TP N°2 – Initiation au C
Variables, Tableaux, Structures

Attention à respecter les instructions suivantes :

1. vous **devez** utiliser l'outil `make` pour compiler vos programmes. Une introduction à cet outil est disponible sur moodle ;
2. chaque programme doit être composé **d'au moins** deux fichiers `.c` (plus les éventuels `.h`). Un fichier contenant le `main` et un ou plusieurs fichiers contenant le reste du TP ;
3. chaque fichier **doit** compiler avec les options `-Wall -Wextra -Werror`.

Si une de ces condition n'est pas vérifiée, le TP ne sera pas validé par votre enseignant.

Exercice 1 : Tableaux d'entiers

Dans cet exercice, nous allons manipuler des tableaux d'entiers. On considère, comme les compilateurs C, que le programmeur utilisant les fonctions qui doivent être écrites sait ce qu'il fait et qu'il connait parfaitement la documentation de celles-ci. Cela veut dire qu'aucune vérification sur la longueur de l'espace mémoire réellement réservé ne devra être faite. On se contentera de travailler avec le nombre d'éléments réellement contenus dans les tableaux manipulés, qui est passé en paramètre.

Q 1. Écrire une fonction permettant d'afficher le contenu d'un tableau dont le prototype est le suivant :

```
void afficher(int liste[], int taille);
```

Q 2. Écrire une fonction retournant la somme des éléments réellement stockés dans un tableau dont le prototype est le suivant :

```
int somme (int liste[], int taille);
```

Q 3. Écrire une fonction copiant dans un tableau le contenu d'un autre tableau, dont le prototype est le suivant :

```
void copie_dans(int dest[], int src[], int taille);
```

Q 4. Écrire une fonction qui ajoute la deuxième liste à la fin de la première, dont le prototype est le suivant :

```
void ajoute_apres(int dest[], int taille_dest, int src[], int taille_src);
```

Exercice 2 : Structures

Dans cette exercice nous allons manipuler des nombres rationnels et pour cela utiliser la structure suivante :

```
struct rat
{
  int den;
  int num;
};
```

On ne vous demande pas de réduire les nombres rationnels.

Q 1. Écrire une fonction retournant le produit de deux nombres rationnels, dont le prototype est le suivant :

```
struct rat rat_produit(struct rat n1, struct rat n2);
```

Q 2. Écrire une fonction retournant la somme de deux nombres rationnels, dont le prototype est le suivant :

```
struct rat rat_somme(struct rat n1, struct rat n2);
```

Q 3. Écrire une fonction retournant le plus petit élément d'une liste de rationnels, dont le prototype est le suivant :

```
struct rat rat_plus_petit(struct rat list[]);
```

On considère que le marqueur de fin du tableau est un élément dont le dénominateur vaut 0.