

Structures de données et algorithmes

TP Listes en chaînage simple et circulaire

Introduction

L'objectif de ce TP est de maîtriser parfaitement les listes simplement chaînées, et circulaires en limitant la dépendance à l'implantation aux opérations du noyau.

Simple chaînage

On utilise un simple chaînage en définissant la structure de données suivante :

```
typedef struct strliste {  
    S v;          // valeur  
    struct strliste *s; // suivant  
} Strliste, *Liste ;
```

1. Implantez les opérations standard définissant le type Liste : liste vide, accès, ajout et suppression en tête, longueur, test de vacuité et d'appartenance.
2. Implantez les opérations sur la première occurrence d'un élément sur le type Liste : recherche de la liste dont la tête est la première occurrence, changement et suppression de la première occurrence. Une opération avançant jusqu'à ce que la tête soit l'élément précédent la première occurrence d'un élément dans la liste qui pourra être utilisée pour faciliter la programmation de la suppression.
3. Implantez les opérations en dernière position sur le type Liste : accès, insertion, suppression. Une opération avançant jusqu'au dernier élément dans la liste pourra être utilisée pour faciliter la programmation de ces opérations ainsi qu'une autre avançant jusqu'à l'avant-dernier élément.
4. Implantez les opérations en i-ème position sur le type Liste : accès, insertion, suppression. Une opération avançant jusqu'au i-ème élément dans la liste pourra être utilisée pour faciliter la programmation de ces opérations. Ces opérations pourront être écrites avec puis sans préconditions.
5. Faites un makefile et écrivez un programme de test avec trois listes différentes en utilisant toutes les opérations : vide, adjonction en tête, insertion, suppression . . .

Simple chaînage circulaire

1. Expliquez l'intérêt de rendre circulaire une liste implantée avec un simple chaînage. Comment faut-il modifier la structure de données pour rendre la liste circulaire ?
2. Implantez les opérations standard définissant le type Liste en considérant un chaînage simple et circulaire : liste vide, accès, ajout et suppression en tête, longueur, test de vacuité et d'appartenance.
3. Implantez les opérations en dernière position sur le type Liste : accès, insertion, suppression ou changement. Une opération de permutation avançant au cran suivant dans la liste pourra être utilisée pour faciliter la programmation de ces opérations.
4. Faites un makefile et écrivez un programme de test avec trois listes différentes en utilisant toutes les opérations : vide, adjonction en tête, insertion, suppression . . .