GIS2A3 — Structures de Données — Épreuve de Juin 2018
Durée 2 heures — Aucun document autorisé

---

1 Petits exercices (5 points)

1.1 Valgrind

Question 1 [1 pt]. Analyser la sortie produite par `valgrind`, donnée en Figure 1.

```
$ valgrind ./a.out
==2210== Memcheck, a memory error detector
==2210== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==2210== Using Valgrind-3.6.0.SVN-Debian and LibVEX; rerun with -h for copyright info
==2210== Command: ./a.out
==2210==
ces rationnels sont différents
==2210==
==2210== HEAP SUMMARY:
==2210==     in use at exit: 16 bytes in 2 blocks
==2210==   total heap usage: 2 allocs, 0 frees, 16 bytes allocated
==2210==
==2210== LEAK SUMMARY:
==2210==    definitely lost: 16 bytes in 2 blocks
==2210==    indirectly lost: 0 bytes in 0 blocks
==2210==      possibly lost: 0 bytes in 0 blocks
==2210==    still reachable: 0 bytes in 0 blocks
==2210==         suppressed: 0 bytes in 0 blocks
==2210== Rerun with --leak-check=full to see details of leaked memory
==2210==
==2210== For counts of detected and suppressed errors, rerun with: -v
==2210== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```

FIGURE 1 – Une sortie produite par `valgrind`.

1.2 Tables de hachage

On considère l'insertion des clés $\square$, $\bigcirc$, $\triangle$, $*$ dans une table de hachage de $N = 13$ alvéoles. La table est gérée avec la technique du double hachage. Les valeurs de hachage des différents éléments sont les suivantes :

	h_1	h_2
$\square$	3	5
$\bigcirc$	7	7
$\triangle$	7	9
$*$	4	3

Question 2 [1 pt]. Insérer les clés dans la table. Donner un schéma de la table résultat. Indiquer les éventuelles collisions.

Question 3 [1 pt]. La fonction h_2 retourne un nombre impair dans tous les cas. Est-ce suffisant, dans ce contexte-ci, pour trouver un alvéole libre s'il en existe au moins un dans la table (justifier) ?

1.3 Complexité

Question 4 [2 pts]. L'algorithme A a une complexité en temps en $O(n)$. L'algorithme B a une complexité en temps en $\Omega(n^2)$. Peut-on affirmer que A est plus rapide que B, quand n tend vers l'infini ? Justifier.

2 Problème (15 points)

Dans l'énoncé, les mots en italique ont un sens précis, expliqué dans le document préparatoire, rappelé en fin d'énoncé.

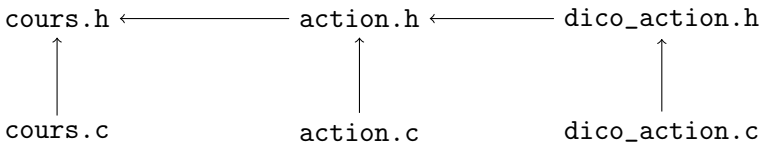

FIGURE 2 – Représentation graphique du logiciel à concevoir.

On cherche à concevoir un logiciel décrit schématiquement Figure 2. Le fichier `cours.h` contient la déclaration d'un type permettant de représenter un *cours*.

Question 5 [2 pts]. Donner une déclaration C pour le type `struct cours`.

2.1 Les actions

Le fichier `action.h` contient la déclaration d'un type permettant de suivre une action sur une période. Cette structure doit contenir le *numéro* de l'action concernée ainsi qu'un « ensemble » de *cours* pour cette action. On ne connaît pas à l'avance le nombre de *cours* à stocker dans la structure. On souhaite mémoriser les *cours* dans l'ordre où ils sont donnés et pas par *jour* croissant.

