COMPTE RENDU TP 4 RN40

BARTHELME ALEXANDRE

Mon code complet est disponible sur mon GitHub à l'adresse :
https://github.com/Iron68A/RN40/tree/main/TP/TP4

Ou par capture d'écran

Les fonctions rajoutées sont :

1) Le compteur :

```c
//fonction qui compte le nombre d'éléments de la liste
    //Itératif
int compter(List l){
    int n =0;
    while (l != NULL){
        n++;
        l = l -> next;
    }
    return n;
}
    //Récursif
int compter_rec(List l){
    if (l == NULL){
        return 0;
    }
    else{
        return 1 + compter_rec(l -> next);
    }
}
```

2) Le nombre d'occurrence d'un produit :

```c
//Fonction nombre d'occurences d'un élément dans la liste ( avec fct rechercher )
    //Itératif
int nbr_occurences(List l, int c){
    int n = 0;
    if(l==NULL){return 0;}
    while (l != NULL){
        if (recherche(l, c) ==l){
            n++;
        }
        l = l -> next;
    }
    return n;
}
    //Récursif
int nbr_occurences_rec(List l, int c){
    if (l == NULL){return 0;}
    else{
        if (recherche(l, c) == l){
            return 1 + nbr_occurences_rec(l -> next, c);
        }
        else{
            return nbr_occurences_rec(l -> next, c);
        }
    }
}
```

3) Effacer la liste :

```c
//Effacer la liste chainée
    //Itératif
void effacer(List l){
    while (l != NULL){
        List tmp = l;
        l = l -> next;
        free(tmp);
    }
}
    //Récursif
void effacer_rec(List l){
    if (l != NULL){
        effacer_rec(l -> next);
        free(l);
    }
}
```

4) Effacer un produit d'un certain prix :

```c
//Fonction qui supprime tous les éléments ayant un prixP donné
    //Itératif
List supprimer_prix(List l, float p){
    while (l != NULL){
        if (l -> prixP == p){
            List tmp = l;
            free(tmp);
        }
        l = l -> next;
    }
    return l;
}
    //Récursif
List supprimer_prix_rec(List l, float p){
    if (l != NULL){
        if (l -> prixP == p){
            List tmp = l;
            free(tmp);
        }
        supprimer_prix_rec(l -> next, p);
    }
    return l;
}
```

5) Renvoyer le i ème element :

```c
//renvoyer le i eme element de la liste
    //Itératif
List ieme(List l, int i){
    if(i<0 || i>compter(l)){printf("La liste ne contient que %n éléments",compter(l));}
    int n = 0;
    while (l != NULL){
        if (n == i){
            return l;
        }
        n++;
        l = l -> next;
    }
    return NULL;
}
    //Récursif
List ieme_rec(List l, int i){
    if (i<0 || i>compter(l)){printf("La liste ne contient que %n éléments",compter(l));}
    if (l == NULL){return NULL;}
    else{
        if (i == 0){
            return l;
        }
        else{
            return ieme_rec(l -> next, i-1);
        }
    }
}
```