

Si consideri il problema dei produttori e consumatori.

Si assuma che il buffer sia un array di 10 interi gestito circolarmente.

Si assumano i seguenti vincoli aggiuntivi:

- il valore 3 non deve essere presente piu' di una volta nell'array,
- il valore 5 non deve essere presente piu' di due volte nell'array.

I processi che tentano di effettuare operazioni al momento non consentite devono essere messi in attesa.

Programmare il sistema sfruttando i semafori con la semantica tradizionale.

Una soluzione possibile.

Variabili:

```
int i = 0; // indice usato dai produttori
int j = 0; // indice usato dai consumatori
```

Semafori ed inizializzazione:

```
empty=10 // numero di posizioni libere
full=0   // numero di posizioni vuote
mutex_p=1 // m.e. per accessi all'indice i
mutex_c=1 // m.e. per accessi all'indice j
sem_3=1   // numero di occorrenze di 3 che possono essere
aggiunte
sem_5=2   // numero di occorrenze di 5 che possono essere
aggiunte
```

produttore:

```
int item;
while(true){
    .....
    item=....;
```

consumatore:

```
int item;
while(true){
    .....
    wait(full);
```

```
if(item==3){wait(sem_3);}
if(item==5){wait(sem_5);}
wait(empty);
wait(mutex_p);
buffer[i]=item;
i=i+1%10;
signal(mutex_p);
signal(full);
.....
}
```

```
wait(mutex_c);
item=buffer[j]
j=j+1%10;
signal(mutex_c);
if(item==3){signal(sem_3);}
if(item==5){signal(sem_5);}
signal(empty);
.....
}
```