

Algorithmique et Programmation – Tp no. 5

Exercices sur les boucles imbriquées

24 octobre 2017

Exercice 1 : Etude de code

Dans cet exercice, plusieurs bouts de programme sont proposés. Testez les sur machine, et modifiez les jusqu'à obtenir le résultat demandé.

Question 1

Testez ce programme sur machine. Explicitez son comportement via un commentaire approprié que vous mettez en tête du fichier.

```
int n;
// lecture de n
do{
    Terminal.ecrireString("Entrez_un_entier_inférieur_à_10");
    n= Terminal.lireInt();
    if (n<1 || n >9) {
        Terminal.ecrireStringln("Le_nombre_doit_etre_compris_entre_1..9._Recommencez");
    }
} while (n<1 || n >9);
// Affichages
for(int i=1; i<=n; i++){
    for (int j=1; j<=i; j++){
        System.out.print(j + "_");
    }
    System.out.println();
}
```

Question 2

Modifiez le programme de la question 1 de manière à afficher :

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

si la valeurs lue pour n est 5. Votre programme doit marcher de façon appropriée pour toute valeur lue comprise entre 1 et 9.

Question 3

Ecrivez un programme pour afficher :

```
5 5 5 5 5
 4 4 4 4
  3 3 3
   2 2
    1
```

Question 4

Ecrivez un nouveau programme permettant d'afficher, pour N=5 :

```
. . . . 5
. . . 4 .
. . 3 . .
. 2 . . .
1 . . . .
```

Exercice 2 : tables de multiplication

Dans cet exercice, vous devez écrire un programme qui commence par lire un nombre entier positif N compris entre 2 et 9, puis affiche toutes les tables de multiplication pour entre 2 et N. Par exemple, si N=4, vous devrez afficher les tables de multiplication de 2, de 3 et de 4. Votre affichage suivra le format :

```
Table de multiplication de 2
-----
 2 x 1 = 2
 2 x 2 = 4
 2 x 3 = 6
.....
```

Exercice 3 : trace d'exécution (question d'examen)

Cet exercice est apparu dans un examen récent.

Sur une feuille de papier, retracez l'exécution du programme suivant en indiquant l'évolution des valeurs des variables qu'il manipule. Vous pourrez utiliser un tableau comportant une colonne pour chaque variable, en précisant sur chaque ligne du tableau le numéro de ligne de code exécutée. Des exemples de tels tableaux sont donnés et expliqués au chapitre 3 des notes de cours :

(<http://deptinfo.cnam.fr/Enseignement/CycleA/APA/nfa031/>)

Nota bene : Vous pouvez vous limiter aux instructions qui réalisent des affectations et aux conditions de la boucle.

```
1 public class Exo2 {
2     public static void main(String[] args) {
3         int n = 42;
4         int bt = 0;
```

```

5     String r="";
6     while (n>0){
7         bt= n%2;
8         n=n/2;
9         r=bt+r;
10    }
11    Terminal.ecrireStringln("Valeur_de_r:_"+ r);
12    Terminal.ecrireStringln("Valeur_de_n:_"+ n);
13    Terminal.ecrireStringln("Valeur_de_bt:_"+ bt);
14 }
15 }

```

On rappelle que $a\%b$ correspond à l'opération *modulo*, qui calcule le reste de la division entière de a par b . Exemple : la division entière $11/3$ donne 3 en résultat (et non pas un nombre à virgule, puisqu'il s'agit d'une division entière). Le reste de $11/3$ est 2. Donc $11 \% 3 == 2$. Notez que les divisions effectuées par ce code **sont des divisions entières**.