

TP1 – Les bases

Projet de programmation M1

22 Septembre 2015

Exercice 1. On peut utiliser la fonction `scanf` pour récupérer des informations saisies par l'utilisateur. Pour stocker un nombre saisi par l'utilisateur dans une variable `int n`, on pourra utiliser :

```
int n;  
scanf("%d", &n);
```

1. Écrire un programme qui demande l'année courante et l'année de naissance de l'utilisateur et calcule son âge.

En C, il existe de nombreuses façons de tester des choses :

Commande	Résultat
<code>a == b</code>	Vrai si et seulement si <code>a</code> et <code>b</code> ont la même valeur.
<code>a != b</code>	Vrai si et seulement si <code>a</code> et <code>b</code> ont une valeur différente.
<code>a < b</code>	Vrai si et seulement si <code>a</code> est strictement inférieur à <code>b</code> .
<code>a <= b</code>	Vrai si et seulement si <code>a</code> est inférieur ou égal à <code>b</code> .
<code>a > b</code> , <code>a >= b</code>	idem
<code>condition1 && condition2</code>	Vrai si <code>condition1</code> et <code>condition2</code> sont vraies.
<code>condition1 condition2</code>	Vrai si l'une des deux conditions est vraie.
<code>!(condition)</code>	Vrai si la <code>condition</code> est fausse.

2. Reprendre le programme précédent et afficher une erreur si l'année de naissance est postérieure à l'année courante.
3. Si une année est divisible par 4, alors elle est bissextile sauf si elle est divisible par 100 et pas par 400. On rappelle que le reste de la division euclidienne de `a` par `b` s'écrit `a % b` en C. Écrire un programme qui demande une année à l'utilisateur et indique si elle est bissextile ou pas. Essayez de n'utiliser qu'un seul bloc `if-then-else`.
4. À votre avis, que fait ce programme :

```
#include <stdio.h>  
  
int main() {  
    int x=0;  
    if (x=5) {  
        printf("La valeur de x est %d.\n", x);  
    }  
    else {  
        printf("On sait bien que x vaut 0.\n");  
    }  
    return 0;  
}
```

Exécute-le sur votre machine. Quoi?!

Exercice 2. On considère le programme suivant :

```
#include <stdio.h>  
  
int main() {  
    int n = 1;  
    while(n != 0) {  
        printf("Entrez un nombre (0 pour quitter) : ");
```

```

scanf("%d", &n);
printf("%d*d vaut %d.\n", n,n,n*n);
}
return 0;
}

```

1. Que fait ce programme ? Modifiez-le pour qu'il s'arrête lorsque l'utilisateur entre un entier divisible par 3.
2. Écrire un programme qui demande en boucle un entier à l'utilisateur jusqu'à ce qu'il entre l'entier 0. Votre programme affiche alors la somme de tous les entiers saisis jusque-là.
3. La machine à café : un café à la machine vaut 40 centimes. Écrire un programme qui demande en boucle à l'utilisateur la valeur (en centimes) des pièces qu'il entre dans la machine. Lorsqu'il a mis assez de monnaie, la machine arrête de demander des pièces et affiche la monnaie due. Par exemple :

```

Entrez une pièce : 5
Entrez une pièce : 20
Entrez une pièce : 50
Merci, je vous dois 35 centimes.

```

Lorsqu'il faut gérer plusieurs cas en C, on peut utiliser la construction `switch` ainsi :

```

switch(x) { // x est une variable ; int par exemple
case 0:
    // cas x=0...
    break;
case 1:
    // cas x=1
case ...
default:
    // comportement par défaut
}

```

4. Améliorez ce programme pour qu'il propose d'abord à l'utilisateur de choisir entre un café (40c), un chocolat (45c) et un capuccino (50c). Le programme doit aussi refuser les pièces qui n'existent pas.

Exercice 3. Il existe plusieurs façons de représenter les entiers en C. C'est à vous de prévoir les besoins de votre programme pour savoir quelle représentation choisir.

Nom	Taille	Entiers représentés	Utilisation avec <code>printf</code>
<code>short</code>	16 bits	de -2^{15} à $2^{15} - 1$	<code>%hd</code>
<code>unsigned short</code>	16 bits	de 0 à $2^{16} - 1$	<code>%hu</code>
<code>int</code>	32 bits	de -2^{31} à $2^{31} - 1$	<code>%d</code>
<code>unsigned int</code>	32 bits	de 0 à $2^{32} - 1$	<code>%u</code>
<code>long</code>	64 bits	de -2^{63} à $2^{63} - 1$	<code>%l</code>
<code>unsigned long</code>	64 bits	de 0 à $2^{64} - 1$	<code>%lu</code>

1. Écrire un programme qui a une variable `short s` qui vaut $32768 = 2^{15}$. Affichez $s + 1$. Qu'observez-vous ?
2. Changer le type de `s` en `unsigned short`. Qu'observez-vous ?
3. La suite de Fibonacci est définie ainsi : $u_0 = 1$, $u_1 = 1$ et $u_{n+2} = u_{n+1} + u_n$. Écrire un programme qui a une variable `int n` et qui calcule et affiche u_n . À partir de quelle valeur de n votre programme n'est-il plus correct ? (**Astuce** : utilisez une boucle `while`).

À rendre pour le 28 Septembre

Envoyez-moi par mail (fcapelli@math.univ-paris-diderot.fr) pour le 28 Septembre, les réponses à l'exercice suivant :

Exercice 4. Avec une boucle `while`, écrire un programme `premier.c` qui demande un entier à l'utilisateur et indique si ce nombre est premier. S'il n'est pas premier, affichez son plus petit diviseur après 1.