

NB : indiquer en commentaire au début de chaque programme s'il a passé correctement le « syntax check » et si la vérification donne un résultat correct.

Exercice 1 : Election sur un réseau complètement maillé (10 points)

1.1. Coder en Promela un algorithme d'élection entre N processus reliés *en maillage complet* par des canaux deux à deux. Chaque processus reçoit un numéro différent au lancement. Sachant que tout processus P_i diffuse son numéro à tous les autres processus, codez en Promela l'algorithme distribué permettant d'élire le processus qui a le *plus petit numéro* sur le réseau. Implémenter le scénario avec $N=4$ (cf. figure 1).

NB : il ne s'agit pas de l'algorithme d'élection de Chang et Roberts vu en cours et en TP.

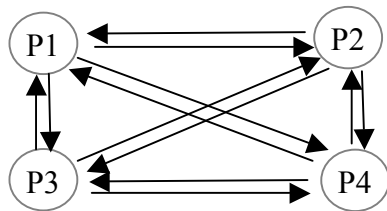


Figure 1 : canaux reliant les processus P1, P2, P3, P4

Vous pouvez utiliser la structure de données suivante :

```
#define N 4

typedef tabcan {
    chan to[N] = [1] of { short }
};

tabcan from[N]; /* utilisation de la structure tabcan permettant de définir des tableaux de canaux
à double indexation : from[x].to[y] */

proctype electeur (short numproc; chan em1, em2, em3, rec1, rec2, rec3)
    /* ... */
}

init
{
    atomic {
        run electeur(1, from[0].to[1], from[0].to[2], from[0].to[3],
                    from[1].to[0], from[2].to[0], from[3].to[0]) ;
        /* ... */
    }
}
```

1.2. Rajouter une ou plusieurs assertions qui vérifient qu'un seul processus est élu, et qu'il s'agit bien de celui dont le numéro est le plus petit. Une vérification locale au processus est-elle suffisante ? Sinon, proposez une vérification globale. Vous pouvez insérer votre réponse à la question en tant que commentaire dans le code.

Exercice 2 : La fabrique de gâteau (10 points)

Pour réaliser un gâteau, les pâtisseries devront disposer d'un élément de chacun des ingrédients suivants : farine, sucre et œuf.

Trois pâtisseries participeront à la production de gâteaux. Les trois pâtisseries sont différents, chacun disposant d'un ingrédient à volonté (un des pâtisseries possède des œufs à volonté, un autre de la farine à volonté et le troisième du sucre à volonté).

Chaque pâtissier tente de se fournir en ingrédients manquants sur la file d'attente (le canal INGREDIENT). Pour chaque ingrédient dont il ne dispose pas à l'infini, il tente de les obtenir sur le canal INGREDIENT *dans l'ordre suivant : Farine, Sucre, Œuf*.

Une fois qu'un pâtissier a les trois ingrédients (qu'il a soit obtenu sur la file d'attente soit dont il dispose à l'infini), il informe le fournisseur qu'un gâteau a été produit et que le fournisseur peut ajouter les ingrédients consommés sur la file d'attente.

Un fournisseur commence par ajouter 3 ingrédients (farine, sucre, œuf) dans le canal INGREDIENT. Il attend qu'au moins un des pâtisseries ait produit un gâteau avant d'ajouter à nouveau les ingrédients consommés sur le canal. Il a donc besoin de savoir quel pâtissier a fini un gâteau pour savoir quels ingrédients ajouter sur la file d'attente. L'information qu'un gâteau a été fini sera transmise sur un canal spécifique (canal INFO).

2.1. Définir la variable *mtype* pour le protocole décrit ci-dessus. Il y a trois types de messages ingrédient (FARINE, SUCRE, OEUF) et trois types de messages indiquant chacun qu'un des pâtisseries a fini un gâteau.

2.2. Écrire les types de processus fournisseur et pâtissier.

2.3. Vérifier votre implémentation du protocole décrit. Est-il bloquant ? Si oui, indiquer un échange de message pouvant amener à un blocage. Vous pouvez insérer votre réponse à la question en tant que commentaire dans le code.

```
mtype = { /* ... */ }

chan INGREDIENT = [3] of {byte};
chan INFO = [3] of {byte};

proctype fournisseur () { /* ... */ }
proctype patissierFarine () { /* ... */ }
proctype patissierSucre () { /* ... */ }
proctype patissierOeuf () { /* ... */ }

init {
    atomic {
        run fournisseur(); run patissierOeuf();
        run patissierFarine(); run patissierSucre();
    }
}
```

2.4. Modifier votre code dans l'hypothèse où les ingrédients ne sont pas pris par les pâtisseries dans un ordre déterminé. On rappelle qu'une fois qu'un pâtissier a les trois ingrédients (qu'il a soit obtenu sur la file d'attente soit dont il dispose à l'infini), il informe le fournisseur qu'un gâteau a été produit et que le fournisseur peut ajouter les ingrédients consommés sur la file d'attente. Vérifier votre implémentation du protocole décrit. Est-il bloquant ?