Licence Informatique 2016-2017 **4TIN302U - ASDE**
Université Bordeaux **Test de compréhension**

NOM : **PRENOM :**

Exercice 1. Étudier le code de la fonction suivante :

```
SList enigma(SList *p_L){
    if (!(*p_L) || !((*p_L)->next))
        return *p_L;

    SList n0 = NULL;
    SList n1 = *p_L;
    SList n2 = (*p_L)->next;

    while (n2){
        SList tmp = n2->next;
        n2->next = n1;
        n1->next = tmp;
        if (!n0)
            *p_L = n2;
        else
            n0->next = n2;
        n0 = n1;
        n1 = tmp;
        if (tmp)
            n2 = tmp->next;
        else
            n2 = NULL;
    }

    return *p_L;
}
```

— Étant donnée la liste `SList list` contenant les valeurs `1, 2, 3, 4`, déroulez pas à pas la fonction `enigma` sur `&list`. Pour cela, vous allez utiliser la représentation graphique des listes simplement chaînées.

— Que fait la fonction `enigma` dans le cas général ?

Exercice 2. Avec votre **casquette utilisateur**,

— Étant données les primitives du type abstrait `SList`, écrivez la fonction **récursive** `Slist asde_slist_del_odd_nodes(SList L)` qui supprime tous les éléments d'indice impair d'une liste passée en paramétre.

Par exemple, si la liste passée en paramétre contient la suite de valeurs : `1, 2, 3, 4, 5`, la liste retournée par la fonction contiendra : `1, 3, 5`.

— Quelle est la compléxité de cette fonction ?