

# Lëtzebuenger Informatiksolympiad 2017

## Finale

Tous les programmes doivent être réalisés sous forme d'applications console (voir remarques sur le site [www.infosolympiad.lu](http://www.infosolympiad.lu) sous la rubrique "Les questionnaires"). Les formats des données ainsi que des résultats sont à respecter absolument.

### TÂCHE 1 [50 POINTS]

### EN ATTENDANT LE CONCERT DES ROLLING STONES



Les Rolling Stones sont attendus à la Rockhal. C'est le concert de l'année. Tout le monde veut avoir les meilleures places. À cette fin les fans vont camper devant la Rockhal déjà une semaine avant le concert.

Pour faire fonctionner le stand de vente de saucisses grillées, il est prévu de donner la concession à un groupe qui doit être formé d'exactly 10 étudiants. Mais avant de pouvoir obtenir cette concession, chaque groupe intéressé doit prouver qu'il sera capable de faire fonctionner le stand pendant une semaine entière, 24 heures sur 24.

Il est exigé que le stand soit à tout moment occupé par au moins trois étudiants. Le stand doit fonctionner pendant exactement 7 jours, de minuit à minuit, donc pendant 168 heures. Les temps d'occupation du stand sont organisés comme suit : 00:00-04:00, 04:00-08:00, 08:00-12:00, 12:00-16:00, 16:00-20:00 et 20:00-24:00. Au cours d'un tel intervalle, il ne peut pas y avoir de changement au niveau des personnes occupant le stand. Sur le formulaire de candidature de la Rockhal tout groupe intéressé doit indiquer pour chacun des 10 étudiants quelles sont les heures pendant lesquelles **il n'est pas disponible** pour être présent sur le stand.

#### Le problème

Pour chaque groupe candidat vous devez déterminer si oui ou non il pourra être capable d'assurer le stand dans les conditions exigées.

#### Les données

La première ligne contient un entier positif,  $n$  ( $1 < n \leq 20$ ), indiquant le nombre de groupes candidats à vérifier. Pour chaque groupe il y a 10 lignes de données, une ligne par étudiant membre du groupe. Chaque ligne contient les temps d'indisponibilité de l'étudiant. Leur format est le suivant :

Le premier nombre entier positif,  $l$  ( $l < 20$ ), de chaque ligne indique le nombre de périodes d'indisponibilité de l'étudiant. Chaque étudiant est indisponible pour au moins une période. Chaque période est décrite

par trois nombres entiers positifs,  $j$ ,  $d$  et  $f$ , où  $j$  représente le jour de la semaine,  $d$  représente l'heure début, et  $f$  représente l'heure fin avec  $1 \leq j \leq 7$ ;  $0 \leq d \leq 24$  et  $0 \leq f \leq 24$ .

Toutes les périodes se rapportent à un seul jour de la semaine, aucune période ne se termine à 0 heure et aucune période ne commence à 24 heures. Chaque période dure au moins une heure.

Exemple : 2 9 13 désigne une période du 2<sup>e</sup> jour de la semaine qui commence à 9 heures et se termine à 13 heures. La durée en est de 4 heures.

Tous les nombres sur une ligne sont séparés par au moins un espace.

### Les résultats

Pour chaque groupe il faut d'abord écrire "Groupe #  $i$  :", le nombre entier  $i$  étant le numéro du groupe. Le premier groupe portera le numéro 1. Ce numéro sera suivi d'un espace, d'un « : », d'un espace et puis soit de "OUI" soit de "NON", selon si le groupe est théoriquement capable d'assurer le stand ou pas.