Question 6 [3 pts]. Donner une déclaration C pour le type `struct action`. Spécifier cette structure en précisant bien la solution choisie pour le stockage de l'ensemble des *cours*. Remarque : il est possible (mais pas nécessaire) de définir des types supplémentaires et/ou d'agrandir le diagramme de la figure 2.

Question 7 [2 pts]. Écrire en C une action `ajout_cours_action`, permettant de rajouter un *cours* à une action. Bien préciser les modes de passage des paramètres.

Question 8 [2 pts]. Écrire en C une fonction `action_complete`, paramétrée par une action `A` et deux *jours* `deb` et `fin`, qui retourne `true` si `A` contient un *cours* pour chaque *jour* entre `deb` (inclus) et `fin` (exclu), `false` sinon.

2.2 Le dictionnaire

On cherche maintenant à concevoir un dictionnaire permettant de mémoriser un ensemble d'actions. On souhaite pouvoir retrouver une action à partir de son *numéro*. On souhaite aussi pouvoir afficher l'ensemble des numéros des actions présentes dans le dictionnaire.

Question 9 [1 pt]. Parmi les implantations de dictionnaires étudiées en cours, y en a-t-il une qui vous semble plus adaptée ? Laquelle ? Pourquoi ?

Question 10 [2 pts]. Donner la déclaration C de la structure choisie pour le dictionnaire. Spécifier cettte structure.

Question 11 [2 pts]. Écrire en C une fonction paramétrée par un dictionnaire `D`, deux *jours* `deb` et `fin` et qui imprime les *numéros* de toutes les actions `A` qui sont présentes dans `D` et complètes, c'est-à-dire qui contiennent un *cours* pour chaque *jour* entre `deb` (inclus) et `fin` (exclu).

Question 12 [1 pt]. On aimerait que, dans la liste de *numéros* affichés par la fonction précédente, les *numéros* qui se terminent par 67 apparaissent en premier. Expliquer comment on pourrait s'y prendre (efficacement).

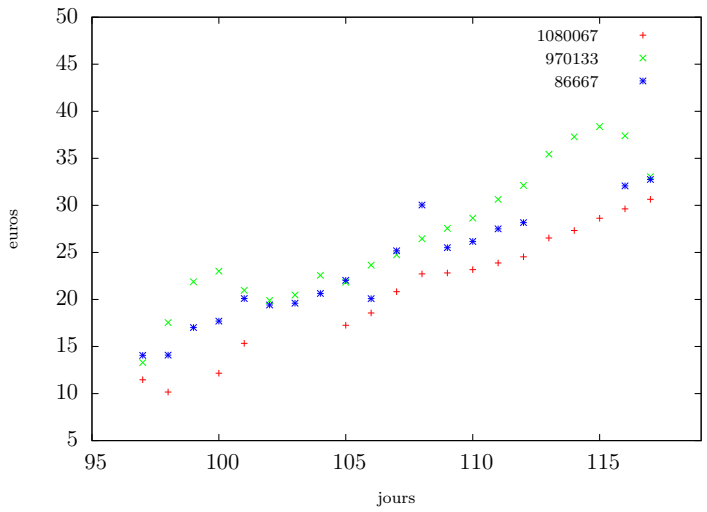

FIGURE 3 – Les cours en euros des trois actions 1080067, 970133 et 86667 entre les jours 97 et 117. Les cours de l'action 1080067 aux jours 99 et $102-104$ sont manquants ainsi que ceux de l'action 86667 aux jours $113-115$.

On cherche à concevoir un logiciel d'analyse des cours d'actions en bourse, sur une période donnée. Le *cours* d'une action est un couple (*jour*, *valeur*) où *jour* est un numéro de jour et *valeur* est une valeur en euros. Chaque action est identifiée par un *numéro*.

Les données de la période à traiter sont reçues sous la forme d'une suite de triplet de nombres : les *numéros* des actions en première colonne et les *cours* sur les deux colonnes suivantes. Les *jours* appartiennent tous à la période (sur l'exemple, ils sont tous compris entre 97 et 117) mais ils n'arrivent pas nécessairement dans l'ordre. Il se peut aussi que certains *cours* soient manquants. Sur l'exemple, les données sont les suivantes (voir Figure 3 pour une représentation graphique) :

```
# numéro jour valeur
 1080067  97 11.47
 1080067  98 10.16
 1080067 100 12.16
  970133  97 14.06
 1080067 106 18.56 <- le cours du jour 106 a été connu avant celui du jour 101
 1080067 101 15.34
   86667  97 13.29
  970133  98 14.09
   86667  98 17.55
[...]
  970133 116 32.07 <- les cours des jours 113-115 ont été connus avec retard
  970133 113 30.81
  970133 114 31.79
  970133 115 31.74
  970133 117 32.76
```