|  |  |  |
| --- | --- | --- |
| **BTS1 Cours** | **Structures et fichiers binaires** | **03/2021** |

# Contexte

Une administratrice de votre lycée souhaite créer un fichier informatique des élèves pour y stocker les informations suivantes :

* Nom (en majuscule, 30 caractères max)
* Prénom (première lettre de chaque « mot » en majuscule, 30 caractères max)
* Age (maximum 27 ans)
* Classe (-2 pour les secondes, -1 pour les premières, 0 pour les terminales, 1 pour BTS1 et 2 pour BTS2)

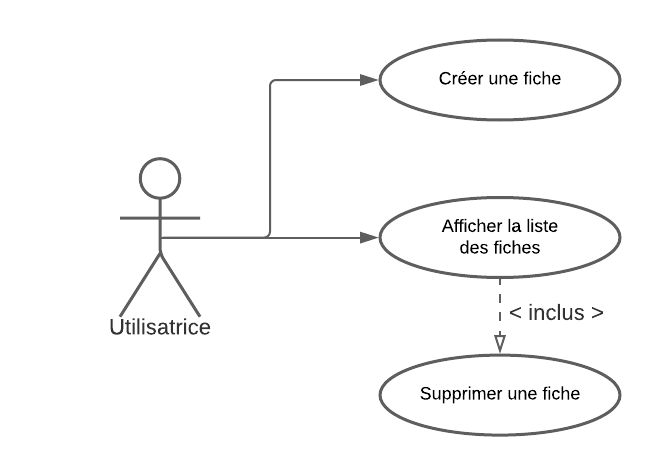
Elle estime qu’elle aura entre 50 et 100 étudiants à saisir.

Elle sera seule à utiliser le programme que vous allez créer.

Elle pourra fermer et rouvrir le programme et retrouver les informations saisies.

Elle souhaite que la sauvegarde soit automatique.

Elle souhaite pouvoir effectuer les actions suivantes :



*Diagramme des cas d’usage (UML)*

L’affichage à l’écran devra respecter le format suivant :

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Prénom | Age | Classe |
| HADDOCK | Archibald | 19 | BTS1 |
| TOURNESOL | Tryphon | 15 | Seconde |
| <------- 31 caractères ------> | <-------------- 31 -------------> | <- 4 -> | <--- 12 ---> |

# Conception fonctionnelle

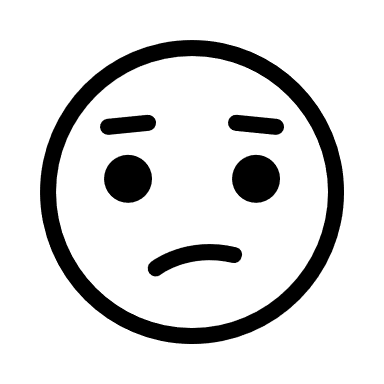
Dans un fichier Excel, préparer le cahier de recette/tests de votre application.

Quelques conseils :

