TP 2: Algorithmique Procédurale
Liste chaînée

Nga Nguyen - Stefan Bornhofen - Romain Dujol - Peio Loubière

Mardi 14 Février 2012

L'objectif de ce TP est d'implémenter les algorithmes sur les listes chaînée vus en cours et en TD. Votre travail devra être déposé sur AREL **au plus tard le Vendredi 6 Avril 2012 à 23h59m59**. Vous répondrez dans un fichier que vous nommerez `NOM-listeChainee.pas`.

1 Déclaration du type `ptr_noeud`

On définit en PASCAL le type de données « liste d'entiers » via la notion de noeud.

```
TYPE
  ptr_noeud = ^noeud ;
  noeud = RECORD
            valeur  : integer   ;
            suivant : ptr_noeud ;
          END ;
```

2 Travail demandé

2.1 Opérations de base

Implémenter les opérations suivantes :

Question 1 : `function ajoutTete(teteliste : ptr_noeud ; n : integer) : ptr_noeud` pour ajouter un nœud contenant la valeur n en début de liste ;

Question 2 : `function ajoutFin(teteliste : ptr_noeud ; n : integer) : ptr_noeud` pour ajouter un nœud contenant la valeur n en fin de liste ;

Question 3 : `function ajoutApres(p : ptr_noeud ; n : integer) : ptr_noeud` pour ajouter un nœud contenant la valeur n apres le nœud p ;

Question 4 : `function ajoutKieme(teteliste : ptr_noeud ; n, k : integer) : ptr_noeud` pour ajouter un nœud contenant la valeur n en $k^{\text{ème}}$ position

Question 5 : `function supprTete(teteliste : ptr_noeud) : ptr_noeud` pour supprimer le nœud en début de liste ;

Question 6 : `function supprFin(teteliste : ptr_noeud) : ptr_noeud` pour supprimer le nœud en fin de liste ;

Question 7 : `function supprApres(p : ptr_noeud) : ptr_noeud` pour ajouter un nœud contenant la valeur n apres le nœud p ;

Question 8 : `function supprKieme(teteliste : ptr_noeud ; k : integer) : ptr_noeud` pour supprimer le nœud situé en $k^{\text{ème}}$ position.

On prendra garde à bien gérer tous les cas : ajout à une position $k >$ longueur(`ptr_noeud`), suppression dans une liste vide, etc. . .

2.2 Opérations avancées

Implémenter les opérations suivantes :

Question 9 : `function appartient(teteliste : ptr_noeud ; n : integer) : ptr_noeud` pour retourner le nœud contenant la valeur n dans *teteliste* ou nil si il n'y a pas un tel nœud ;

Question 10 : `function suppPremier(teteliste : ptr_noeud ; n : integer) : ptr_noeud` pour supprimer le premier nœud contenant la valeur n ;

Question 11 : `function suppTous(teteliste : ptr_noeud ; n : integer) : ptr_noeud` pour supprimer tous les nœuds contenant la valeur n ;

Question 12 : `function renverser(teteliste : ptr_noeud) : ptr_noeud` pour renverser la liste.

2.3 Plateforme de test

Écrire un programme principal qui permettra de tester vos fonctions :
- à partir d'une liste vide au démarrage du programme, l'utilisateur devra être capable d'effectuer via un menu textuel les opérations précédent (à l'exception de `ajouterApres` et `supprApres`) ;
- après chaque action, votre programme affichera l'état actuel de la liste.