

T D1 : Alg Dem

Exercice 1 : Multiplication.

Voici une implémentation naïve de la multiplication d'entiers.

```
mult :: Int -> Int -> Int
mult 0 m = 0
mult n m = m + mult (n-1) m
```

Q 1.1 Exécutez pas à pas `mult 3 5`.

Q 1.2 Donnez deux implémentations de `mult` en Python :

1. une implémentation récursive,
2. une implémentation itérative.

Q 1.3 Voici une spécification pour `mult x y` :

— **pré-condition** : $x \geq 0$

— **post-condition** : $\text{mult } x \ y = x * y$

Démontrez par induction sur x que `mult x y` vérifie bien cette spécification.

$$\begin{aligned} \text{Q 1.1: } \text{mult } 3 \ 5 &= 5 + \text{mult } 2 \ 5 \\ &= 5 + 5 + \text{mult } 1 \ 5 \\ &= 5 + 5 + 5 + \text{mult } 0 \ 5 \\ &= 5 + 5 + 5 + 0 = 15 \text{ OK.} \end{aligned}$$

Q 1.2

```
def mult(a, b):
```

```
    if b == 0:
```

```
        return 0
```

```
    else:
```

```
        return a + mult(a-1, b)
```

```
def mult(a, b):
```

```
    res = 0
```

```
    while b > 0:
```

```
        res += a
```

```
        b -= 1
```

```
    return res.
```

Q 1.3

On va montrer que $x \geq 0 \Rightarrow \text{mult } x y = x * y$.
par induit sur x .

Notre hypothèse d'induction sera :

$$\forall x'. x' < x, \text{mult } x' y = x' * y$$

On a deux cas :

Cas 1 : $x = 0$.

On a alors $\text{mult } 0 y = 0 = 0 * y$ OK.

Cas 2 : $x > 0$.

On a $\text{mult } x y = y + \text{mult } (x-1) y$
 $+ x > 0 \Rightarrow x-1 > -1 \Rightarrow x-1 \geq 0$ OK.

Or $x-1 < x$, on peut donc appliquer IH.

$$\text{mult } x y = y + x-1 * y = y(1 + x-1) = x * y$$

Tous les cas ont été montrés. En vertu du principe d'induction,
on a alors $\text{mult } x y = x * y \forall x \forall y$.

Exercice 2 : Exponentiation rapide.

Voici une implémentation de l'exponentiation de deux entiers :

```
exp :: Int -> Int -> Int
exp a 0 = 1
exp a b = a * exp a (b-1)
```

Q 2.1 Calculez pas à pas `exp 3 3`.

Q 2.2 Donnez deux implémentations `exp` en Python :

1. une version récursive,
2. une version itérative.

Q2.1

$$\begin{aligned} \text{exp } 3 \ 3 &= 3 * \text{exp } 3 \ 2 \\ &= 3 * 3 * \text{exp } 3 \ 1 \\ &= 3 * 3 * 3 * \text{exp } 3 \ 0 \\ &= 3 * 3 * 3 * 1 = 27 = 3^3 \text{ ok.} \end{aligned}$$

Q2.2

def exp(a, b):

if a==0:

return 1

else:

return a * exp(a, b-1)

def exp(a, b)

res = 1

for i in range(b):

res *= a

return res.

Q 2.3 Montrez que pour tout $b_1 \geq 0$ et $b_2 \geq 0$ que $\text{exp } a \ (b_1 + b_2) = \text{exp } a \ b_1 * \text{exp } a \ b_2$. Pour cela, vous procéderez par induction sur b_1 .