* Le cahier de tests permet de vérifier que le programme développé répond aux attentes du cahier des charges et aux spécifications.
* Il doit être précis et exhaustif :
  + Précis dans les actions/directives (enchaînement, écrans, libellés), dans les valeurs à saisir et le résultat attendu (le testeur ne doit pas avoir à se poser de questions)
  + Exhaustif car il doit vérifier l’ensemble des choix de conception et des contraintes sur les données (taille, format, validation, cohérence) exprimées lors de la conception.
* Utiliser un fichier tableur (type Excel) permet de regrouper les cas de tests dans un seul fichier mais aussi de les classer et d’ajouter des contrôles comme afficher le nombre de cas validés ou non.
* Les cas de tests doivent être numérotés. Une numérotation de 10 en 10 permet au testeur étourdi d’insérer des cas oubliés sans avoir à refaire la numérotation 😊.
* Voici un exemple de cahier de tests pour une page de connexion sur un site internet :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N°** | **Prérequis** | **Directives** | **Résultat attendu** | **OK/KO** | **Commentaires** |
| 10 | Première visite ou déconnexion préalable. | Accéder à la page de connexion. | Un formulaire de connexion s’affiche avec les éléments suivants :   * Un champ « Identifiant » * Un champ « Mot de passe » * Un bouton « Connexion » * Un lien « Mot de passe oublié » |  |  |
| 20 | 10 | Saisir « u0001 » dans le champ « Identifiant ».  Saisir « passwd01 » dans le champ « Mot de passe ». | Le champ « Identifiant » est en clair.  Les caractères du champ « Mot de passe » sont masqués et remplacé par des points noirs. |  |  |
| 30 | 20 | Cliquer sur le bouton « Connexion ». | La page d’accueil du site s’affiche. |  |  |
| 35 | 30 | Accéder à la page de connexion. | La page d’accueil du site s’affiche directement. |  |  |
| 40 | 10 | Saisir « u0001 » dans le champ « Identifiant ».  Saisir « passwd02 » dans le champ « Mot de passe ».  Cliquer sur le bouton « Connexion ». | Le message d’erreur « Couple Identifiant/Mot de passe invalide » apparaît en rouge au-dessus du formulaire. |  |  |
| 50 | 40 | Saisir « u0001 » dans le champ « Identifiant ».  Vider le champ « Mot de passe ».  Cliquer sur le bouton « Connexion ». | Le message d’erreur « Couple Identifiant/Mot de passe invalide » apparaît en rouge au-dessus du formulaire. |  |  |
| 60 | 50 | Vider le champ « Identifiant ».  Saisir « passwd02 » dans le champ « Mot de passe ».  Cliquer sur le bouton « Connexion ». | Le message d’erreur « Couple Identifiant/Mot de passe invalide » apparaît en rouge au-dessus du formulaire. |  |  |
| 70 | 60 | Saisir « u0001 » dans le champ « Identifiant ».  Saisir « passwd01 » dans le champ « Mot de passe ».  Cliquer sur le bouton « Connexion ». | La page d’accueil du site s’affiche. |  |  |
| 80 | Première visite ou déconnexion préalable. | Cliquer sur le lien « Mot de passe oublié » | La page « Mot de passe oublié » s’affiche. |  |  |

# Conception technique

Vous allez devoir stocker jusqu’à 100 fiches (nom + prénom + âge + classe), avec vos connaissances actuelles, vous pourriez vous en sortir en déclarant 400 variables au noms différents … ou alors avec un tableau, mais…

Les tableaux permettent-ils de regrouper sous un seul nom des données de type différents ?

⃝ Oui ✓Non

Si vous avez répondu « Oui » à la question précédente, aller faire un petit tour dans votre cours sur les tableaux…

Il nous faudrait donc 4 tableaux différents pour stocker les 4 informations attendues… le risque de manipuler de multiples tableaux ou des tableaux multiples est de se mélanger dans les indices sans compter que le code peut rapidement devenir illisible.

Le C nous offre un « type » de variable parfaitement adapté à cet usage : les structures.

# Recherche sur les structures

Répondre aux questions suivantes en réalisant des recherches sur Internet. Pour les éléments de code, tester vos propositions avec votre IDE. Vos recherches feront office de support de cours, soignez votre rédaction.

|  |  |
| --- | --- |
| 1 | Quelle est la différence entre un tableau et une structure ? |
|  | Un tableau permet de regrouper sous un même nom des variables de même type et on accède à ses éléments via un indice.  Une structure permet de regrouper sous un même nom des variables de types différents et on accède à ses membres via leur nom. |
| 2 | Comment appelle-t-on les éléments d’une structure ? |
|  | Les éléments d’une structure sont appelés membres. On trouvera aussi attributs ou champs. |
| 3 | Comment déclare-t-on une structure ? |
|  | Une structure se déclare de la manière suivante :  struct <identifiant\_structure> {      <type\_membre\_1> <identifiant\_membre\_1>;      <type\_membre\_2> <identifiant\_membre\_2>;      ...      <type\_membre\_n> <identifiant\_membre\_n>;  };  Ne pas oublier le « ; » à la fin. |
| 4 | Proposer un exemple de déclaration de structure « point » pour stocker les coordonnées entières x et y d’un point sur un graphique. |
|  | struct point {      int x;      int y;  }; |
| 5 | Proposer la déclaration de la structure « student » qui répondra au besoin de notre utilisatrice. |
|  | struct student {      char lastname[31];      char firstname[31];      int age;      int grade;  }; |
| 6 | A quel(s) endroit(s) peut-on placer la déclaration d’une structure ? |
|  | Il faut que la structure soit déclarée avant d’être utilisée.  Elle peut être déclarée de manière globale et être utilisée partout dans le programme ou de manière locale et, dans ce cas, elle n’est utilisable que dans la fonction où elle a été déclarée. |
| 7 | Comment déclare-t-on une variable d’un type structuré ? |
|  | struct <identifiant\_structure> <identifiant\_variable>; |
| 8 | Proposer un exemple de déclaration d’une variable avec la structure « point ». |
|  | struct point p1; |
| 9 | Comment accède-t-on aux éléments de la structure ? |
|  | On utilise le point « . » pour accéder aux membres d’une variable de type structurée :  <identifiant\_variable>.<identifiant\_membre> |
| 10 | Proposer un exemple qui crée un point p et qui affecte les valeurs 3 et 7 aux membres x et y de ce point. |
|  | p1.x = 3;  p1.y = 7; |
| 11 | Proposer un exemple qui déplace le point de la question 10 de 3 sur l’axe des abscisses et de -2 sur l’axe des ordonnées. |
|  | p1.x += 3;  p1.y = p1.y - 2; |

