

Licence Professionnelle EON, contrôle des connaissances : Programmation en C
Luca Saiu — novembre 2018

Aucun document autorisé. Toute communication entre étudiants est interdite. Téléphones portable et ordinateurs éteints et rangés.

Le barème est donné à titre indicatif. **Il a une pénalité pour toute réponse incorrecte** : ne rien écrire est préférable à une erreur. Toute réponse ambiguë sera considérée comme incorrecte.

Les réponses sont à donner directement sur le papier de l'énoncé.

Durée : 90 minutes. La partie B est au verso.

Les programmes ou fonctions suivants, même quand ils sont corrects, peuvent être mal décalés (« indentés ») ; du code mal décalé **n'est pas** considéré comme erroné ou incorrect. Faites attention aux point-virgules ! Du code qui ne termine jamais, également, **n'est pas** considéré comme erroné ou incorrect.

Il y a une différence entre une fonction renvoyant un résultat est une commande affichant du texte sur le terminal ; il y a une différence entre une fonction acceptant un paramètre et une valeur lue à partir du terminal. Suivez **exactement** les consignes de l'énoncé.

Nom (lisible) : _____

Prénom (lisible) : _____

— **Partie A : compréhension** —

(chaque question a une seule réponse correcte)

```
11 return 0;  
12 }
```

Question A.1. (2 points) La phrase du langage C

```
return (a + 3);
```

- | | |
|--|-----------------------------------|
| a) est une expression | e) est une définition de fonction |
| b) est une commande | f) est un appel de fonction |
| c) est un bloc | g) est mal écrite |
| d) est un prototype (ou déclaration) de fonction | |

- | | |
|--------------|----------------------|
| a) affiche 3 | f) affiche 8 |
| b) affiche 4 | g) affiche 9 |
| c) affiche 5 | h) affiche 10 |
| d) affiche 6 | i) est mal écrit |
| e) affiche 7 | j) ne termine jamais |

Question A.2. (2 points) Le programme suivant ...

```
1 #include <stdio.h>  
2  
3 int  
4 main (void)  
5 {  
6     int i;  
7     for (i = 0; i < 10; i ++)  
8         print ("foo\n");  
9     print ("foo\n");  
10    return 0;  
11 }
```

- | | |
|------------------------|-------------------------|
| a) affiche foo 9 fois | e) affiche foo 20 fois |
| b) affiche foo 10 fois | f) affiche foo 100 fois |
| c) affiche foo 11 fois | g) est mal écrit |
| d) affiche foo 12 fois | h) ne termine jamais |

Question A.3. (2 points) Le programme suivant ...

```
1 #include <stdio.h>  
2  
3 int  
4 main (void)  
5 {  
6     int i;  
7     int n = 1;  
8     for (i = 0; i < 4; i ++)  
9         n = n + i;  
10    printf ("%i\n", n);
```

Question A.4. (3 points) La fonction foo suivante ...

```
1 void  
2 foo (int n)  
3 {  
4     if (n != 0)  
5         n = 0;  
6     while (n < 3)  
7         n = 0;  
8     return n;  
9 }
```

- | | |
|------------------------------------|-----------------------|
| a) renvoie la valeur initiale de n | d) renvoie toujours 3 |
| b) renvoie toujours 0 | e) renvoie toujours 4 |
| c) renvoie toujours 2 | f) est mal écrit |
| | g) ne termine jamais |

Question A.5. (3 points) La fonction foo suivante ...

```
1 int  
2 quux (bool frob, int x)  
3 {  
4     while (frob)  
5         x = 1;  
6     return 7;  
7 }
```

- | | |
|---|---|
| a) renvoie toujours true | f) peut renvoyer 7 ou renvoyer true, selon la valeur de frob |
| b) renvoie toujours false | g) peut renvoyer 7 ou renvoyer false, selon la valeur de frob |
| c) renvoie toujours 7 | h) est mal écrit |
| d) renvoie toujours la valeur de x | i) ne termine jamais |
| e) peut renvoyer 7 ou boucler infiniment, selon la valeur de frob | |

— Partie B : programmation —

(ce n'est pas nécessaire de répondre à toute question de cette partie pour avoir la note maximale, si le reste est correct)

Question B.1. (7 points) Écrivez la définition d'une fonction `m` qui, étant donné un paramètre entier `n`, affiche une *table de multiplication* de 1 à `n` compris, sans renvoyer aucun résultat, dans un format similaire à ce qui suit. Par exemple l'appel `m (3)` affichera :

```
1 fois 1 = 1
1 fois 2 = 2
1 fois 3 = 3
2 fois 1 = 2
2 fois 2 = 4
2 fois 3 = 6
3 fois 1 = 3
3 fois 2 = 6
3 fois 3 = 9
```

Question B.2. (7 points) Écrivez la définition d'une fonction `s` qui, étant donné deux paramètres entiers `a` et `b`, renvoie la somme des nombres naturels entre 1 et `a` (compris) qui sont des multiples de `b`. Par exemple l'appel `s (10, 3)` renverra 18, car $3 + 6 + 9 = 18$ et les seuls multiples de 3 dans $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ sont 3, 6 et 9.