On veut montrer que :

$$(b_1 \geq 0 \wedge b_2 \geq 0) \Rightarrow \text{exp } a \ (b_1 + b_2) = \text{exp } a \ b_1 * \text{exp } a \ b_2$$

On procède par induction sur b_1 .

cas 1, $b_1 = 0$

$$\text{on a donc } \text{exp } a \ (0 + b_2) = \text{exp } a \ b_2$$

$$= 1 * \text{exp } a \ b_2 = \text{exp } a \ 0 * \text{exp } a \ b_2 \quad \underline{\text{ok}}$$

cas 2 : On suppose que $\forall b_1'. b_1' < b_1$, mult $a \ (b_1' + b_2)$

$$= \text{exp } a \ b_1' * \text{exp } a \ b_2$$

$$\text{On a } \text{exp } a \ (b_1 + b_2) = a * \text{exp } a \ (b_1 + b_2 - 1)$$

$$= a * \text{exp } a \ ((b_1 - 1) + b_2)$$

Ok, $b_{l-1} < b_l$, donc on peut utiliser IH.

et $b_l > 0 \Rightarrow b_{l-1} > -1 \Rightarrow b_{l-1} \geq 0$ ok.

$$a * \exp a ((b_{l-1}) + b_l) = a * \exp a (b_{l-1}) * \exp a b_l$$

$$\begin{aligned} \text{Donc : } \exp a (b_l + b_{l-1}) &= a * \exp a (b_{l-1}) * \exp a b_l \\ &= \exp a b_l * \exp a b_{l-1} \\ &\text{(par def de exp) ok.} \end{aligned}$$

CC.L.

Nous allons maintenant étudier l'algorithme d'exponentiation rapide dont voici une implémentation.

```
expR :: Int -> Int -> Int
expR a 0 = 1
expR a b | b `mod` 2 == 0 = e * e
          | otherwise     = a * e * e
  where
    e = expR a (b `quot` 2)
```

La fonction quot calcule la division euclidienne et $b \text{ `quot` } 2$ est le résultat de la division euclidienne de b par 2. Notez bien que pour tout a et b , si $b > 0$, alors $a = 2 * (a \text{ `quot` } b) + a \text{ `mod` } b$.

Q 2.4 Calculez $\text{expR } 3 \ 3$.

$$\begin{aligned} \text{expR } 3 \ 3 &= 3 * \text{expR } 3 \ 1 * \text{expR } 3 \ 1 \\ &= 3 * 3 * \text{expR } 3 \ 0 * 3 * \text{expR } 3 \ 0 \\ &= 3 * 3 * 1 * 3 * 1 = 27 = \exp 3 \ 3 \quad \underline{\text{ok}} \end{aligned}$$

Q 2.6 La fonction $\text{expR } a \ b$ implémente l'algorithme d'exponentiation rapide. Voici ses pré et post-condition :

— Pré-condition : $b \geq 0$

— Post-condition : $\text{expR } a \ b = \exp a \ b$.

Démontrez par induction sur b que expR vérifie sa spécification.

$$\forall b'. b' \geq 0 \wedge b' < b \Rightarrow \text{expR } a \ b' = \exp a \ b'$$

cas 1 : $b = 0$

$$\text{expR } a \ b = 1 = \exp a \ 0 = \exp a \ b. \quad \underline{\text{ok}}$$

Cas 2 : $b > 0$

Cas 21 : $b \equiv 0 \pmod{2}$

$$\text{expR } a \ b = \text{expR } a \ (b \text{ `quot` } 2)^2$$

$$\text{On a } \frac{b}{2} < b \Rightarrow b < 2b \text{ car } 2 > 0.$$

$$\text{On a } b/2 > 0 \text{ car } b > 0.$$

IH :

$$\begin{aligned} \text{expR } a \ b &= \text{exp } a \ (b \text{ `quot` } 2) * \text{exp } a \ (b \text{ `quot` } 2) \\ &= \text{exp } a \ b \text{ (en vertu de ce qu'on a montré avant)} \end{aligned}$$

Cas 22 $b \equiv 1 \pmod{2}$.

$$\begin{aligned} \text{expR } a \ b &= a * \text{expR } a \ (b \text{ `quot` } 2)^2 \quad \text{IH} \\ &= a * \text{exp } a \ (b \text{ `quot` } 2) * \text{exp } a \ (b \text{ `quot` } 2) \\ &= a * \text{exp } a \left(\frac{2k+1}{2} + \frac{2k+1}{2} \right) \\ &= a * \text{exp } a \ (2k+1) \\ &= \text{exp } a \ (b+1) \quad ? \text{ (trac qui cloche)} \end{aligned}$$

