DS Algorithmique - 3IF - 08 janvier 2015

Contraintes

Les solutions au DS sont validées via la plateforme domjudge. Chaque élève a son propre login et son propre mot de passe. Vous pouvez envoyer plusieurs fois une solution pour un même problème sans aucune pénalité.

Vous avez le droit aux supports des cours, TD, TP et aussi à des ressources Internet, mais vous n'avez pas le droit de communiquer avec d'autres personnes. La DSI sauvegarde tout le trafic TCP/IP pendant le DS. Vous pouvez utiliser seulement le protocole http pour accéder aux différents sites Web. Une connexion à un outil de messagerie, réseau social etc invalide automatiquement votre participation au DS avec les conséquences prévues par le règlement des études.

Vous pouvez utiliser le langage C ou le C++, mais sans utiliser la STL.

Problème numéro 3 : File implémentée par une liste chaînée
---

* Introduction

Pour rappel, une file est un type de données abstrait qui représente une collection gérée en FIFO et offrant l'interface suivante :
- EnFile permet d'ajouter un élément
- DeFile permet de supprimer un élément et de le renvoyer
- EstVide renvoie vrai si la file est vide

* Code de base

Le code de base qui est proposé ci-dessous et que vous devez utiliser permet de gérer toutes les entrées-sorties et vous donne déjà les structures de données, vous n'avez que trois fonctions à implémenter, dont le prototype est donné en commentaire.

```
/* problème numero 3 - file implementée par une liste chainée */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct Element {
   struct Element * suivant;
   int valeur;
} Element;

typedef struct {
  Element * debut; /* le début de la liste chaînée */
  Element * fin;   /* le dernier élément de la liste */
} File;

void EnFile(File * file, int valeur);
int DeFile(File * file);
int EstVide(File file);
void Init(File * file);
void Destroy(File file);

void error(void);
int main(void)
```

```
{
   int val;
   char lecture[100];
   File file;

   Init(&file);

   if (fscanf(stdin,"%99s",lecture)!=1)
      error();
   while (strcmp(lecture,"bye")!=0)
   {
      if (strcmp(lecture,"enfile")==0)
      {
         if (fscanf(stdin,"%99s",lecture)!=1)
            error();
         val = atoi(lecture);
         EnFile(&file,val);
      }
      else if (strcmp(lecture,"defile")==0)
      {
         val = DeFile(&file);
         printf("%d\r\n",val);
      }
      else if (strcmp(lecture,"estvide")==0)
      {
         printf("%s\r\n",EstVide(file)?"oui":"non");
      }

      if (fscanf(stdin,"%99s",lecture)!=1)
         error();
   }
   Destroy(file);
   return 0;
}

void EnFile(File * file, int valeur)
/* cette procédure enfile valeur dans la file */
/*  - file est un pointeur sur la structure de données File et
 *    représente une file bien formée
 *  - valeur est la valeur à insérer */
{
   /* insérer le code ici */
}

int DeFile(File * file)
/* cette fonction défile la valeur et la renvoie */
/*  - file est un pointeur sur la structure de données File et
 *    représente une file bien formée
 *  - si la file est vide la valeur 0 est renvoyée */
{
   /* insérer le code ici */
}

int EstVide(File file)
/* cette fonction indique si la file est vide */
/*  - file est de type File et représente une file bien formée
 *  - la valeur renvoyée vaut 1 si la file est vide, 0 sinon */
{
   /* insérer le code ici */
}

void Init(File * file)
{
   file->debut = NULL;
   file->fin = NULL;
}

void Destroy(File file)
{
   Element * courant = file.debut;
   Element * suivant;
   while (courant != NULL)
   {
      suivant = courant->suivant;
      free(courant);
      courant = suivant;
   }
```

```
}

void error(void)
{
   printf("input error\r\n");
   exit(0);
}
```