

## Séances 1 et 2

**Module Liste** Le module liste gère une liste simplement chaînée. Les données enregistrées dans les nœuds sont des pointeurs génériques (`void *`). La gestion mémoire des objets pointés n'est prise en charge dans le module.

Le module devra apporté les fonctionnalités minimales suivantes :

- création d'une liste,
- lecture et écriture de la donnée d'un nœud,
- insertion d'une donnée en tête de liste,
- ajout d'une donnée en queue de liste,
- suppression d'une donnée en tête de la liste,
- suppression de la première instance d'une donnée dans la liste,
- insertion d'une donnée dans une liste ordonnée (une fonction fournie en paramètre est appelée pour comparer deux données),
- destruction d'une liste complète.

L'interface publique du module est donnée ci-dessous.

```
typedef struct _list_node s_node;
s_node * list_create(void);
    // creation d'une nouvelle liste vide
void * list_get_data(void s_node * node);
    // lire la donnee d'un noeud
void list_set_data(s_node * node, void * data);
    // ecrire la donnee d'un noeud
s_node * list_insert(s_node * head, void * data);
    // creation et insertion d'un noeud en tete de liste
    // retourne la tete de liste
s_node * list_append(s_node * head, void * data);
    // creation et ajout d'un noeud en queue de liste
    // retourne la tete de liste
s_node * list_orderedAppend(s_node ** head,
                           int (*compare)(s_node *, void *),
                           void *param);
    // ajout d'un noeud dans une liste ordonnee
    // selon le resultat de "compare" ;
    // si la donnee existe deja, elle n'est pas ajoutee
s_node * list_remove(s_node * head, void * data);
    // suppression de la premiere instance d'une
    // donnee dans la liste, retourne la tete de liste
s_node * list_headRemove(s_node * head);
    // suppression de la tete de liste
    // retourne la nouvelle tete de liste
void list_destroy(s_node * head);
    // destruction d'une liste
    // (La liberation des donnees n'est pas prise en charge)
```

Le programme de test utilisera les fonctions définies dans le fichier `testListHelpers.c`.