

## Interrogation du 22/09/2009

*Corrigé du deuxième sujet*

### Exercice I

L'instruction `++y` incrémente `y` avant l'opération d'affectation à `x` et l'instruction `z--` décrémente `z` après l'opération d'affectation à `x`. Ainsi `y` qui valait 2 vaut maintenant 3, `z` qui valait 2 vaut maintenant 1 et `x` est égal à la somme de la nouvelle valeur de `y` et de l'ancienne valeur de `z`, c'est-à-dire 5. On a donc l'affichage suivant

1. `x=5, y=3, z=1`

L'expression `x&& y` est vraie puisque `x` et `y` sont non nuls, ainsi quelque soit la valeur de l'opérande de droite du "ou", le résultat est vrai et vaut donc 1. On a donc l'affichage suivant

2. `1`

L'affectation `x=2` retourne 2 (on remarquera qu'il est écrit `x=2` et pas `x==2`). Comme 2 correspond à vrai, on affiche

3. `x=2`

### Exercice II

La fonction suivante répond à la question

```
int min (int a, int b, int c)
/*
  Determine le minimum de trois nombres entiers
  ENTREE : les trois nombres entiers
  SORTIE : le plus petit des trois nombres
*/
{
  if (a<b) {

    /* a est plus petit que b, la finale est entre a et c */
    if (a<c) return a;
    else return c;
  }

  else {
    /* a est plus grand que b, la finale est entre b et c */
    if (b<c) return b;
    else return c;
  }
}
```