|  |  |  |
| --- | --- | --- |
|  | TP 2  Les Listes Chaînées |  |

|  |  |  |
| --- | --- | --- |
| Objectifs | Durée | Outil |
| - Se familiariser avec la manipulation des listes simplement chaînée. | 3h00 | DevCpp |

Nous désirons dans ce TP réaliser une liste chaînée d’étudiants. La structure que vous utiliserez pour manipuler les étudiants est la suivante :

**struct etudiant{**

**int numInsc ;**

**char nom[20];**

**char prenom[20];**

**int age;**

**struct etudiant \* next;**

**};**

1. Définir le type **listetudiant** comme pointeur sur un étudiant.
2. Réalisez une fonction permettant de créer un étudiant en renseignant les différents champs de la structure (nom, prénom et âge) à partir du clavier.

**Prototype : listetudiant creerEtudiant() ou struct etudiant\* creerEtudiant()**

1. Réaliser une fonction permettant d’ajouter un étudiant en **tête** de la liste. Vous ferez appel à la fonction précédente pour la création du nouvel étudiant à ajouter.

**Prototype : listetudiant ajouterEnTete(listetudiant l)**

1. Réaliser une fonction permettant d’ajouter un étudiant en **fin** de la liste. Vous ferez appel à la fonction **creerEtudiant()** pour la création du nouvel étudiant à ajouter.

**Prototype : listetudiant ajouterEnFin(listetudiant l)**

1. Réaliser une fonction permettant d’afficher l’ensemble des étudiants de la liste.

**Prototype :** **void afficherListe(listetudiant l)**

1. Réaliser une fonction permettant de supprimer un étudiant en tête de liste.

**Prototype : listetudiant supprimerEtudiantEnTete(listetudiant l)**

1. Réaliser une fonction permettant de supprimer un étudiant en fin de liste.

**Prototype : listetudiant supprimerEtudiantEnFin(listetudiant l)**

1. Réaliser une fonction permettant de rechercher un étudiant de la liste dont le numéro d’inscription est passé en paramètre et retourne un pointeur sur cet étudiant.

**Prototype : listetudiant rechercherElement(listetudiant l, int num) ;**

1. Réaliser une fonction qui retourne un pointeur sur l’étudiant de la liste dont le numéro d’inscription est le plus petit.

**Prototype :** **listetudiant minimumListe(listetudiant l) ;**

1. Réaliser une fonction permettant d’échanger la place de deux étudiants dans une liste chainée. Si l’un des deux étudiants n’existe pas, ne rien faire. Vous devez effectuer un échange des structures, et pas uniquement des numéros d’inscription ;