

Identification

Prénom:
Nom:
Code permanent:

Veillez choisir **une seule réponse**, avec une croix (×) ou un crochet (✓).

Q1. Considérez le programme suivant

```
#include <stdlib.h>
#include <stdio.h>

int compare_int(const void* a, const void* b) {
    return *(int*)a - *(int*)b;
}

int main(void) {
    int t[] = {2, 3, 5, 7, 11};
    int a = 4;
    int* i = bsearch(/* TODO 1 */,
                    /* TODO 2 */,
                    5,
                    /* TODO 3 */,
                    compare_int);
    printf("%s\n", i == NULL ? "trouvé!"
           : "pas trouvé!");
}
```

Que doit-on mettre à la place de /* TODO 1 */ si on souhaite rechercher la clé 4?

- *a 1 &a a

Q2. Toujours avec le même programme, que doit-on mettre à la place de /* TODO 2 */?

- &t *t t &&t

Q3. Toujours avec le même programme, que doit-on mettre à la place de /* TODO 3 */?

- sizeof(int&) sizeof(int)
 sizeof sizeof(int*)

Q4. Que représente le type FILE introduit par la bibliothèque `stdio`?

- Un canal standard
 Un fichier en lecture/écriture
 Un fichier en lecture seule

Un flux

Q5. Parmi les structures de données suivantes, laquelle correspond à un type abstrait?

- Une liste doublement chaînée
 Un ensemble
 Un tableau dynamique
 Un tableau statique

Q6. Quel nom donne-t-on à la propriété qu'une structure de données doit satisfaire avant et après qu'on lui ait appliqué une opération?

- Assertion Prédicat
 Invariant Spécification

Q7. Que se passe-t-il si on exécute le programme suivant?

```
#include <stdlib.h>

int main(void) {
    int* p = NULL;
    free(p);
    return 0;
}
```

- Le comportement est indéterminé
 Le comportement est normal
 Une boucle infinie
 Une erreur de segmentation

Q8. Que rapportera Valgrind sur la sortie standard si on l'exécute avec le programme suivant?

```
#include <stdlib.h>

int main(void) {
    int* t[2];
    t[0] = malloc(sizeof(int));
    t[1] = malloc(sizeof(int));
    t[0] = malloc(sizeof(int));
    free(t[0]);
    free(t[1]);
    return 0;
}
```

- Qu'il n'y a aucun problème de mémoire
 Qu'il y a une erreur de segmentation
 Qu'il y a une fuite de mémoire
 Qu'il y a une lecture interdite

[Voir explications page suivante](#)

Q1. La fonction `bsearch` requiert un pointeur pour la clé, même si celle-ci est disponible dans une valeur. Cela permet ensuite de comparer la valeur pointée avec les autres valeurs du tableau. C'est donc pour cela que la réponse est `&a`.

Q2. La fonction `bsearch` requiert un pointeur vers le premier élément du tableau dans lequel on souhaite effectuer une fouille binaire. Dans notre cas, le pointeur est donc `t`. Noter que `&t` n'est pas correct, car il introduit deux niveaux d'indirection (c'est un double pointeur).

Q3. La fonction `bsearch` doit connaître la taille occupée par chacun des éléments du tableau. Dans ce cas-ci, le tableau contient des entiers (`int`), donc cette taille est donnée par `sizeof(int)`.

Q4. Bien que le type `FILE` soit nommé comme cela, il représente un concept plus général que celui de fichier: il s'agit d'un flux de données. Lorsque les données sont du texte, on parle alors de flux de texte. En particulier, les canaux standards `stdin`, `stdout` et `stderr` (qui ne sont pas des fichiers) sont des instances du type `FILE`.

Q5. Un ensemble est un type abstrait, car il peut être représenté de plusieurs façons: à l'aide d'une table de hachage, d'un tableau trié ou d'un arbre binaire de recherche. Les trois autres types sont concrets, car ils correspondent à des implémentations possibles de différents types abstraits. À titre d'exemple, une file peut être implémentée à l'aide d'une liste doublement chaînée, une pile par un tableau dynamique ou par un tableau statique.

Q6. Un invariant est une propriété qu'une structure de données doit respecter en tout temps, c'est-à-dire que toute opération qui modifie la structure doit s'assurer que la propriété soit maintenue à la fin. Une assertion est une instruction qui vise à vérifier si une certaine propriété est respectée. Un prédicat est une fonction qui indique si un objet donné vérifie une certaine propriété. Par extension, une fonction qui retourne une valeur booléenne est aussi appelée un prédicat. Une spécification est un concept plus général, qui vise à décrire ce qui est attendu comme comportement d'un programme dans telle ou telle situation.

Q7. Il n'y a aucun problème avec le programme donné. La fonction `free` a un comportement normal lorsqu'on lui passe la valeur `NULL`, elle ne fait tout simplement rien.

Q8. Lors du troisième appel à la fonction `malloc`, on retourne un pointeur vers un nouveau bloc de mémoire, qui est stocké dans `t[0]`. Or, `t[0]` pointe déjà vers un autre bloc de mémoire, qui ne sera plus référencé à partir de là. On aurait dû appeler la fonction `free` ou utiliser `realloc` pour éviter la fuite de mémoire.