

Quizz 4 (séance 5) en C - Gestion des dossiers des élèves

Xuan Son NGUYEN

Octobre 2021

Ce quizz doit vous permettre une auto-évaluation. Vous ne serez pas noté mais le sujet couvre les compétences qui devraient maintenant être les vôtres. Vous avez 60 minutes pour développer ce projet. Les 15 dernières minutes seront consacrées à la correction par l'enseignant.

Un barème totalement indicatif vous indique quel pourrait être votre note au contrôle dans l'état actuel de vos compétences.

1 Introduction et mise en place - 2 points

On veut créer un programme pour la gestion des dossiers des élèves d'une classe. La classe contient 20 élèves.



- Créez un répertoire vide dans lequel vous allez développer votre projet,
- Créez 3 fichiers : main.c, fonctions.c et fonctions.h,
- Complétez le fichier main.c avec la fonction `int main (int argc, char ** argv)` pour afficher le message "***** Gestion des dossiers des eleves*****"
- Compilez puis testez.

2 Lire les données des élèves dans un tableau - 3 points

Définissez dans fonctions.h une structure pour les élèves et quatre fonctions permettant de lire les données des élèves dans un tableau, d'afficher ces données, et de trier les élèves selon leur note et leur nom.

```

1  #include <stdio.h>
2  #include <string.h>
3
4  #define MAX_NOM_LENGTH 50
5  #define MAX_LIST_SIZE 20
6
7  typedef struct etudiant
8  {
9      int id;
10     char nom[MAX_NOM_LENGTH];
11     int note;
12 } StructEtu;
13
14 typedef StructEtu ListStructEtu[MAX_LIST_SIZE];
15
16 void entrerDonnees(ListStructEtu list, int s);
17 void afficherDonnees(ListStructEtu list, int s);
18 void trierDonneesNote(ListStructEtu list, int s);
19 void trierDonneesNoteNom(ListStructEtu list, int s);

```

Écrivez la fonction `entrerDonnees` pour lire les données des élèves. Testez-la dans `main.c` comme suit :

```

1  #include "fonctions.h"
2
3  int main()
4  {
5      ListStructEtu data;
6      int n;
7
8      printf("Nombre d'etudiants (max=20)? : ");
9      scanf("%d", &n);
10
11     entrerDonnees(data, n);
12
13     return 0;
14 }

```

3 Afficher les dossiers des élèves - 3 points

Écrivez la fonction `afficherDonnees` pour afficher les données des élèves. Testez-la dans `main.c` comme suit :

```
1  #include "fonctions.h"
2
3  int main()
4  {
5      ListStructEtu data;
6      int n;
7
8      printf("Nombre d'etudiants(max=20)? : ");
9      scanf("%d", &n);
10
11     entrerDonnees(data, n);
12     printf("\nDonnees entrees");
13     afficherDonnees(data, n);
14
15     return 0;
16 }
```

Un exemple de l'exécution du programme ci-dessus :

```
1  Nombre d'etudiants(max=20)? : 2
2
3  Donnees pour l'etudiant #1
4  Numero d'etudiant (Id) : 1
5  Nom (max=50 caracteres) : a
6  Note : 1
7
8  Donnees pour l'etudiant #2
9  Numero d'etudiant (Id) : 2
10 Nom (max=50 caracteres) : b
11 Note : 2
12
13 Donnees entrees
14
15 Id      Nom      Note
16 1       a       1
17 2       b       2
```

4 Trier les dossiers des élèves par leur note - 5 points

On veut trier les dossiers des élèves sur la base des notes par ordre décroissant. Écrivez la fonction `trierDonneesNote` en utilisant un algorithme de tri vu dans le TD4. Testez-la dans `main.c` comme suit :

```
1  #include "fonctions.h"
2
3  int main()
4  {
5      ListStructEtu data;
6      int n;
7
8      printf("Nombre d'etudiants(max=20)? : ");
9      scanf("%d", &n);
10
11     entrerDonnees(data, n);
12     printf("\nDonnees avant le tri");
13     afficherDonnees(data, n);
14     trierDonneesNote(data, n);
15     printf("\nDonnees apres le tri");
16     afficherDonnees(data, n);
17
18     return 0;
19 }
```

Un exemple de l'exécution du programme ci-dessus :

```

1  Nombre d'etudiants(max=20)? : 2
2
3  Donnees pour l'etudiant #1
4  Numero d'etudiant (Id) : 1
5  Nom (max=50 caracteres) : a
6  Note : 2
7
8  Donnees pour l'etudiant #2
9  Numero d'etudiant (Id) : 2
10 Nom (max=50 caracteres) : b
11 Note : 3
12
13 Donnees avant le tri
14
15 Id      Nom      Note
16 1       a        2
17 2       b        3
18
19 Donnees apres le tri
20
21 Id      Nom      Note
22 2       b        3
23 1       a        2

```

5 Trier les dossiers des élèves par leur note et nom - 5 points

On veut trier les dossiers des élèves sur la base des notes par ordre décroissant. Les élèves ayant eu la même note doivent être triés dans l'ordre alphabétique. Écrivez la fonction `trierDonneesNoteNom` en utilisant un algorithme de tri vu dans le TD4. Testez-la dans `main.c` comme suit :

```
1  #include "fonctions.h"
2
3  int main()
4  {
5      ListStructEtu data;
6      int n;
7
8      printf("Nombre d'etudiants(max=20)? : ");
9      scanf("%d", &n);
10
11     entrerDonnees(data, n);
12     printf("\nDonnees avant le tri");
13     afficherDonnees(data, n);
14     trierDonneesNoteNom(data, n);
15     printf("\nDonnees apres le tri");
16     afficherDonnees(data, n);
17
18     return 0;
19 }
```

Un exemple de l'exécution du programme ci-dessus :

```

1  Nombre d'etudiants(max=20)? : 3
2
3  Donnees pour l'etudiant #1
4  Numero d'etudiant (Id) : 1
5  Nom (max=50 caracteres) : a
6  Note : 1
7
8  Donnees pour l'etudiant #2
9  Numero d'etudiant (Id) : 2
10 Nom (max=50 caracteres) : c
11 Note : 2
12
13 Donnees pour l'etudiant #3
14 Numero d'etudiant (Id) : 3
15 Nom (max=50 caracteres) : b
16 Note : 2
17
18 Donnees avant le tri
19
20 Id      Nom      Note
21 1       a       1
22 2       c       2
23 3       b       2
24
25 Donnees apres le tri
26
27 Id      Nom      Note
28 2       b       2
29 3       c       2
30 1       a       1

```

6 Comparaison des algorithmes de tri - 2 points

Réalisez les questions 4 et 5 en utilisant différents algorithmes de tri vu dans le TD4 et comparez leur temps de calcul.