# Première implémentation

Fort de vos recherches, proposer une première version simple du programme qui permet de saisir une fiche, puis de l’afficher.

# Recherche sur les structures

Répondre aux questions suivantes en réalisant des recherches sur Internet. Pour les éléments de code, tester vos propositions avec votre IDE. Vos recherches feront office de support de cours, soignez votre rédaction.

|  |  |
| --- | --- |
| 12 | Comment doit-on s’y prendre pour copier un tableau dans un autre ? |
|  | Il faut utiliser une boucle qui recopie valeur par valeur un tableau dans l’autre (ou une fonction du type memcpy (<https://koor.fr/C/cstring/memcpy.wp>) |
| 13 | Comment doit-on s’y prendre pour copier une variable de type structuré dans une autre de même type ? |
|  | Les structures ont les mêmes spécificités que les variables de type « simple », on peut les recopier par affectation :  p1 = p2; |
| 14 | Proposer un exemple avec le point de la question 10. |
|  | struct point p1, p2;  p1 = p2; |
| 15 | Peut-on comparer deux variables de type structuré (i.e. pt1 == pt2) ? |
|  | Non, il faut comparer membre à membre :  if (p2.x == p1.x && p2.y == p1.y) {      // faire quelque chose  } |
| 16 | Comment initialise-t-on une variable de type structuré ? |
|  | On initialise une variable de type structuré en affectant les valeurs des membres entre accolades dans l’ordre de déclaration, ou en spécifiant le nom du membre. |
| 17 | Proposer un exemple d’initialisation des deux points suivants : A(5, 3) et B (2, -2) |
|  | struct point A = { 5, 3 };  struct point B = { .y = -2, .x = 2 }; |
| 18 | Peut-on créer un pointeur sur une variable de type structuré ? Si oui, proposer un exemple avec la structure point. |
|  | On peut créer un pointeur sur une variable de type structuré :  struct point \* p3;  p3 = &p1 ; |
| 19 | Si oui, comment accède-t-on aux membres à partir du pointeur ? Afficher les membres du pointeur précédent avec un printf en exemple. |
|  | On accède aux membres de la structure pointée au moyen de la flèche « -> » :  printf("(%d, %d)\n", p3->x, p3->y); |
| 20 | Si une structure est un type, on peut créer des structures ayant des membres de type structuré, proposer une déclaration d’une structure triangle composée de 3 points. |
|  | Il est possible de créer des structures utilisant des membres de type structuré :  struct point {      int x;      int y;  };  struct triangle {      struct point a;      struct point b;      struct point c;  };  struct triangle t1;  t1.a.x = 1;  t1.a.y = 2;  t1.b.x = -3;  t1.b.y = 4;  t1.c.x = -1;  t1.c.y = -2;  struct triangle t2 = {{1, 2}, {-3, 4}, {-1, -2}};  struct triangle t3 = {.a = {1, 2}, .b = {-3, 4}, .c = {-1, -2}}; |
| 21 | Peut-on créer un tableau de type structuré ? Si oui, proposer un exemple avec la structure point.  Il est possible de créer des tableaux de type structuré : |
|  | struct point p[4];  p[0] = p1;  p[1] = p2; |
| 22 | Si oui, comment accède-t-on aux membres des éléments du tableau ? En écriture pointeur ? En écriture tableau ? Afficher les membres du tableau précédent avec un printf en exemple. |
|  | En écriture tableau :  p[2].x = 4;  En écriture pointeur :  p->x = 4;  printf("(%d, %d)\n", p->x, p[0].y); |
| 23 | Peut-on utiliser des types structurés avec des fonctions ? en type retour ? en passage par valeur ? par adresse ? Donner des exemples pour chacun. |
|  | Les types structurés sont utilisables :  // En type retour  struct point getPoint(int \_x, int \_y) {      struct point p = { \_x, \_y };      return p;  }  // Passage par valeur  void displayPoint(struct point p) {      printf("(%d, %d)", p.x, p.y);  }  // Passage par adresse  void editPoint(struct point \* p, int \_x, int \_y) {      p->x = x;      p->y = y;  } |

# Amélioration du programme

Utiliser un tableau pour stocker les nouvelles fiches au fur et à mesure de leur saisie.

