Devoir surveillé - Éléments de correction

10 novembre 2020 - Durée 1h15

Document autorisé : **Mémento C** vierge de toute annotation manuscrite

On considère le paquetage **Traitement** dont l'implémentation est la suivante :

```c
#include <stdio.h>
#include <stdlib.h>
#include "traitement.h"

int traitement(tableau_entiers * t){
    int i, j, k;

    for(i = 0; i < t->taille; i++){
        for(j = i+1 ; j < t->taille; j++){
            if(t->tab[i] == t->tab[j]){
                for(k = j; k < t->taille; k++){
                    t->tab[k] = t->tab[k +1];
                }
                t->taille--;
                j--;
            }
        }
    }

    return 0;
}
```

*Le fichier **traitement.h** est donné en annexe.*

Exercice 1. (2 pts)

Que fait la fonction `traitement()` ? La fonction traitement

Exercice 2. (2 pts)

Quel est le format des entrées de cette fonction `traitement()` ?

Exercice 3. (4 pts)

Écrire un programme de test du paquetage **Traitement** :

1. Il utilisera les paquetages fournis en annexe (**type_tableau** et **es_tableau**).
2. Il affichera le résultat de l'appel de la fonction `traitement()`.
3. Il lira un tableau depuis un fichier dont le nom est donné en argument de la ligne de commande.

Exercice 4. (4 pts)

Décrire un jeu de tests fonctionnels permettant de tester la fonction `traitement()`.

Exercice 5. (5 pts)

Q1. (2 pts) Schématiser la structure globale et les relations entre les paquetages.

Q2. (3 pts) Écrire un Makefile permettant de compiler le programme.

Exercice 6. (3 pts)

Nous souhaitons ajouter un module `oracle`, permettant de tester la fonction `traitement()`. Décrire les fichiers ajoutés (sans détailler l'implémentation des fonctions) et mettre à jour la structure globale réalisée pour la question 1 de l'exercice 5.

A. Annexes : paquetages utilisés

Paquetage Traitement

Fichier traitement.h

```c
#include <stdio.h>
#include <stdlib.h>
#include "type_tableau.h"

int traitement(tableau_entiers t);
```

Paquetage type_tableau

Fichier type_tableau.h

```c
#ifndef _TYPE_TABLEAU_H_
#define _TYPE_TABLEAU_H_

#define TAILLE_MAX 10000

/* Definition du type vecteur_entiers :
   tableau d'entiers de taille TAILLE_MAX */
typedef int vecteur_entiers[TAILLE_MAX];

/* Structure contenant un tableau (de taille TAILLE_MAX) et un entier
   taille : le nombre d'entiers du tableau */
typedef struct {
  int taille;
  vecteur_entiers tab;
} tableau_entiers;

#endif /* _TYPE_TABLEAU_H_ */
```

Paquetage es_tableau

Fichier es_tableau.h

```c
#ifndef _ES_TABLEAU_H_
#define _ES_TABLEAU_H_

#include <stdio.h>
#include "type_tableau.h"

void lire_tableau(FILE * fichier, tableau_entiers * t);

void ecrire_tableau(FILE * fichier, tableau_entiers t);

#endif /* _ES_TABLEAU_H_ */
```

Fichier es_tableau.c

```c
#include <stdio.h>
#include "es_tableau.h"

void lire_tableau(FILE * fichier, tableau_entiers * t) {
  int i;

  /* Lecture de la taille du tableau */
  fscanf(fichier, "%d", &(t->taille));

  /* Lecture des valeurs du tableau */
  for (i = 0; i < t->taille; i++) {
    fscanf(fichier, "%d", &(t->tab[i]));
  }
}

void ecrire_tableau(FILE * fichier, tableau_entiers t) {
  int i;

  /* Ecrire la taille du tableau dans le fichier */
  fprintf(fichier, "%d\n", t.taille);

  /* Ecrire les valeurs du tableau */
  for (i = 0; i < t.taille; i++) {
    fprintf(fichier, "%d\n", t.tab[i]);
  }
}
```