

**Exercice 1 : Arithmétique et expressions**

Considérez les déclarations suivantes :

```
char c = 3;    short s = 7;  
int i = 3;     long l = 4;  
float f = 3.3; double d = 7.7;
```

Pour chacune des expressions ci-dessous, indiquez leur type et leur valeur. Par exemple l'expression `2 / 3 * c` donne `(int)0` car `2 / 3` est égal à 0 en division entière, et comme l'opération passe par l'ALU, le resultat est promu en `int` avant d'être multiplié par `c`.

(a) `c / 2`

(a) \_\_\_\_\_

(b) `s + c / 10`

(b) \_\_\_\_\_

(c) `l + i / 2.0`

(c) \_\_\_\_\_

(d) `d + f`

(d) \_\_\_\_\_

(e) `(int)d + f`

(e) \_\_\_\_\_

(f) `(int)d + l`

(f) \_\_\_\_\_

(g) `c << 2`

(g) \_\_\_\_\_

(h) `s & 0xf0`

(h) \_\_\_\_\_

(i) `s && 0xf0`

(i) \_\_\_\_\_

(j) `d + f == s + l`

(j) \_\_\_\_\_

**Exercice 2 : Analyse de code**

Dans chacune des structures de contrôle ci-dessous, indiquer la nature de l'erreur.

(a) 

```
double x = 100.0;  
size_t i = 0;  
do  
    x = x / 2.0;  
    i++;  
while (x > 1.0);
```

.....  
.....

(b) `long x = 100; if (x = 0) printf("Erreur : la valeur 0 est interdite !\n");`

.....  
.....

(c) `double x = 100.0;  
switch (x) {  
 case 0:  
 printf("x est nul.\n");  
 break;  
 default:  
 print("OK.\n");  
}`

.....  
.....

(d) `for (int i = 0; i < 10; i++); {  
 printf("%d\n", i);  
}`

.....  
.....

(e) `int i = 0;  
while i < 100  
{  
 printf("%d\n", ++i);  
}`

.....  
.....

**Exercice 3: Lecture de code**

On s'intéresse ici au passage par adresse. Observez le programme suivant et indiquez ce que vous voyez sur la sortie standard.

```

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

int test(int a, int *b, int *c, int *d) {
    a = *b;
    *b = *b + 5;
    *c = a + 2;
    d = c;
    return *d;
}

int main() {
    int a = 0, b = 100, c = 200, d = 300, e = 400;
    e = test(a, &b, &c, &d);
    printf("%05d %d %d %d %d", a, b, c, d, e);
}

```

.....

**Exercice 4: Programmation**

(a) Écrire une fonction qui reçoit une chaîne de caractère en paramètre et qui retourne vrai si la chaîne est un palindrome, c'est à dire qu'elle se lit de la même manière de gauche à droite et de droite à gauche. Utiliser la syntaxe pointeur pour le paramètre chaîne.

.....  
.....  
.....  
.....  
.....

(b) Écrire une fonction qui reçoit en paramètre un **tableau d'entiers** et qui retourne la position de la **première** occurrence d'une valeur dans ce tableau, ou -1 si la valeur n'est pas présente. Utiliser la syntaxe pointeur pour le paramètre tableau.

.....  
.....  
.....  
.....  
.....

(c) Écrire une fonction qui calcul la longueur totale des segments de droite dont les points sont reçus en paramètre. Les données sont un tableau composé de N par 2. Les indices du sous tableau sont les coordonnées X et Y des points.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....