

## Devoir Surveillé

Nom :

Prénom :

### Exercice 1 — *Questions de cours* (3 points)

On veut écrire une fonction `estZero` qui teste si son argument est égal à zéro. On doit pouvoir compiler le code suivant :

```
int n = lireInt(); /* demande un entier (positif) à l'utilisateur */
while (!estZero(n)) {
    printf("pas encore zéro\n");
    n = n / 2;
}
```

- (1) Quel type doit retourner cette fonction ? Pourquoi ?

- (2) Écrivez cette fonction, n'oubliez pas de déclarer les arguments (et leurs types) et le type de retour de la fonction. Vous aurez les points si le code ci-dessus peut marcher avec votre fonction.

- (3) L'exemple de code ci-dessus est-il une boucle infinie ? Justifiez.

### Exercice 2 — *QCM* (3 points)

Pour chacun des morceaux de codes suivants remplissez le tableau associé en répondant aux questions suivantes :

- Est-il accepté par le compilateur sans erreur ni warning ? Si non justifiez en quelques mots.
- Si oui est-ce que l'exécution sera correcte ? Si non justifiez en quelques mots.
- Si les deux réponses ci-dessus son « OK » et que le programme comporte une variable `x` alors donnez sa valeur à la fin du morceau de programme.

*Attention* : les réponses fausses sans justification ou avec une justification sans fondement valent un malus. Ne répondez que si vous êtes sûr de vous.

1.	<pre>char t = "t";</pre>	compilation		exécution	
		erreur	OK	incorrect	OK,x =
		warning			
		Justification :			
2.	<pre>char t = 't';</pre>	compilation		exécution	
		erreur	OK	incorrect	OK,x =
		warning			
		Justification :			
3.	<pre>char * t = "toto";</pre>	compilation		exécution	
		erreur	OK	incorrect	OK,x =
		warning			
		Justification :			
4.	<pre>char t = 'toto';</pre>	compilation		exécution	
		erreur	OK	incorrect	OK,x =
		warning			
		Justification :			
5.	<pre>char t[] ={'a','1','\$','m','6','-'}; char x = t [6];</pre>	compilation		exécution	
		erreur	OK	incorrect	OK,x =
		warning			
		Justification :			
6.	<pre>double t[] = { 4.5, 5.7, 7.9 }; int x = t[0] + t[1] + t[2];</pre>	compilation		exécution	
		erreur	OK	incorrect	OK,x =
		warning			
		Justification :			
7.	<pre>double a=4.5, b=5.7, c=7.9; int x = a + b + c;</pre>	compilation		exécution	
		erreur	OK	incorrect	OK,x =
		warning			
		Justification :			

### Exercice 3 — *Exécution de programmes*

Chacun de ces programmes suivants est sensé réaliser l’affichage suivant :

```
1
2
3
4
5
```

avec un dernier saut de ligne après le 5.

Cochez les programmes qui vous semblent faire effectivement cela.

1. ☐

```
int i = 0;
while(i<5) {
    i = i + 1;
    ecrireInt(i);
    ecrireSautDeLigne();
}
```

2. ☐

```
int i = 1;
while(i<=5) {
    ecrireInt(i);
    i = i + 1;
}
ecrireSautDeLigne();
```

3. ☐

```
int i = 0;
while(i<5) {
    ecrireInt(i);
    i = i + 1;
}
```

4. ☐

```
int i = 0;
while(i<5) {
    ecrireInt(i);
    ecrireSautDeLigne();
    i = i + 1;
}
```

5. ☐

```
int i = 5;
while(i>=1) {
    ecrireInt(i);
    ecrireSautDeLigne();
    i = i - 1;
}
```

6. ☐

```
int i = 5;
while(i>=1) {
    ecrireInt(6-i);
    ecrireSautDeLigne();
    i = i - 1;
}
```

7. ☐

```
int i = 1;
while(i<=5) {
    ecrireInt(i);
    ecrireSautDeLigne();
}
```

8. ☐

```
int i = 1;
while(i<=5) {
    ecrireInt(i);
}
ecrireSautDeLigne();
```

9. ☐

```
int i = 1;
while(i<=6) {
    ecrireInt(i);
    ecrireSautDeLigne();
    i = i + 1;
}
```

10. □

```
int i = 2;
while(i<=6) {
    ecrireInt(i-1);
    ecrireSautDeLigne();
    i = i + 1;
}
```

#### Exercice 4 — Boucles simples (6 points)

1. Écrivez une procédure `void comptejusqua(int n)` qui affiche tous les nombres de 0 jusqu'à `n` sur une même ligne. Par exemple `comptejusqua(6)` affiche à l'écran `0123456`.

2. Écrivez une fonction `int majoritePositifs(int t [], int taille)` qui retourne 1 si strictement plus de la moitié des cases de `t` (entre 0 et `taille-1`) contiennent des entiers positifs (ou nuls) et zéro sinon.

3. Écrivez une fonction `int memesEcartes(int t [], int taille, int delta)` qui retourne 1 si chaque case (entre 1 et `taille-1`) du tableau `t` est exactement égale à (`delta` + la valeur de la case précédente). Par exemple sur le tableau `{2;5;8;11}` et une valeur de `delta=3`, la fonction retourne 1 car chaque case (sauf la première évidemment) est égale à 3 + la case précédente.

#### Exercice 5

Considérez la procédure suivante qui prend en entrée `a` et `b` et calcule  $2b + 2a$ . Complétez les pointillés par le code qui vous semble correct.

```
int f(int a, int b) {  
    b = 2*b;  
    b = b+a;  
    ..... // A compléter  
    ..... // A compléter  
}
```

**Exercice 6 — Affichage de programme** (2 points)

- (1) Qu'affiche le programme suivant si le nombre entré par l'utilisateur est 10?

```
void main(){
    int a,i,compt;
    i=0;
    compt=0;
    ecrireString("Entrez un nombre");
    a=lireInt();
    while(i<=a) {
        if(i%2==0){
            ecrireString(" "+i);
            ecrireSautDeLigne();
            compt=compt+1;
        }
        i=i+1;
    }
    ecrireString("En tout "+ compt + " entiers.");
}
```

- (2) Donnez les affichages du programme suivant :

```
void main(){
    int a =5;
    int b= 7;
    int c =13;
    c = calc(a,b);
    ecrireString("a vaut: ");
    ecrireInt(a);
    ecrireSautDeLigne;
    ecrireString("b vaut: ");
    ecrireInt(b);
    ecrireSautDeLigne;
    ecrireString("c vaut: "+c );
    ecrireInt(c);
    ecrireSautDeLigne;
}
int calc(int a, int c){
    int b;
    ecrireString("a vaut: ");
    ecrireInt(a);
    ecrireSautDeLigne;
    ecrireString("c vaut: ");
    ecrireInt(c);
    ecrireSautDeLigne;
    b = a+c;
    a = b+1;
    return b;
}
```

