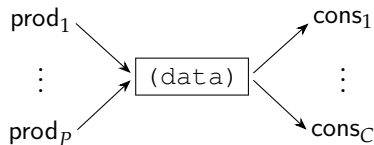


Tampon borné

On veut écrire un programme composé de P threads *producteurs* (qui fournissent des données à traiter) et de C threads *consommateurs* (qui traitent des données). Les deux types de threads communiquent via un tampon fini de taille fixée B . Chaque donnée est munie d'un identificateur unique (un nombre entier positif). Le fonctionnement est le suivant :



- chaque thread producteur produit un nombre D fixé de données ; pour chaque donnée produite, le thread lui attribue un numéro n unique globalement (obtenu par incrémentations successives d'une variable globale), la place dans le tampon et affiche un message de la forme « P i n v » où i est le numéro du producteur (entre 0 et $P - 1$), n est le numéro de la donnée et v sa valeur ;
- chaque thread consommateur obtient une donnée en la prenant dans le tampon et affiche un message de la forme « C i n v ».

Ces affichages vous permettront de tester la correction de votre programme. Par exemple :

```
> ./tampon 3 3 2 2
P 0 1 484      # le producteur p0 produit la donnée numéro 1 de valeur 484
P 1 2 713      # p1 produit la donnée numéro 2 de valeur 713
C 0 1 484      # le consommateur c0 consomme la donnée numéro 1
C 0 2 713      # c0 consomme la donnée numéro 2
P 2 3 967      # p2 produit la donnée numéro 3 de valeur 967
# note : les données 4 et 5 ont sans doute été déterminées par p0 et p1 mais ceux-ci sont en attente que p2 ait terminé
P 2 6 735      # p2 produit la donnée numéro 6 de valeur 735
# etc.
```

1. Écrivez le programme qui prend en paramètres les nombres D , P , C et B . Les données seront générées pseudo-aléatoirement avec les fonctions de bibliothèque `rand` et `srand`. Vous pouvez exceptionnellement utiliser des variables globales.
2. Pour que le système puisse s'arrêter correctement lorsqu'il n'y a plus de données et que les consommateurs puissent détecter qu'il n'est plus nécessaire d'attendre, on suggère de produire des données *fictives* (par exemple la valeur -1). Lorsqu'un thread consommateur reçoit une telle donnée, il doit s'arrêter. Vous n'oublierez pas de tester votre programme avec des valeurs de P et de C différentes.