

## Module : Programmation en Langage C

### Filière Génie Civil / 1<sup>er</sup> Année / Année 2017-2018

#### Corrigé des TD/TP N° 4

#### Exercice 1

Soit le fragment de code suivant :

```
int a[10], *ptr1, *ptr2;  
ptr1 = &a[4]; ptr2 = &a[6];
```

Laquelle des instructions suivantes est incorrecte ?

- (a) `ptr1 + 2` (**correcte**)
- (b) `ptr2 - 2` (**correcte**)
- (c) `ptr2 + ptr1` (**incorrecte**)
- (d) `ptr2 - ptr1` (**correcte**)

#### Exercice 2

Soit le fragment de code suivant :

```
int a = 176, *ptr = &a, n = *ptr;
```

Si l'adresse de `a` est 2801 et l'adresse de `ptr` est 2600. Quelle est la valeur de chacune des expressions suivantes :

- (a) `&a` (valeur = **2801**)
- (b) `*ptr` (valeur = **176**)
- (c) `n` (valeur = **176**)
- (d) `&ptr` (valeur = **2600**)

#### Exercice 3

Si `i` désigne un entier et `p` et `q` deux pointeurs sur un entier, lesquelles des instructions suivantes sont correctes ?

- (a) `p = i;` (**incorrecte**)
- (b) `*p = &i;` (**incorrecte**)
- (c) `&p = i;` (**incorrecte**)
- (d) `p = &q;` (**incorrecte**)
- (e) `p = q;` (**correcte**)

(f) `p = *q;`      (**incorrecte**)

(g) `*p = *q;`      (**correcte**)

(h) `q = &i;`      (**correcte**)

#### **Exercice 4**

Déterminer la valeur de chacune des variables après l'exécution du code suivant :

```
int m = 44;          /* instruction 1 */
int * p = &m;       /* instruction 2 */
int n = (*p)++;    /* instruction 3 */
int * q = p - 1;   /* instruction 4 */
```

On suppose que la variable `m` est rangée dans la mémoire à l'adresse **F8C**.

Variables	Instruction 1	Instruction 2	Instruction 3	Instruction 4
<b>m</b>	<b>44</b>	<b>44</b>	<b>45</b>	<b>45</b>
<b>p</b>	<b>?</b>	<b>F8C</b>	<b>F8C</b>	<b>F8C</b>
<b>n</b>	<b>?</b>	<b>?</b>	<b>44</b>	<b>44</b>
<b>q</b>	<b>?</b>	<b>?</b>	<b>?</b>	<b>F88</b>

#### **Exercice 5**

1. Déclarer une structure « Complexe » pour stocker un nombre complexe (représentation algébrique). On rappelle qu'un nombre complexe se compose d'une partie réelle et d'une partie imaginaire.

```
struct Complexe
{
    double re; /* la partie réelle */
    double im; /* la partie imaginaire */
};
```

2. Ecrire un programme qui saisit deux nombres complexes `z1` et `z2`, puis calcule `z1+z2`, `z1-z2`, `z1*z2`, `z1/z2`, `|z1|` et `arg(z1)`.

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

struct Complexe
{
    double re; /* la partie réelle */
    double im; /* la partie imaginaire */
};

main()
{
    struct Complexe z1, z2, z3, z4, z5, z6;
    double module, argument, temp;

    printf("Saisir z1 : ");
    scanf("%lf%lf", &z1.re, &z1.im);
```

## Module : Programmation en Langage C

### Filière Génie Civil / 1<sup>er</sup> Année / Année 2017-2018

```
printf("Saisir z2 : ");
scanf("%lf%lf", &z2.re, &z2.im);

/* calcul de z1 + z2 dans z3 */
z3.re = z1.re + z2.re;
z3.im = z1.im + z2.im;
printf("z1 + z2 = %lf+%lfi\n", z3.re, z3.im);

/* calcul de z1 - z2 dans z4 */
z4.re = z1.re - z2.re;
z4.im = z1.im - z2.im;
printf("z1 - z2 = %lf+%lfi\n", z4.re, z4.im);

/* calcul de z1 * z2 dans z5 */
z5.re = z1.re * z2.re - z1.im * z2.im;
z5.im = z1.re * z2.im + z1.im * z2.re;
printf("z1 * z2 = %lf+%lfi\n", z5.re, z5.im);

/* calcul de z1 / z2 dans z6 */
temp = pow(z2.re, 2.0) + pow(z2.im, 2.0);
z6.re = (z1.re * z2.re + z1.im * z2.im) / temp;
z6.im = (-z1.re * z2.im + z1.im * z2.re) / temp;
printf("z1 / z2 = %lf+%lfi\n", z6.re, z6.im);

/* calcul de |z1| dans module */
module = sqrt(pow(z1.re, 2.0) + pow(z1.im, 2.0));
printf("|z1| = %lf\n", module);

/* calcul de arg(z1) dans argument */
argument = atan(z1.im / z1.re);
printf("arg(z1) = %lf\n", argument);
system("pause");
}
```