

Informatique S1 – Programmation C Exercices

TD 5 : Les boucles *for*

Dans ce TD, nous allons réaliser des exercices couvrant l'usage des boucles *for* et l'usage des boucles imbriquées dans le langage C.

Exercice 1

Remplir les trous dans le programme ci-dessous, sachant qu'il doit calculer x^y , avec x et y sont fournis par l'utilisateur.

```
#include <stdio.h>

int main () {
    _____ x, y; /* valeur fournis par l'utilisateur */
    float _____;
    int i;

    /* lecture des variables */
    _____ ("Entrez x : ");
    scanf ("%f", &x);
    printf ("Entrez y : ");
    _____ ("%f", &y);

    /* on multiplie x y-fois */
    for ( _____ , _____; i < y; _____ ) {
        z = z * x;
    }

    /* presentation des resultats */
    printf ("x ^ y = %f \n", z);
}
```

Exercice 2

Sachant que le factoriel d'un numéro entier n est égale à $1 * 2 * \dots * n-1 * n$, écrire un programme qui calcule le factoriel d'un numéro n donné par l'utilisateur, en utilisant pour cela la boucle *for*.

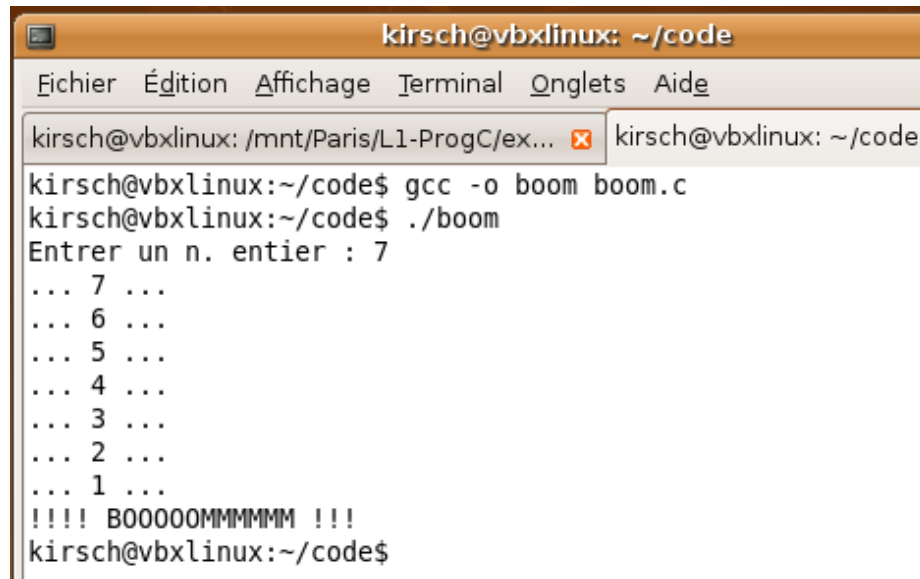
Exercice 3

Utiliser la boucle *for* pour calculer la moyenne entre 10 numéros réels fournis par l'utilisateur.

Rappel : la moyenne d'un ensemble de n numéros est la somme de ces numéros divisée par n

Exercice 4

Ecrire un programme qui réalise un compte à rebours à partir d'un nombre n fourni par l'utilisateur, tel que celui présenté dans la figure ci-dessous.



```
kirsch@vbxlinux: ~/code
Fichier Édition Affichage Terminal Onglets Aide
kirsch@vbxlinux: /mnt/Paris/L1-ProgC/ex... x kirsch@vbxlinux: ~/code
kirsch@vbxlinux:~/code$ gcc -o boom boom.c
kirsch@vbxlinux:~/code$ ./boom
Entrer un n. entier : 7
... 7 ...
... 6 ...
... 5 ...
... 4 ...
... 3 ...
... 2 ...
... 1 ...
!!!! B00000MMMMMM !!!
kirsch@vbxlinux:~/code$
```

Exercice 5

- a) Ecrire un programme qui utilise une boucle **for** pour afficher un tableau qui affiche la conversion des degrés Celsius (de -15°C à 45°C) en degrés Fahrenheit.

Rappel : $c = 5 * (f - 32) / 9$

- b) Modifier le programme précédent pour qu'il affiche les degrés de 5 en 5 (-15, -10, -5, ..., 35, 40, 45).

Celsius	Fahrenheit
-15.0	5.00
-10.0	14.00
-5.0	23.00
0.0	32.00
5.0	41.00
10.0	50.00
15.0	59.00
20.0	68.00
25.0	77.00
30.0	86.00
35.0	95.00
40.0	104.00
45.0	113.00

Exercice 6

Pour chacun des deux programmes suivants, les recopier en les indentant, faire leur tracer, et dire ce qui s'affiche à l'exécution. Expliquer la différence entre les deux programmes, et dire lequel des deux correspond le mieux aux spécifications annoncées dans les commentaires.

Code A :

```
/* Programme 1 */
/* Affichage des carres et des cubes des petits entiers */
#include<stdio.h>
main(){
int a, carre, cube;
printf("Nombre\t Carre\t Cube\n");
//Point d'observation 1
for(a=1; a<=5; a++){
carre = a*a;
cube = carre*a;
//Point d'observation 2
printf(" %i\t", a);
printf(" %i\t", carre);
printf(" %i\n", cube);
//Point d'observation 3
}
}
```

Code B :

```
/* Programme 2 */
/* Affichage des carres et des cubes des petits entiers */
#include<stdio.h>
main(){
int a, carre, cube;
printf("Nombre\t Carre\t Cube\n");
//Point d'observation 1
for(a=1; a<=5; a++)
carre = a*a;
cube = carre*a;
//Point d'observation 2
printf(" %i\t", a);
printf(" %i\t", carre);
printf(" %i\n", cube);
//Point d'observation 3
}
```

Exercice 7

a) Remplir les trous du programme ci-dessous. Que fait-il ?

```
#include <stdio.h>

int main () {
    int i, j ;
    int max = 10;

    /* affiche l'entete */
    printf ("      #");
    _____ (i=1; i<=max; i++)
        printf ("%4d", i);
}
```

```
printf ("\n");
printf ("#####");

for (____; i<max; i++)
    printf ("####");

printf ("\n");

/* affiche les differents lignes */
for (i=1; ____; i++) {
    printf ("%4d #", i);
    for (____; j<=max; ____ ) {
        printf ("%4d", i*j);
    }
    printf ("\n");
}
}
```

- b) Comment faire en sorte que le tableau affiche les mêmes calculs jusqu'au numéro 15 ?

Exercice 8

- a) Ecrire un programme qui demande 2 entiers x, y à l'utilisateur et qui affiche en retour un rectangle de x lignes et y colonnes. Par exemple, si l'utilisateur choisit x = 5, y = 3 le programme devra afficher :

```
xxx
xxx
xxx
xxx
xxx
```

- b) Modifier le programme précédent. Cette fois, un seul nombre x est demandé et le programme affiche un triangle de x lignes comme suit :

```
x
xx
xxx
xxxx
xxxxx
```

- c) Dernière modification, le programme doit afficher le triangle ci-dessous pour x = 5 :

```
  x
 xxx
xxxxx
xxxxxxx
xxxxxxxxx
```