Correction du DS 2

Informatique pour tous, première année

Julien Reichert

Exercice 1

1. Il est tout à fait possible d'accéder à un élément d'une séquence fournie explicitement, même si on trouve plus souvent des accès à un élément d'une séquence en tant que variable. C'est simplement dû au fait que les constantes sont rares en pratique. L'indice `-3` est le troisième en partant de la fin, mais le dernier ne peut évidemment pas être `-0`, donc la réponse est le caractère `'7'`.

2. Les entiers n'ont pas de longueur, la réponse `a)` est donc fausse. Les chaînes de caractères en ont une, la réponse `b)` est donc vraie. Quant aux listes, leur longueur est définie comme le nombre d'éléments sans considérer leur valeur, donc la liste `c)` est de longueur un, alors que la liste `d)` est de longueur sept. De même pour la liste `e)` qui est en fait `['a','b','c','d','e','f','g']`. Enfin, un objet `range` a certes une longueur, mais les éléments de `range(1,7)` sont les nombres de `1` à `6`.

3. La liste `l1` est composée de quatre éléments : les listes `[1]`, `[2,2]`, `[3,3,3]` et `[4,4,4,4]`. Ceci est dû à l'utilisation de la méthode `append` qui n'ajoute qu'un élément en fin de liste, quel que soit son type (en l'occurrence, ici, une liste). Au contraire, la liste `l2` est effectivement « plate » et de longueur dix. La liste `l3` est composée de dix listes (type des éléments ajoutés par `append`) toutes égales à `[42]`. En effet, quand `i3` est pair, cette liste est ajoutée deux fois, et quand `i3` est impair, elle est ajoutée trois fois.

La chaîne `s1` est composée des caractères de l'extrait de la Déclaration des droits de l'homme et du citoyen de 1789 aux positions de `2` (incluse) à `22` (exclue) par pas de `2`. Il y en a donc effectivement dix. La chaîne `s2` est composée des caractères de `s1` du début au dernier exclu, en ajoutant au début un `"A"`. Il y en a encore dix. La chaîne `s3` est composée des caractères de `s2` entre la position `1` (incluse) et la fin. Le premier ayant été écarté, il n'en reste que neuf.

Exercice 2

Voir la correction de l'exercice 4 du DS précédent...

Exercice 3

Commençons par annoncer la spécification : la fonction correspond à $f : \mathbb{N} \setminus \{0\} \to \mathbb{N}$ telle que $f(n) = k$ vérifiant $k! \leq n < (k+1)!$.

La terminaison de la boucle, donc de la fonction, se prouve en exhibant un variant. On prend simplement `n - tot`, qui est clairement un entier naturel au début, et qui est strictement décroissant, avec sortie de la boucle dès qu'il devient strictement négatif, puisqu'il s'agit de la différence d'une valeur fixe et d'un entier valant initialement `1` et multiplié à chaque tour de boucle par un entier valant initialement `2` et augmentant d'un également à chaque tour.

En pratique, s'il s'agissait de déterminer la complexité, il faudrait répondre quelque chose du genre « une réciproque de la factorielle » (en s'appuyant sur la fonction Γ qui étend la factorielle à $\mathbb{R}$, par exemple).

Prouvons la correction en choisissant un invariant de boucle pertinent. Il suffit de dire qu'à la fin du $k-1$-ième tour de boucle (en étendant ceci au 0-ième tour signifiant avant l'entrée dans la boucle), `tot` est la factorielle de `rep`, celui-ci valant k. C'est vrai avant d'entrer dans la boucle car la factorielle de un est un, et on le prouve par récurrence : considérons qu'après le $k-1$-ième tour `tot` est la factorielle de `rep` qui vaut k.

Après le k-ième tour de boucle, `rep` vaut un de plus, soit $k+1$, et `tot` a été multiplié par la nouvelle valeur de `rep`, il vaut donc $(k+1)!$.

Finalement, à la sortie de la boucle, `tot` vaut encore la factorielle de `rep` (parler de k n'est plus utile maintenant, cela a seulement facilité la récurrence), mais on sait aussi que `tot` est strictement supérieur à `n` alors que ce n'était pas le cas au tour précédent.

Ainsi, on peut dire que la factorielle de `rep-1` est inférieure ou égale à `n` qui est strictement inférieur à la factorielle de `rep`. On conclut en soulignant que c'est `rep-1` qui est retourné.