci* (**non** è richiesto di modellizzare il thread che rappresenta l'ospite).

Le richieste di entrare nella struttura docce devono essere evase nello stesso ordine di arrivo nell'ambito dello stesso ingresso (maschile e femminile). Tutte le operazioni necessarie a gestire correttamente i problemi di concorrenza legati alla risorsa condivisa devono essere completamente trasparenti ai thread che modellizzano gli ospiti: nessuna struttura o meccanismo di controllo deve essere posto all'esterno della risorsa condivisa.



Esercizio 3. Calcolare la complessità delle seguenti procedure come funzione di n nel caso pessimo (nel worst case).

```
xxx (int : n )
{
  int i, j, k, m, n, total;
  i = 1 ;
  j = 0 ;
  total = 0 ;
  while ( i <= n ) {
    i = 4 * i;
    j = j + 2 ;
  }
  for ( m = 1 ; m <= i; m++ )
    for ( n = 1 ; n <= j ; n++ )
      total = total + n + m;
}
```

```
yyy (int: n)
{
  int i, j, x, y;
  x = 0 ;
  y = 0 ;
  i = 1 ;
  while ( i <= n ) {
    if ( i >= (n / 2))
      then for (j = i; j <= n; j++)
        x = x * 2 ;
      else for (j = 1; j <= i; j ++ )
        y = y + 1 ;
    i = i + 1 ;
  }
}
```

Esercizio 4. Si mostrino tutti i B-Alberi legali con grado minimo uguale a 2 che rappresentano l'insieme $\{1,2,3,4,5\}$.

Esercizio 5. Dati un intero k e un array $A[1..n]$ di interi positivi ordinati in ordine crescente, descrivere ed analizzare un algoritmo per trovare una coppia di posizioni distinte i, j tali che $1 \leq i < j \leq n$ e $A[i] + A[j] = k$. Il tempo di esecuzione dell'algoritmo deve essere $O(n)$ nel caso pessimo.