

## Informatique S1 – Programmation C

### Exercices

#### TD 2 : Structure générale d'un programme C

Dans ce TD, nous allons réaliser des exercices couvrant la structure générale d'un programme en C et l'usage des variables. Nous allons faire des programmes simples en utilisant les concepts vus en cours.

#### Compilation

Avant de commencer les exercices, il faut savoir comment compiler ses programmes. Pour compiler un programme en C, on utilisera le compilateur « gcc » (Gnu C). Il faut donc ouvrir un terminal, aller sur le répertoire où se trouvent le (ou les) source(s) qu'on veut compiler, et taper la commande « gcc -o source source.c » (en remplaçant le « source.c » par le nom du fichier qu'on veut compiler). Par exemple :

```
gcc -o helloworld helloworld.c
```

Puis, pour exécuter le programme, il suffit de taper :

```
./helloworld
```

(remplacer le « helloworld » par le nom indiquer avec l'option « -o » de la commande gcc)

#### Exercice 1

- a) Le programme ci-dessous présente plusieurs erreurs. Trouvez-les.

```
#include <stdio.h>

main ()
    int myInt1, myInt2;

    printf ("Entrer le nombre d'heures : ")
    scanf ("%d", myInt1);

    myint2 = myint1 * 3600;

    printf ("Il y a %d s en %d h " myInt1, myInt2);
}
```

- b) Analyser le programme ci-dessus. Décrire que devrait faire chaque ligne.  
c) Corriger le programme ci-dessus et l'exécuter.

#### Exercice 2

Écrire un programme en langage C qui affiche le message suivant :

Bonjour, entrer un nombre entier s'il vous plait :

L'utilisateur doit alors entrer un nombre entier (par exemple 72), et le programme doit répondre avec le même numéro que l'utilisateur a fourni :

Le nombre que vous avez entre est 72

### Exercice 3

Écrire un programme en langage C qui lit un nombre entier fourni par l'utilisateur, calcule et affiche à l'utilisateur le *double* de ce nombre. Comment faire pour que le programme affiche également le triple ? (Astuce : pour lire le nombre, utiliser « scanf » vu en cours).

### Exercice 4

- Écrire un programme qui demande à l'utilisateur deux nombres entiers et qui affiche la somme entre les deux nombres.
- Étendre le programme précédent pour qu'il affiche également la soustraction entre les deux nombres lus (premier – second).
- Étendre encore le programme précédent pour qu'il affiche la division (premier / second) et le reste de la division (premier % second).
- À partir de la dernière version du programme précédent, que se passe-t-il si l'utilisateur introduit les nombres « 9 » et « 2 » respectivement ? Quelles réponses aurons-nous pour la division et le reste ? Comment faire pour afficher le résultat de la division lorsque ce résultat n'est pas un nombre entier ?
- Que se passe-t-il pour l'addition si on introduit les nombres « 2147483647 » et « 1 » ? Expliquer.

### Exercice 5

- Tracer l'exécution du programme ci-dessous (pour chaque point d'observation, noter sur un tableau la valeur de chaque variable). À la fin de l'exécution (point d'observation 3), quel sera la valeur de la variable « fahr » si l'utilisateur fournit la valeur « 10 » ? Vérifier en tapant et en exécutant le programme.
- Quel est l'effet du « %f » dans le « printf » à la ligne 22 ? Quel est la différence par rapport au « %d » utilisé précédemment ?

```
1
2  #include <stdio.h>
3
4  /* Conversion C-F */
5  main() {
6      float celc, fahr;
7
8      fahr = 0.0;
9      celc = 0.0;
10
11     //point d'observation 1
12
13     printf("Temperature (C): ");
14     scanf("%f",&celc);
15
16     //point d'observation 2
17
18     fahr = (celc*9)/5 + 32;
19
20     //point d'observation 3
21
22     printf ("Temperature (F): %f\n", fahr);
23
24 }
25
```

**Exercice 6**

Tracer l'exécution du programme ci-dessous. Que fait ce programme ? Vérifier en le tapant et en l'exécutant.

```
#include<stdio.h>

/* Suite de Fibonacci de 1 a 7
 * La suite de Fibonacci est un probleme mathematique defini comme suit:
 *   f0 = f1 = 1
 *   fn+2 = fn + fn+1, pour tout n >= 0
 */

main()
{
    /* variables:
     * - p comme precedent
     * - s comme suivant
     */
    int p, s, n;

    //Point d'observation 1
    printf("%i\n", n);

    p=1;
    s=1;

    //Point d'observation 2

    n=s;
    s=p+s;
    p=n;    //Point d'observation 3
    n=s;
    s=p+s;
    p=n;    //Point d'observation 4
    n=s;
    s=p+s;
    p=n;    //Point d'observation 5
    n=s;
    s=p+s;
    p=n;    //Point d'observation 6
    n=s;
    s=p+s;
    p=n;    //Point d'observation 7

    printf("Le septieme terme de la suite de Fibonacci vaut %i\n", s);
}
```

## **Exercice 7**

- a) On souhaite créer un programme qui fait l'échange entre les valeurs de deux variables (la variable « a » reçoit la valeur de la variable « b » et vice-versa). Le programme « échange1.c » illustré ci-dessous résout-il ce problème ? Si non, comment pouvons-nous résoudre cette question ?

```
#include<stdio.h>
main()
{
    int a, b;
    a=3;
    b=5; //Point d'observation 1
    printf("a vaut %i\n", a);
    printf("b vaut %i\n", b);
    a=b;
    b=a; //Point d'observation 2
    printf("a vaut %i\n", a);
    printf("b vaut %i\n", b);
}
```

- b) Pour résoudre le problème posé par la question 7a, proposer un programme pour échanger les valeurs des variables « a » et « b » à l'aide d'une troisième variable « t », dite variable « tampon ».