Correction de la série de TP n°2

Exercice 1

Le programme affiche :

```
c==3 ?
Faux
c est different de 0,
 le compilateur considere la condition verifiee
```

La division de deux entiers est tromquée au plus grand entier inférieur, donc $c=2$. De plus, la condition est considérée comme vérifiée dès que l'expression (ici c) est différente de 0. Lorsque l'on remplace les entiers par des flottants, on a

```
c==3 ?
Vrai
c est different de 0,
 le compilateur considere la condition verifiee
```

Exercices 2 et 3

Basés sur l'utilisation de $\%$.

Exercice 4

```
#include <stdio.h>
 int
 main(void){
 int i, j, k = 12, l = 8; /* definition de 4 entiers */
 /* lecture des valeurs de i et de j */
 printf("\n Entrer la valeur de i :");
 scanf("%d", &i);
 printf("\n Entrer la valeur de j :");
 scanf("%d", &j);
/* ecriture du resultat selon les valeurs de i et de j
* si i est nul, impression de la valeur de k
 * si j est nul, impression de la valeur de i + l
* impression de la valeur de i +j dans les autres cas
 */
 printf("\n resultat : %d\n", (!i ? k : (!j ? i + l : i + j)));
 return 0;
 }
```

Exercice 5

```
#include <stdio.h>

int main (void){
  int i, j, k, l;
/* definition des 4 entiers */
/* affectation des entiers a 1*/
  i = j = k = 1;
/* incrementation de i de 4 manieres differentes */
  printf(" premiere solution i++ = %d \n", i++);
  printf(" deuxieme solution ++i = %d \n", ++i);
  printf(" troisieme solution i = i + 1 = %d \n", i = i + 1);
  printf(" quatrieme solution i += 1 = %d \n", i += 1);
/* affectation de i + j + 1 a k et incrementation de i et de j */
  printf(" resultat apres affectation et incrementation\n");
  k = i++ + j++ + 1; /* ou i++ + ++j ou ++i+j++ ou ++i+ ++j -1 */
  printf(" valeur de k : %d \n", k);
  printf(" valeur de i : %d \n", i);
  printf(" valeur de j : %d \n", j);
/* ecriture d une expression qui realise les operations suivantes :
 * 3 * i dans i et 3 * i + j dans j
 * et divise k par 2 logiquement et divise j par k
 * et affecte a l la puissance 2 correspondant à la valeur precedente
 */

 l = 1 << (j += (i *= 3)) / (k >>1);
  printf(" Resultat apres calcul de l expression \n");
  printf(" valeur de i ......... %d \n", i);
  printf(" valeur de j ......... %d \n", j);
  printf(" valeur de k ......... %d \n", k);
  printf(" valeur de l ......... %d \n", l);

  return 0;
}
```