Utiliser des fonctions pour gérer la saisie des fiches et l’affichage de la liste.

Créer la fonction d’affichage du tableau (sans menu de suppression pour l’instant).

# Sauvegarde

Jusque-là les fiches sont créées et stockées en mémoire vive mais pas persistées entre deux exécutions du programme. Il faut donc passer par la création d’un fichier. A la différence des précédents programmes développés, ce fichier ne sera pas lu par un autre programme et n’a donc pas besoin d’être un fichier texte (et de son lot de fastidieuses transformations de chaînes de caractères).

Les fichiers binaires et les structures sont de bons amis.

Répondre aux questions suivantes en réalisant des recherches sur Internet. Pour les éléments de code, tester vos propositions avec votre IDE. Vos recherches feront office de support de cours, soignez votre rédaction.

|  |  |
| --- | --- |
| 24 | Comment ouvre-t-on un fichier binaire ? |
|  | Comme les fichiers textes avec fopen ! Toutefois il faut ajouter un « b » au mode d’accès (« r », « w », « a ») :  FILE \* fichier\_a\_lire = fopen("../chemin\_du\_fichier.extension", "rb"); |
| 25 | La fonction fwrite() permet d’écrire dans un fichier binaire, quel est sa signature ? Décrire chaque paramètre. |
|  | size\_t fwrite(      void \* restrict buffer,      size\_t blocSize,      size\_t blocCount,      FILE \* restrict stream  );  où :   * buffer est un pointeur vers la donnée à enregistrer * blocSize est la taille unitaire de la donnée à enregistrer * blocCount est le nombre de données à enregistrer * stream est le flux vers le fichier binaire   Exemple :  FILE \* fichier = fopen("./fichier.b", "wb");  int a = 42;  char c = 'g';  float pi = 3.1416;  int lost[6] = { 4, 8, 15, 16, 23, 42 };  struct point p = { 1, 2 };  fwrite(&a, sizeof(int), 1, fichier);  fwrite(&c, sizeof(char), 1, fichier);  fwrite(&pi, sizeof(float), 1, fichier);  fwrite(&lost, sizeof(int), 6, fichier);  fwrite(&p, sizeof(struct point), 1, fichier);  fclose(fichier); |
| 26 | Comment récupère-t-on la taille d’une variable ? Est-ce que cela marche avec une variable de type structuré ? |
|  | L’opérateur sizeof permet de récupérer la taille d’une variable ou d’un type de donnée, y compris les types structurés. |
| 27 | La fonction fread() permet de lire dans un fichier binaire, quel est sa signature ? Décrire chaque paramètre. |
|  | size\_t fread(      void \* restrict buffer,      size\_t blocSize,      size\_t blocCount,      FILE \* restrict stream  );  où :   * buffer est un pointeur vers un emplacement pour stocker la donnée lue * blocSize est la taille unitaire de la donnée à lire * blocCount est le nombre de données à lire * stream est le flux vers le fichier binaire   La lecture dans un fichier doit être faite dans le même ordre que l’écriture.  Exemple :  FILE \* fichier = fopen("./fichier.b", "rb");  int b;  char d;  float pi2;  int numbers[6];  struct point p2;  fread(&d, sizeof(char), 1, fichier);  fread(&b, sizeof(int), 1, fichier);  fread(&pi2, sizeof(float), 1, fichier);  fread(numbers, sizeof(int), 6, fichier);  fread(&p2, sizeof(struct point), 1, fichier);  printf("%d\n%c\n%.4f\n%d, %d\n(%d, %d)",          b, d, pi2, numbers[0], numbers[4], p2.x, p2.y);  fclose(fichier);  Résultat :  > 42  > g  > 3.1416  > 4, 23  > (1, 2) |

# Terminer le programme

Créer une fonction de sauvegarde du tableau des fiches dans un fichier binaire.

Créer une fonction de chargement du tableau des fiches à partir d’un fichier binaire.

Intégrer vos fonctions dans votre programme.

Développer la fonctionnalité de suppression d’une fiche.

# Aller plus loin

Trier les fiches par ordre alphabétique du nom de famille

Permettre de modifier une fiche