

# Examen – Session principale

Matière : Atelier programmation II

Filière : MPI

Nom : .....

CIN : .....

Semestre: Second semestre

Date: 01 Juillet 2020

Prénom : .....

Numéro Inscription : .....

## Feuilles de réponse

### Exercice 1 :

Donner le rôle de chacune des fonctions :

- Question 1 : fonction\_1(2 pts)

**Fusionner deux listes triées en ordre croissant en une liste triée par ordre croissant**

- Question 2 : fonction \_2(2pts)

**Décomposer la liste initiale en deux listes : une liste qui contient les éléments de rangs pairs et une liste qui contient les éléments rangs impairs.**

- Question 3 : fonction \_3 (2pts)

**Tri par fusion d'une liste donnée**

- Question 4 : fonction \_4(1.5 pts)

**Supprimer le nœud contenant la valeur n.**

- Question 5 : fonction \_5(2 pts)

**La liste finale contient des éléments qui se répètent au maximum deux fois. Supprimer les redondances nécessaires pour que chaque élément apparait au plus deux fois.**



### Exercice 1 :

Soit la fonction suivante :

```
#include<stdio.h>
#include<stdlib.h>

struct element{
int donnee;
element * suivant;
};

int fct (element * A, element * B){
    element * tmp, * ptr;

    tmp = B;
    while(tmp!=NULL){
        ptr=A;
        while(ptr != NULL){
            if(ptr->donnee == tmp->donnee){
                ptr = ptr->suivant;
                tmp = tmp->suivant;
            }
            else
                break;
        }
        if(ptr == NULL)
            return 1;
        else
            if(tmp->donnee != A->donnee)
                tmp = tmp->suivant;
    }
    return 0;
}
```

Expliquez le rôle de la fonction *fct* en donnant des exemples.

**La fonction *fct* permet de vérifier si tout les éléments de la liste B appartiennent à la liste A. Cette fonction a comme arguments les deux listes et devra retourner 1 si tout les éléments de B appartiennent à A et 0 sinon. Les deux listes ne peuvent pas contenir la même valeur plusieurs fois.**

### Exercice 2 :

La fonction *dupliquer* permet de dupliquer les nombres pairs dans une liste d'entiers L.

Exemple : si la liste L contient au début les éléments [3,11,8,5,16,7,4], L contiendra [3,11,8,8,5,16,16,7,4,4] après l'exécution de la fonction *dupliquer*.

Compléter le code de la fonction dupliquer :

```
element * dupliquer(element * debut){
    element * tmp, * nouv;
```



```

tmp = debut;
while(tmp!=NULL){
    if(tmp->donnee % 2 == 0){
        nouv = (element *) malloc(sizeof(element));
        nouv->donnee = tmp->donnee;
        nouv->suivant = tmp->suivant;
        tmp->suivant = nouv;
        tmp = nouv->suivant;
    }
    else
        tmp = tmp->suivant;
}
return debut;
}

```

Exercice 2 :

Soit les deux programmes suivants :

```

#include<stdio.h>
#include<stdlib.h>

struct element{
    int donnee;
    element * suivant;
};

element * operation(element * debut){
    element * ptmp, * tmp;
    ptmp = debut;
    tmp = debut->suivant;
    while(tmp != NULL){
        if(ptmp->donnee == tmp->donnee){
            ptmp->suivant = tmp->suivant;
            free(tmp);
            tmp = ptmp->suivant;
        }
        else{
            ptmp = tmp;
            tmp = tmp->suivant;
        }
    }
    return debut;
}

```

```

#include<stdio.h>
#include<stdlib.h>

struct pile{
    int donnee;
    pile * precedent;
};

pile * mystere(pile *A, pile *B){
    pile * C;
    int x;
    C = NULL;
    while(est_vide(A)==0 && est_vide(B)== 0){
        if(sommet(A) <= sommet(B))
            x = depiler(&A);
        else
            x = depiler(&B);
        C = empiler(C, x);
    }
    while(est_vide(A) == 0){
        x = depiler(&A);
        C = empiler(C, x);
    }
    while(est_vide(B) == 0){
        x = depiler(&B);
        C = empiler(C, x);
    }
    return C;
}

```

1. Soit une liste chaînée L contenant les valeurs suivantes <7, 3, 3, 9, 6, 6, 6, 2>. Donnez le rôle et le résultat d'exécution de la fonction *operation* pour la liste L.

➔ **Supprimer les redondants <7, 3, 9, 6, 2>**

2. Donnez la pile C obtenue en précisant le sommet après exécution de la fonction *mystere* pour les piles A = [8, 7, 3, 2] 2 est le sommet de la pile A et B = [16, 12, 11, 6, 4] 4 est le sommet de la pile B. (On suppose qu'on fait appel aux fonctions de manipulation des piles).

**C = [4, 6, 11, 12, 16], 16 est le sommet de la pile.**