

Zadaniu się precyzyjne jest  
stoliki numerowane  
kiedy się stare pom.

**Zadanie 3. (10 pkt)** Klub brydżowy liczy  $N$  członków i ma salę z  $S$  czteroosobowymi stolikami do gry, ponumerowanymi od 1 do  $S$ . Wśród klubowiczów jest  $P$  stałych par. Pozostałych  $N - 2P$  członków klubu dobiera się w pary losowo. Stałe pary mają ustalone unikalne identyfikatory (od 1 do  $P$ ), pozostali mają identyfikator 0.

Gracz przychodzący na spotkanie klubowe, jeśli ma stałego partnera czeka na niego. Gracz bez stałego partnera zgłasza chęć gry wtedy, kiedy ma pewność, że będzie miał partnera do pary.

Gra odbywa się przy stoliku. Para zajmuje miejsca przy stoliku o najmniejszym numerze spośród dostępnych. Aby rozegrać partię przy stoliku powinny być dwie pary. Pary rozgrywają partię (predefiniowana funkcja `gramy_partie`), po czym odchodzą od stolika. Możesz założyć, że funkcja `gramy_partie` jest tak skonstruowana, że jej wykonanie kończy się jednocześnie we wszystkich grających procesach.

Korzystając z semaforów binarnych i ogólnych zapisz treść procesu `Gracz(id_pary : 0..P)`. Możesz korzystać z abstrakcyjnych struktur danych (stosy, kolejki, itp.), o ile opiszesz interfejs.

**Zadanie 4. (8 pkt)** W budynku o dwóch kondygnacjach pracują urzędnicy, którzy od czasu do czasu muszą przemieścić się na inną kondygnację. W tym celu korzystają z windy kursującej w budynku. Może ona pomieścić nie więcej niż  $M$  osób. Aby skorzystać z windy urzędnik musi poczekać na windę. Po wejściu do niej urzędnik czeka aż winda zatrzyma się na docelowej kondygnacji i wysiada. Po zatrzymaniu się windy najpierw wszyscy z niej wysiadają, a potem — w miarę wolnych miejsc — wsiadają ewentualni chętni. Winda kursuje między piętrami. Jeśli nikt nie korzysta z windy stoi ona na piętrze, na którym wysiadł ostatni pasażer.

Napisz monitor `MON` synchronizujący windę i urzędników. Algorytmy działania urzędnika i windy są następujące:

```
process Urzednik;  
var gdzie_jestem: integer;  
begin  
  gdzie_jestem := ?;  
  while true do  
    begin  
      pracuje(gdzie_jestem);  
      MON.wzywam_winde_i_jade (gdzie_jestem);  
      gdzie_jestem := 3 - gdzie_jestem;  
    end  
  end  
end;
```

```
process Winda;  
var pietro: 1..2 := 1;  
begin  
  while true do  
    begin  
      MON.dokad_jechac (pietro);  
      { ustala wartosc zmiennej pietro  
        zmien_pietro (pietro);  
      MON.przyjazd (pietro);  
    end;  
  end;  
end;
```