

# Algorithmique et structures de données 2

## - Contrôle de connaissances -

8 avril 2023

VERSION 1

Nom Prénom : \_\_\_\_\_

Numéro d'étudiant.e : \_\_\_\_\_

### Consignes

- durée : 1H;
- lisez tout le sujet et commencez par ce qui vous semble le plus facile, les questions peuvent être faites dans l'ordre que vous voulez;
- un coup d'oeil sur la copie du voisin ou de la voisine donne lieu à un avertissement. Le deuxième avertissement donne lieu à une exclusion de la salle et un 0/20;

Rappels :

- `sizeof(char) = 1`
- Python utilise le passage de paramètre *par variable*. Si une variable d'un type mutable est passée en paramètre d'une fonction et y est modifiée, la modification perdure après la fin de l'exécution de la fonction.

1. Quels sont les risques liés aux fuites mémoires? (1 pt)

[Questions 2–6] (V/F) Parmi les affirmations suivantes lesquelles sont vraies? Justifiez votre réponse avec une courte phrase, un exemple ou un contre-exemple. (2,5 pts)

2. Un tableau est un objet de taille fixe en C.

3. Un pointeur n'est pas toujours une variable

4. La taille d'un pointeur dépend du type de la variable vers laquelle il pointe

5. Si p est de type `int*` alors `*p` est de type `int`.

6. Si p est de type `int*` alors `&p` est de type `int**`.

FIN du V/F

---

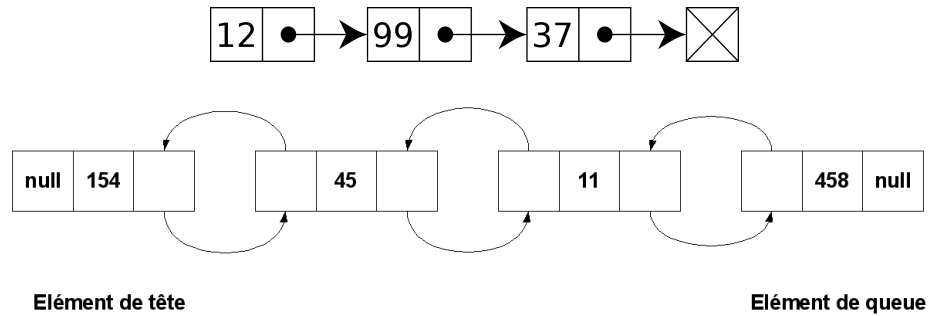
7. Proposez l'implémentation de **structures** en C pour

- une liste simplement chaînée contenant des valeurs entières (exemples ci-dessous).
- une liste doublement chaînée contenant des valeurs entières (exemple ci-dessous)

Nommez correctement votre structure pour qu'on puisse déclarer un pointeur L1 vers une liste simplement chaînée et un pointeur L2 vers une liste doublement chaînée comme ceci (3 pts) :

```
liste_s* L1;
```

```
liste_d* L2;
```



8. Pourquoi les indices des tableaux sont ils des entiers ? Quel est le lien avec la mémoire ? Vous pouvez prendre un exemple pour illustrer. (1,5 pt)

9. Quelle est la différence entre le passage de paramètre *par valeur* et le passage de paramètre *par variable*? Pourquoi utilise-t-on le passage par référence en C? Écrivez une courte fonction en C permettant d'illustrer votre propos. (1 pt.)

10. Sachant le code suivant :

```
5 int main(){
6     int* p=NULL;
7     int tab[4];
8     printf("%ld, %ld, %ld\n",sizeof(tab), sizeof(&tab), sizeof(tab[0]));
9     p = malloc(5*sizeof(int));
10    printf("%ld, %ld\n",sizeof(p), sizeof(*p));
11    p = malloc(4*sizeof(int));
12    printf("%ld, %ld\n",sizeof(p), sizeof(*p));
13    for (int i=0; i<5; i++){
14        if (i%2 == 0) { // i est pair
15            p[i]=i;
16        } else {
17            p[i] = 0;
18        }
19    }
20    for (int i=0; i<5; i++){
21        printf("%d",p[i]);
22    }
23    free(p);
24    return 0;
25 }
```

(a) Qu'affiche l'exécution de ce programme ? (1 pt.)

(b) Ce programme génère-t-il une fuite mémoire ? Si non, pourquoi, si oui de combien d'octet(s) et comment la corriger ? (1 pt.)

(c) Que se passerait-il si on supprimait les lignes 9, 10, 11 et 12, qu'on recompilait puis qu'on exécutait le programme ? Pourquoi ? (1 pt.)

(d) Quel est le test correspondant à une précaution d'usage qui manque dans ce code pour s'assurer de ne pas faire un accès mémoire illégal à l'exécution ? Pourquoi ce test est-il nécessaire ? Quelles sont les lignes à englober dans le bloc if correspondant ? (1 pt.)

11. Citez 4 propriétés d'une variable. (1 pt)

12. Comment s'appelle l'opération qui consiste à utiliser l'opérateur \* sur une variable de type pointeur ? (0,25 pts)

13. Expliquez la raison d'être des tables de hachage ainsi que leur fonctionnement. Vous pouvez vous appuyer sur un exemple (autre que celui vu en TP). Qu'appelle-t-on une collision ? (2,5 pts)

[Questions 14–17] Le stockage des données est-il contigu dans le cas (marquez vrai ou faux à côté des réponses possibles) (2 pts) :

14. d'un tableau

15. d'une liste chaînée

16. d'un ABR

17. d'un type composé (struct)

FIN du V/F

---