Un distributore automatico e' fornito di confezioni di biscotti e cracker. Dispone di 50 cestelli, ognuno dei quali può ospitare una sola confezione.

Esistono 4 categorie di processi:

* fornitori di biscotti: inseriscono una confezione di biscotti nel distributore
* consumatori di biscotti: acquistano una confezione di biscotti dal distributore
* fornitori di cracker: inseriscono una confezione di cracker nel distributore
* consumatori di cracker: acquistano una confezione di cracker dal distributore

Quando un fornitore di cracker vuole inserire una confezione e ci sono cestelli liberi, non può farlo se sono valide entrambe le seguenti condizioni:

* il numero di confezioni di cracker già presenti e' >= del numero di confezioni di biscotti già presenti
* il numero di confezioni di cracker già presenti e' >= 3.

I fornitori che tentano di rifornire prodotti ma non possono farlo devono essere messi in attesa.

I consumatori che desiderano acquistare prodotti non presenti, rinunciano (non devono essere messi in attesa).

Programmare il sistema usando i semafori con la semantica tradizionale.

Variabili:

* waitingFornBisc=0
* waitingFornCrack=0
* numBisc=0
* numCrack=0

Semafori:

* semFornBisc=0
* semFornCrack=0
* mutex=1

# FornitoreBiscotti

while(true) {

wait(mutex);

if(numBisc+numCrack==50) {

waitingFornBisc++;

signal(mutex);

wait(semFornBisc);

wait(mutex);

}

numBisc++;

signal(mutex);

// Nel caso i consumatori andassero in waiting:

wait(mutex);

if(waitConsBisc>0) {

waitingConsBisc--;

signal(mutex);

signal(semConsBisc);

}

else

signal(mutex);

}

# FornitoreCracker

while(true) {

wait(mutex);

if(numBisc+numCrack==50 || (numCrack>=numBisc && numCrack>=3)) {

waitingFornCrack++;

signal(mutex);

wait(semFornCrack);

wait(mutex);

}

numCrack++;

signal(mutex);

// Inserisco cracker

}

# ConsumatoreBiscotti

while(true) {

wait(mutex);

if(numBisc!=0) {

numBisc--;

if(waitingFornBisc>0) {

waitingFornBisc--;

signal(semFornBisc);

}

else if (waitingFornCrack>0 && (numCrack<numBisc || numCrack<3) {

waitingFornCrack--;

signal(semFornCrack);

}

}

signal(mutex);

}

# ConsumatoreCracker

while(true) {

wait(mutex);

if(numBisc!=0) {

numBisc--;

if(waitingFornBisc>0) {

waitingFornBisc--;

signal(semFornBisc);

}

else if (waitingFornCrack>0 && (numCrack<numBisc || numCrack<3) {

waitingFornCrack--;

signal(semFornCrack);

}

}

signal(mutex);

}

// un esercizio del cazzo