

Esercizio 1

Utilizzare i semafori per sincronizzare i seguenti thread in modo che stampino DOVEVEDO. (Inserire le P e le V direttamente nel codice qui sotto):

```

thread D {
  for(i=0;i<2;i++) {
    P(SemD)
    print "D";
    V(SemO);
  }
}

thread O {
  for(i=0;i<2;i++) {
    P(SemO);
    print "O";
    if (i==0) V(SemV);
  }
}

thread V {
  for(i=0;i<2;i++) {
    P(SemV);
    print "V";
    V(SemE);
  }
}

thread E {
  for(i=0;i<2;i++) {
    P(SemE);
    print "E";
    if (i==0) V(SemV);
    else V(SemD);
  }
}

```

Semafori e valori di inizializzazione: I semafori `SemD`, `SemO`, `SemV`, `SemE` sono inizializzati rispettivamente a 1,0,0,0 e regolano la stampa della rispettiva lettera.

Breve spiegazione della soluzione proposta: L'unico thread che può stampare è D in quanto il semaforo corrispondente (`SemD`) è verde. Dopo ogni stampa viene abilitato il thread successivo. Nel caso della D e della V il thread successivo è, rispettivamente, O e E, sia in DOVE che in VEDO. Nel caso del thread O non si deve sbloccare nulla quando `i == 1`, perché è l'ultima lettera a essere stampata. Il thread E sblocca V (la V di VEDO) alla prima iterazione e D (la D di VEDO) alla seconda.

Esercizio 2

Progettare un Monitor statistica che gestisca un pool di N sensori. I sensori rilevano la temperatura e la inviano al monitor tramite il metodo `notifica`. Il thread collettore legge la media delle N misurazioni e le stampa. Il metodo è bloccante e ritorna la media solo quando tutti i sensori hanno notificato le misurazioni.

```

thread sensore(i) {
  // rileva temperatura t
  statistica.notifica(i,t);
}

thread collettore {
  m = statistica.media();
  print(m)
}

```

Monitor:

```

Monitor statistica {
  float somma=0;
  int quanti_notificato=0;

  notifica(int i,float t) {
    somma += t;
    quanti_notificato++;
    if (quanti_notificato == N)
      notify();
  }

  float media() {
    while(quanti_notificato<N)
      wait();

    return somma/N;
  }
}

```

Breve spiegazione della soluzione proposta: Il metodo `notifica` colleziona le misurazioni nella variabile `somma` e conteggia quanti sensori hanno notificato in `quanti_notificato`. Quando tutti i sensori hanno notificato viene fatta una `notify()` per svegliare il thread collettore. Il metodo `media` attende che tutti abbiano notificato, calcola la media e la ritorna.