

# 4TIN303U - PROGRAMMATION C

## Dynamic allocation 1

► **Exercice 1.** Écrivez plusieurs versions d'une fonction qui calcule le minimum et le maximum d'un tableau d'entiers. Toutes ces fonctions auront en paramètre au moins un tableau d'entiers et le nombre d'éléments du tableau.

- v1 : retourne les deux valeurs dans un vecteur alloué dynamiquement.

```
int * minMax1(int *t, unsigned int n);
```

- v2 : retourne un pointeur sur une structure allouée dynamiquement. Vous devrez définir la structure resMinMax en question, contenant deux entiers.

```
resMinMax * minMax2(int *t, unsigned int n);
```

► **Exercice 2.** Écrivez également quelques lignes de code illustrant leur utilisation.

- sur un tableau alloué localement
- sur un tableau passé en argument de la ligne de commande et alloué dynamiquement

On voudra ainsi dans cette version que la commande

```
min-max-tab-entiers 1 2 9 0 99 -9
```

donne le résultat:

```
1st version : -9 99
```

```
2nd version : -9 99
```

► **Exercice 3.** Soit la structure suivante :

```
struct point {  
    float x;  
    float y;  
};
```

- Écrivez une fonction

```
point *createArrayOfPoint(unsigned int length);
```

qui alloue un tableau de taille *length* (passé en paramètre) pouvant contenir des éléments de type `struct point`.

- Écrivez une fonction

```
point computeGCenter(point* array, unsigned int length);
```

qui calcule le centre de gravité des points du tableau donné en paramètre.