

## Programmation Linéaire TP 2

---

### #1 Maximiser la disponibilité d'un service à la clientèle

Un service à la clientèle veut améliorer sa qualité en assurant une disponibilité continue afin de répondre aux différentes demandes de ses clients. Le service dispose de 6 créneaux de 4 heures, et de 60 agents.

Le tableau suivant indique pour chaque créneau le nombre minimal d'agents devant être présents.

Numéro du créneau	1	2	3	4	5	6
Heure de début	00h00	04h00	08h00	12h00	16h00	20h00
Nombre d'agents	15	7	10	20	18	22

On suppose qu'un agent doit travailler dans 2 créneaux consécutifs (i.e. 8 heures de suite). Le responsable du service souhaite déterminer le nombre minimal d'agents requis sur les 6 créneaux.

#### Réalisation :

Reprenez le PL que vous avez défini dans le TP-1. Ré-écrit ce PL avec un programme en C++ utilisant l'API `lp_solve` pour trouver le nombre minimum d'agents requis pour assurer la disponibilité du service. Votre programme doit résoudre le PL et afficher pour chaque créneau le nombre minimum d'agents.

### #2 Problème de flot minimum

Dans cet exercice, nous allons reprendre le problème du flot minimum introduit dans le TP-1. Notre objectif dans ce TP est d'écrire un programme en C++ générique qui accepte n'importe quel graphe de flot minimum et qui soit capable de générer le PL associé et de donner la solution optimale.

Une instance d'un graphe de flot minimum est fournie sur Ecampus dans le dossier TP2. Cette instance est écrite dans un fichier utilisant le format DIMACS. Une description de ce format est donnée en commentaire à l'intérieur du fichier.

#### Réalisation :

Écrivez un programme en C++ utilisant l'API `lp_solve` qui accepte un fichier contenant un graphe d'un flot minimum et qui retourne le flot minimum optimal qui traverse le graphe. Votre programme doit afficher le flot qui traverse chaque arc du graphe ainsi que le flot global généré par la source.