### Exemple d'exécution

#### Données

```
2
7 1 0 24 2 0 24 3 0 24 4 0 24 5 0 24 6 0 24 7 0 24
6 1 0 24 2 0 24 3 0 24 4 0 24 5 0 24 6 0 24
5 1 0 24 2 0 24 3 0 24 4 0 24 5 0 24
4 1 0 24 2 0 24 3 0 24 4 0 24
4 1 0 24 2 0 24 3 0 24 4 0 23
1 1 0 24
2 1 10 12 2 3 7
7 1 0 24 2 0 24 3 0 24 4 0 24 5 0 24 6 0 24 7 0 24
7 1 0 24 2 0 24 3 0 24 4 0 24 5 0 24 6 0 24 7 0 24
7 1 0 24 2 0 24 3 0 24 4 0 24 5 0 24 6 0 24 7 0 24
7 1 0 1 2 0 1 3 0 1 4 0 1 5 0 1 6 0 1 7 0 1
7 1 1 2 2 1 2 3 1 2 4 1 2 5 1 2 6 1 2 7 1 2
1 2 1 2
1 3 1 2
1 4 1 2
1 5 1 2
1 6 1 2
1 7 1 2
1 2 2 3
1 2 3 4
```

#### Résultats

```
Groupe # 1 : NON
Groupe # 2 : OUI
```

#### Remarque

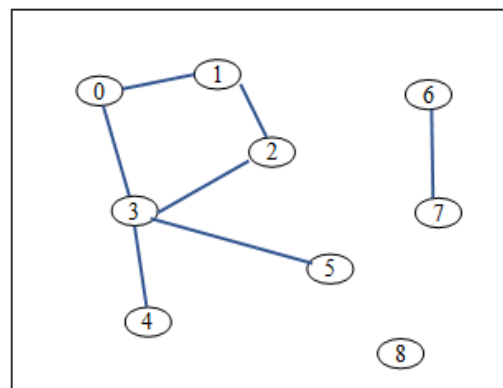
Les fichiers texte correspondant à l'exemple d'exécution sont mis à disposition dans le répertoire LIO\$\EnvoiSujets (STONES.IN et STONES.OUT).

**Remettez le programme sous le nom STONES.xxx, avec xxx = PAS ou DPR ou C(PP) ou JAVA.**

**Tâche**

Max est nouvellement responsable du placement des fibres optiques dans le pays. Il a toujours peur des ruptures de câbles causées par des travaux. C'est pourquoi il prévoit de remédier à cette situation et il veut donc connaître tous les tronçons sensibles existant entre deux localités. On entend par tronçon sensible un tronçon sans lequel deux localités ne seraient plus interconnectées.

Votre travail consiste à écrire un programme qui réalise cette tâche.

**Entrée et sortie du programme****Entrée**

Le fichier d'entrée comporte des données pour éventuellement plusieurs cas.

La première ligne de chaque cas contient un entier  $K$  désignant le nombre de villages à considérer avec  $0 \leq K \leq 1000$ . Les  $K$  lignes suivantes indiquent chacune de quelle façon un village donné est relié éventuellement à d'autres villages. Une telle ligne débute par un numéro de village  $NV$  avec  $0 \leq NV \leq K-1$ . Ce numéro de village est suivi d'au moins un espace et - entre parenthèses - du nombre de villages  $N$  auquel il est relié. Ce dernier est suivi de  $N$  entiers, séparés par un espace au moins, désignant les villages reliés. Ces entiers ne sont pas ordonnés. Il en est de même pour les  $K$  lignes.

Toutes les connexions sont évidemment bidirectionnelles. Il n'y a pas de connexion d'un village avec lui-même et il peut exister des réseaux de villages non interconnectés entre eux.

**Sortie**

Le programme affiche pour chacun des cas le nombre de tronçons sensibles suivi d'une liste de tous les tronçons en question selon le format indiqué ci-dessous. Cette liste devra être triée en ordre croissant de leur premier élément.

## Exemple d'exécution

### Entrée (3 cas)

```
9
0 (2) 1 3
1 (2) 2 0
6 (1) 7
7 (1) 6
3 (4) 4 0 5 2
4 (1) 3
5 (1) 3
2 (2) 1 3
8 (0)

0

1
0 (0)
```

### Sortie

```
3 troncons sensibles
3 - 4
3 - 5
6 - 7

0 troncons sensibles
```

### Remarque

Les fichiers texte correspondant à l'exemple d'exécution sont mis à disposition dans le répertoire LIO\$\EnvoiSujets (FIBRE.IN et FIBRE.OUT).

**Remettez le programme sous le nom FIBRE.xxx, avec xxx = PAS ou DPR ou C(PP) ou JAVA.**