

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' \geq 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.

```
free = 50; // numero cestelli liberi.  
waitB = 0; // numero fornitori Biscotti in attesa.  
waitC = 0; // numero fornitori Cracker in attesa.  
totB = 0; // numero confezioni Biscotti nel distributore.  
totC = 0; // numero confezioni Cracker nel distributore.
```

```
semi = 0; // semaforo per i fornitori di biscotti.
```

```
fornitore biscotti{  
    wait(mutex);  
    if(free==0){  
        wait_B++;  
        signal(mutex);
```

```

        wait(sem_B);
        {inserisco i biscotti}
    }
    else{
        free=free-1;
        totB=totB+1;
        if(totB=totC+1 & waitC>0 & free>0){
            waitC=waitC-1;
            signal(sem_C);
            totC=totC+1;
            free=free-1;
        }
        signal(mutex);
        {inserisco i biscotti}
    }
}

fornitore cracker{
    wait(mutex);
    if(free==0 OR (totC>=totB & totC>=3)){
        wait_C = wait_C+1;
        signal(mutex);
        wait(sem_C);
        {inserisco i cracker}
    }
    else{ totC=totC+1;
        free=free-1;
        signal(mutex);
        {inserisco i cracker}
    }
}

consumatore biscotti{
    wait(mutex);
    if(totB==0){signal(mutex);"ciao ciao";}
    else{
        totB=totB-1;
        if(wait_B > 0){
            wait_B=wait_B-1;totB= totB+1;
            signal(sem_B);
        }
        else{
            if(waitC > 0 & (totC<totB || totC<=3)){
                waitC=waitC-1;totC= totC+1;
                signal(sem_C);
            }
            else{
                free=free+1;
            }
        }
        signal(mutex);
    }
}

consumatore cracker{

```

```

wait(mutex);
if(totC==0){signal(mutex);"ciao ciao";}
else{
    totC=totC-1;
    if(waitC > 0 & (totC<totB || totC<=3){
        wait_C=wait_C-1;totC= totC+1;
        signal(sem_C);
    }
    else{
        if(wait_B >0){
            wait_B = wait_B-1;totB= totB+1;
            signal(sem_B);
        }
        else{
            free=free+1;
        }
    }
    signal(mutex);
}
}

```