Exercice 3 : Plus grand diviseur premier

La fonction `pgfp :: Int -> Int` est censée renvoyer le plus grand facteur premier de n lorsque celui-ci est un entier plus grand que 2.

```
pgfpAux :: Int -> Int -> Int
pgfpAux i n | i == n      = i
             | n `mod` i == 0 = pgfpAux i (n `quot` i)
             | otherwise    = pgfpAux (i+1) n
```

```
pgfp :: Int -> Int
pgfp n = pgfpAux 2 n
```

Q 3.1 Effectuer le calcul pas à pas de `pgfpAux 2 2520`. Pour quelle valeurs de i effectue-t-on un appel récursif de la forme `pgfpAux i (n `quot` i)`? Trouver une propriété commune à ces valeurs.

Q 3.1 Effectuer le calcul pas à pas de $\text{pgfpAux } 2 \ 2520$. Pour quelle valeurs de i effectue-t-on un appel récursif de la forme $\text{pgfpAux } i \ (n \text{ 'quot' } i)$? Trouver une propriété commune à ces valeurs.

$$\begin{aligned}
 \text{pgfpAux } 2 \ 2520 &= \text{pgfp } 2 \ 1260 \\
 &= \text{pgfp } 2 \ 630 \\
 &= \text{pgfp } 2 \ 315 \\
 &= \text{pgfp } 3 \ 315 \\
 &= \text{pgfp } 3 \ 105 \\
 &= \text{pgfp } 3 \ 35 \\
 &= \text{pgfp } 4 \ 35 \\
 &= \text{pgfp } 5 \ 35 \\
 &= \text{pgfp } 5 \ 7 \\
 &= \text{pgfp } 6 \ 7 \\
 &= \text{pgfp } 7 \ 7 = 7.
 \end{aligned}$$

On fait un appel récursif lorsque:

$$\exists k \in \mathbb{N}. k > 1, n = i k; \quad i \text{ quot } i k = k \text{ donc } \equiv 0 \pmod{n}$$

Q 3.2 Quelle propriété lie n et les entiers k compris entre 2 et $i - 1$ lors des appels (récursifs) à $\text{pgfpAux } i \ n$?

$$\forall k. k \in \llbracket 2, i-1 \rrbracket \quad n > k \Rightarrow \text{appel récursif de } \text{pgfpAux}.$$

Q 3.3 À partir de la question précédente, donner la pré-condition de $\text{pgfpAux } i \ n$ qui lui permette de renvoyer le plus grand diviseur premier de n

$$n \geq i$$

Q 3.4 Indiquer le variant de $\text{pgfpAux } i \ n$ et justifier qu'il est bien positif.

Le variant est n , et on a par définition de n
 $n \geq 2$. ok.

Q 3.5 Démontrer que pgfpAux vérifie sa spécification. C'est-à-dire que pour la pré-condition que vous avez donnée précédemment, $\text{pgfpAux } i \ n$ renvoie le plus grand diviseur premier de n . Pour cela vous utilisez une induction sur le variant que vous avez donné précédemment.

On veut montrer par induction sur n que:

$$n \geq i \Rightarrow \text{pf}$$

$\forall n' < n$, $\text{pgfp Aux } i \text{ } n'$ calcule le pgfp de $n' \geq i$.

Cas 1: $n = i$

On a alors $\text{pgfp Aux } i \text{ } n = i$ ok

car le plus grand diviseur premier de $n \geq n$ est n .

Cas 2: $n > i$

Cas 2.1: $n \equiv 0 \pmod i$

On a alors:

$$\text{pgfp Aux } i \text{ } n = \text{pgfp Aux } i \text{ } (n \text{ `quot` } i)$$

Comme $i \geq 2$ (par def), on a $\frac{n}{i} < n$.

On a aussi $n > i \Rightarrow \frac{n}{i} \geq 2$ car $i \geq 2$.

On peut appliquer IH

pf $\text{pgfp Aux } i \text{ } n$ calcule le pgfp de $\frac{n}{i}$ supérieur ou égal à i