

14

Bien



14,00



COMBETTE
Elise

PIL
2013

jeudi 10 Avril

CE Programmation en C

Exercice 1 (2)

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

int main()
{
    int n;
    printf("Saisir un entier:");
    scanf("%d", &n);
    if ((n % 2) == 0) printf("%d est pair.", n);
    else printf("%d est impair.", n);
    return 0;
}
```

Exercice 2 (4)

```
int tempo;
tempo = x;
x = z;
z = tempo;
tempo = y;
y = z;
z = tempo;
```

4 affectations suffisent

Exercice 3

(2)

2.1. est ~~fauxse~~, l'addition de deux nombres entiers positifs donnera forcément un entier positif.
 $a > 0$ et $b > 0 \Rightarrow a + b > 0$.

2.2. est ~~fauxse~~, car la condition du "if" sera toujours ~~vraie~~ ^{non}, même quand $x \neq y$: le "=" est le symbole d'affectation, alors que le "=" mathématique en C est "==".

2.3. est ~~fauxse~~, on doit écrire la condition de la façon suivante: " $x < y$ & & $y < z$ ".
↳ ~~justifié~~ (il n'y a pas d'erreur de syntaxe)

2.4. est ~~fauxse~~ à cause de l'ordre de priorité des opérations, on doit mettre des parenthèses pour forcer la priorité: " $(x/y) > 1$ ".
↳ ~~justifié~~ (il n'y a pas d'erreur de syntaxe)

2.5. est ~~fauxse~~, la condition est sémantiquement incorrecte: " $(x == y) \parallel (x == z)$ ".
↳ ~~justifié~~.

Exercice 4

(1,5)

```
int n, i, c;  
printf("Saisir la longueur du demi-carré d'étoiles: ");  
scanf("%d", &n);
```

```
for (i = 0; i < n; i++)
```

⚠ C'est la condition de continuation de la boucle qu'il faut écrire.

```
{  
    for (c = 0; c < (n - i); c++) printf("*");  
    printf("\n");  
}
```

N'affiche pas les espaces avant les étoiles? \Rightarrow also faux

Exercice 5 (2,5)

// Générateur de nombres aléatoires.

void initialisation(int t[100])

```
{  
    int i=0;  
    for (i=0; i<100; i++) t[i] = i * rand();  
}
```

int traitement(int t[100], int estCroissant)

```
{  
    int i=0;  
    for (i=0; i<99; i++)  
    {  
        if (t[i] < t[i+1]) estCroissant = 0; | arrêter dès que possible (i.e. dès que pas croissant)  
    }  
    return estCroissant;  
}
```

↳ n'a pas lieu d'être un paramètre

(Devrait être une variable ~~globale~~ de travail locale initialisée localement)

void affichage(int estCroissant)

```
{  
    if (estCroissant == 1) printf("Le tableau est croissant");  
    else printf("Le tableau n'est pas croissant");  
}
```

int main()

```
{  
    int t[100];  
    int estCroissant = 1;  
    initialisation(t[100]); | t[100] si le 100 était de t (qui n'en fait pas !)  
    estCroissant = traitement(estCroissant);  
    affichage(estCroissant);  
    return 0;  
}
```


u