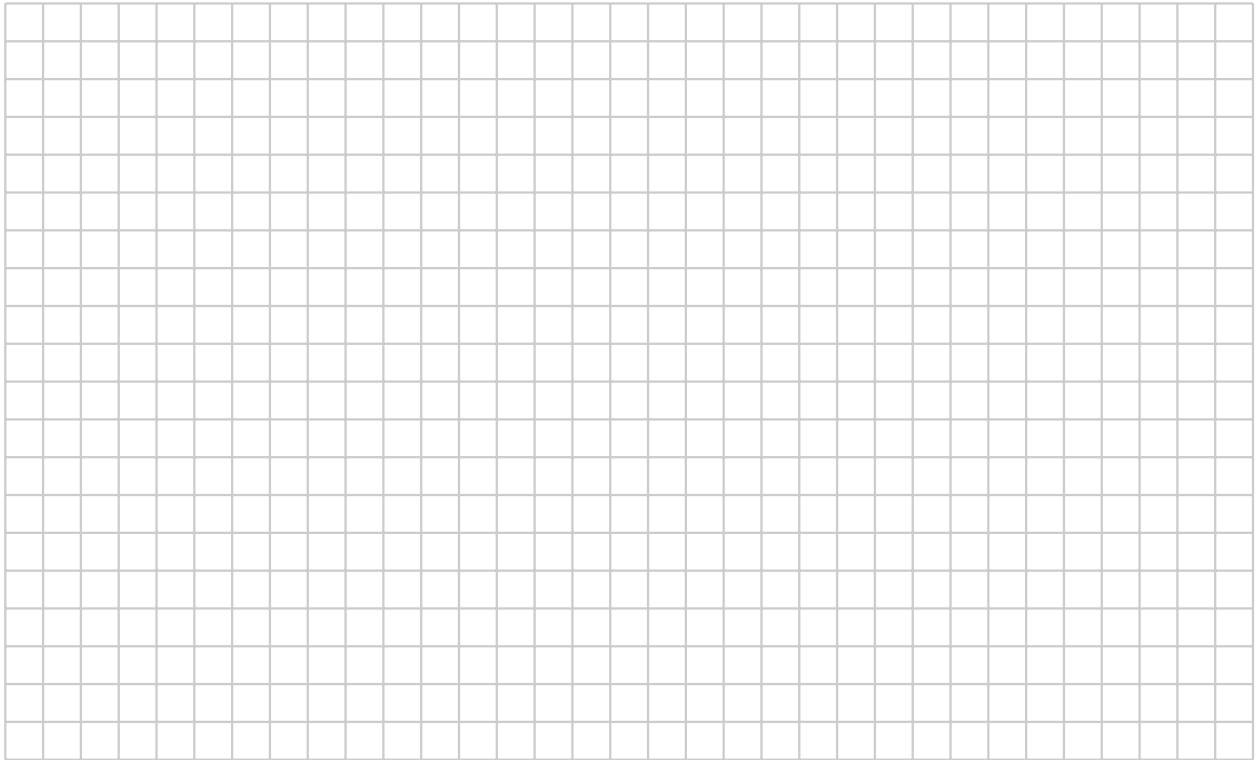




**Exercici 2 [2 punts].** Dissenyeu una aplicació basada en dos processos  $P_1$  i  $P_2$ . El procés  $P_1$  sempre escriu lletres 'A' i el procés  $P_2$  lletres 'B'. L'aplicació que es demana ha d'escriure contínuament la seqüència "ABABBABBBABBBBA...".



**Exercici 3 [2 punts].** Una forma d'implementar un semàfor binari és fer-ho usant un *spinlock*. Això és simplement emprar una espera contínua en comptes de forçar un canvi de context en l'operació de `wait()`:

```
void wait(S) {  
    while (S == 0); S=0;  
}
```

Com que `S` és una variable compartida caldria garantir que l'accés a `S` es fa en exclusió mútua. Assumint que s'usa la tècnica d'inhabilitar interrupcions i que es disposa de dues funcions `inhabilita()` i `habilita()`, quina hauria de ser la implementació de l'operació `wait()`?





**Exercici 4 [1 punt].** Considereu els processos  $P_1, P_2, P_3$ . Assumiu que tots tres processos tenen sengles regions crítiques en que accedeixen a diversos recursos compartits  $R_1, \dots, R_n$ . Segons com siguin les diferents regions crítiques, els processos podrien arribar a un *deadlock*. Esbosseu un exemple de processos que tinguin risc d'arribar a un *deadlock* en accedir als recursos. Com hauria de ser el codi de les diferents regions crítiques? Afegiu una traça d'execució que acabi amb *deadlock* per demostrar que efectivament és possible.

**Exercici 5 [1 punt].** El problema del «barber dormilega» és un problema clàssic de sincronització inventat per Edsger W. Dijkstra. Suposeu un barber que treballa en una barberia amb 4 cadires per esperar-se. El barber afaita els clients d'un en un i treballa seguint la pauta descrita a continuació.

Quan el barber acaba d'afaitar un client, si n'hi ha cap esperant-se a les cadires el fa passar i l'afaita. Altrament, es posa a dormir fins que arriba un nou client. Per un altre cantó, els clients quan entren a la barberia, si veuen cap cadira buida s'esperen. En cas contrari, se'n van sense afaitar. Inicialment, la sala d'espera és buida i el barber dorm.

Es demana que modelex aquest sistema amb dos tipus de processos: el barber i els clients. El procés barber s'executa contínuament i emula el comportament del barber. Un procés client emula el comportament d'un client que es va a afaitar i acaba o bé afaitat al cap d'una estona o bé sense afaitat per que no ha trobat lloc a la barberia. Com succeïa en el cas dels productors/consumidors, podeu representar l'estat de la sala d'espera i del barber usant semàfors.

