Exercice 1.8

La boucle `for` permet de former une instruction à partir de trois instructions et d'une expression booléenne. Cette instruction s'écrit `for(`p_1 `; b ;` p_2`)` p_3. C'est une abréviation pour l'instruction p_1 `; while (b) {`p_3 p_2 `;}`. Que fait l'instruction suivante ?

```
{y = 1; for(x = 1; x <= 10; x = x + 1) y = y * x;}
```

Exercice 1.9

Donner la définition de la fonction Σ pour la déclaration d'une variable sans valeur initiale.

Exercice 1.10

Soit `e` l'environnement — impossible à construire en Java — `[x = r, y = r]`, `m` la mémoire `[r = 4]`, `p` l'instruction `x = x + 1 ;` et `m'` la mémoire Σ`(p,e,m)`. Quelle est la valeur associée à `y` dans l'état `e, m'` ? Même question pour l'environnement `[x =` r_1`, y =` r_2`]` et la mémoire `[`r_1 `= 4,` r_2 `= 4]`.

Dessiner ces deux états.

Exercice 1.11

On suppose que toutes les mémoires contiennent une référence spéciale : `out`. Écrire la définition de la fonction Σ pour la construction de sortie `System.out.print` de la section 1.2.

Exercice 1.12

Dans cet exercice, on considère un type infini fictif pour les entiers.

À chaque instruction `p` du noyau impératif de Java, on associe la fonction partielle des entiers dans les entiers qui, à l'entier `n`, associe la valeur associée à `out` dans la mémoire Σ`(p,[x = in, y = out],[in = n, out = 0])`.

Une fonction partielle `f` des entiers dans les entiers est dite *calculable* s'il existe une instruction `p` telle que `f` soit la fonction ainsi associée à `p`.

Montrer qu'il existe une fonction non calculable.

Indication : utiliser le fait qu'il n'existe pas de surjection de $\mathbb{N}$ dans l'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{N}$.