

## Examen de Programmation C

Mercredi 19 décembre 2018, de 14h à 16h

### Exercice 1. L'heure est-elle correcte ?

1 + 1 = 2 pts

Soit une heure donnée sous forme h-m-s.

- 1. Écrire un programme qui lit une heure, et qui indique si celle-ci est correcte ou non.
- 2. Modifier le programme pour afficher, si celle-ci est correcte, le durée en h-m-s qui la sépare de 23-59-59.

### Exercice 2. Deux boucles à un point

1 + 1 = 2 pts

- 1. Écrire un programme qui lit une valeur entière  $n > 1$ , puis qui affiche la suite de chiffres suivante, en utilisant une boucle **for** :

122333444455555... $\underbrace{nn \cdots nn}_n$ .

- 2. Modifier le programme pour utiliser une boucle **while**.

### Exercice 3. Approximation de l'intégrale d'une fonction

2 pts

Soit  $f(x)$  une fonction définie, continue et intégrable sur un intervalle  $I = [a, b]$ . On cherche à approcher l'intégrale de  $f(x)$  sur l'intervalle  $I$  par la méthode de Simpson, en divisant  $I$  en  $n$  sous-intervalles :

$$\int_a^b f(x) dx \approx \frac{h}{3} \left( f(x_0) + 2 \cdot \sum_{i=1}^{n/2-1} f(x_{2i}) + 4 \cdot \sum_{i=1}^{n/2} f(x_{2i-1}) + f(x_n) \right)$$

avec  $h = (b - a)/n$  et  $x_i = a + i \cdot h$ .

- 1. Écrire une programme qui lit les valeurs réelles  $a$  et  $b$ , et l'entier  $n$ , puis qui calcule et affiche une approximation de l'intégrale de  $f(x) = \sin(x)$  sur l'intervalle  $[a, b]$ .

### Exercice 4. Suite de Conway

1 + 2 = 3 pts

La suite de Conway est une suite qui commence avec un terme  $C_0 = 1$  et où un terme  $C_i$  se détermine en comptant combien de fois chaque chiffre apparaît dans le terme précédent  $C_{i-1}$ , comme ci-dessous :

- $C_0 = 1$
- $C_1 = 11$  car dans  $C_0$ , il y a 1 chiffre 1
- $C_2 = 21$  car dans  $C_1$ , il y a 2 chiffres 1
- $C_3 = 1211$  car dans  $C_2$ , il y a 1 chiffre 2 et 1 chiffre 1 ...

- 1. Écrire une fonction qui prend un entier  $n$  en paramètre (32 bits), et qui retourne le poids du chiffre décimal non nul de poids le plus fort. Par exemple, dans l'entier le 2018, le chiffre décimal non nul de poids le plus fort est le 2 et son poids est 3.
- 2. Écrire une fonction `conway` qui prend en paramètre un entier  $n$ , et qui calcule et affiche le  $n$ -ième terme de la suite de Conway.

### Exercice 5. Somme et produit d'entiers

1 + 2 = 3 pts

- 1. Écrire une fonction qui prend en paramètre deux entiers  $a$  et  $b$ , et qui renvoie à la fonction appelante la somme et le produit de  $a$  et  $b$ . Illustrer son utilisation dans un programme principal exemple.
- 2. Expliquer la différence entre passage de paramètre par valeur et par adresse, en illustrant la réponse avec un exemple et un schéma.

## Exercice 6. Un exercice renversant

1 + 1 + 1 = 3 pts

Soit  $t$  un tableau d'entiers de taille  $n$ . On souhaite renverser le contenu du tableau  $t$  sans utiliser de tableau intermédiaire. Par exemple, si  $t = \{1, 2, 3, 4\}$ , après renversement, on obtiendra  $\{4, 3, 2, 1\}$ .

- 1. Écrire une fonction qui prend en paramètre un tableau d'entiers de taille 10, puis qui renverse et qui affiche le contenu de  $t$ .
- 2. Écrire un programme principal qui permet d'utiliser cette fonction sur le tableau  $t = \{1, 2, \dots, 9, 10\}$ .
- 3. Donner le prototype de la fonction, qui permet de faire le même traitement sur un tableau de taille  $n$ .

## Exercice 7. Les pointeurs

1 + 2 + 1 = 4 pts

- 1. Rappeler la définition d'un pointeur vue en cours.
- 2. Quel est le résultat de l'exécution du programme suivant. Donner une explication.

```
int
main(void) {
    int a = 17;
    int * ptr = &a;
    *ptr = a + 1;
    printf("--> a= %d et *ptr= %d \n", a, *ptr);
    a = (*ptr--) - 1;
    printf("--> a= %d et *ptr= %d \n", a, *ptr);
    return 0;
}
```

Soit le début de programme suivant.

```
int
main(void) {
    int tab[17];
    // ... partie a completer
    return 0;
}
```

- 3. Compléter le programme pour lire le contenu du tableau `tab` en utilisant l'arithmétique des pointeurs.

## Exercice 8. Sudoku

2 + 3 = 5 pts

Une grille de Sudoku se compose de 9 blocs de 9 éléments, disposés de la manière suivante.

1								4
				9				8
			6	8		5	7	1
	4		8					
			5	2	7			3
7	5		3			9	1	
	4		1					5
8	7	4		6				
2		5			8			6

L'objectif est, à partir d'une configuration initiale, de remplir la grille de telle sorte qu'elle reste valide, c'est-à-dire, que chaque ligne, chaque colonne et chaque bloc ne contiennent qu'une seule occurrence de chaque nombre entre 1 et 9. Ici on modélise une grille par un tableau 2D de taille  $9 \times 9$ , où chaque case contiendra un chiffre en 1 et 9, ou bien 0 pour indiquer une case vide.

- 1. Écrire une fonction qui prend une grille de Sudoku en paramètre, non forcément remplie, qui vérifie si cette grille est valide, et qui retourne un entier correspondant (0 pour faux et 1 pour vrai).
- 2. Écrire une fonction qui, étant donnée une grille de Sudoku non forcément remplie mais valide, la remplit complètement en s'assurant qu'elle reste valide. Expliquer chacun des paramètres.