

TP2 MPSI : Boucles et références

fabien.viger@ens.fr

19 janvier 2005

But du TP

Dans ce TP, on va apprendre à se servir des boucles `for`, `while`, et des références. On va aussi utiliser l'interactivité de `Cam1`. Les questions **bonus** sont facultatives, ne les abordez que si vous pensez avoir le temps de finir le TP. Et bien sûr, **à n'importe quel moment, toute question est la bienvenue.**

1 Boucles `for`, références

Question 1 : Factorielle

Faire une fonction `fact` qui calcule $n!$

La tester sur quelques valeurs.

Pour quelle valeur `Cam1` dépasse-t-il sa capacité? (quand il commence à donner des résultats fantaisistes)

Question 2 : Somme, Produit

Écrire deux fonctions `somme` et `produit` qui prennent en argument une fonction f de type `int -> int` et un entier n , et qui calculent respectivement $\sum_{i=1}^n f(i)$ et $\prod_{i=1}^n f(i)$

Quelle fonction `g` faut-il donner en argument à `produit` pour que `produit g` soit équivalente à `fact`?

Question 3 : Fibonacci

La suite de Fibonacci est définie par $u_0 = u_1 = 1$ et la relation de récurrence suivante :

$$\forall n \in \mathbb{N}, \quad u_{n+2} = u_{n+1} + u_n$$

Écrire une fonction `fibonacci` qui calcule le n -ième terme de la suite de Fibonacci.

Bonus Sachant que le nombre d'or, $\Phi = \frac{1+\sqrt{5}}{2}$ est égal à $\lim_{n \rightarrow \infty} \frac{u_{n+1}}{u_n}$, s'inspirer de `fibonacci` pour écrire un programme qui va calculer une approximation du nombre d'or à des rangs sans cesse croissants, jusqu'à ce que cette

approximation ne change plus. (Pour tester l'“égalité” de deux flottants, utiliser par exemple le test `abs_float(x -. y) < 1e-8`). Comparer la valeur ainsi obtenue avec

```
(1. +. sqrt 5. /. 2.);;
```

Question 4 : Itérer...

Écrire une fonction `iter` qui prend en argument une fonction f de type `int -> 'a -> 'a`, un entier n et une valeur de départ x_0 de type `'a`, telle que le résultat renvoyé par

```
iter f n x0;;
```

soit égal à

```
f n (f (n-1) (f (n-2) (... (f 1 x0))))
```

Noter que `prefix` est un mot-clé de Caml permettant de currier un opérateur : essayer par exemple

```
let plus = prefix + ;;
```

```
plus 3 6;;
```

Utiliser `prefix *` avec la fonction `iter` pour imiter la fonction `fact`.

2 Interactivité

Question 5 : Entrée...

Regarder le type des fonctions `read_int`, `read_float` et `read_line`, et essayer de les exécuter, en tapant par exemple :

```
read_line();;
```

puis en tapant n'importe quoi.

Question 6 : ... et Sortie

De même, regarder les fonctions `print_char`, `print_int`, `print_float`, `print_string`, `print_newline` et `print_endline`.

Regarder et comprendre ce que fait le programme suivant :

```
let question () =  
  print_endline "Quel est ton nom ?";  
  let nom = read_line() in  
  print_endline "et ton age ?";  
  let age = read_int() in  
  print_string ("Tu t'appelles " ^ nom ^ " et tu as ");  
  print_int age;  
  print_endline " ans.";;
```

```
question();;
```

Question 7 : Un premier programme interactif

On va faire un petit jeu de black-jack (*très simplifié*), où on joue seul contre l'ordinateur. La règle est simple : on a une cagnotte de départ fixée. L'ordinateur nous tire une carte (un nombre au hasard entre 1 et 11), et à ce moment là on mise une certaine somme. Ensuite, l'ordinateur tire une seconde carte, et nous demande si l'on veut en tirer une autre, auquel cas il en tire une troisième, et nous redemande si on veut continuer, etc.. Si le total des cartes dépasse 21, on perd la somme mise, mais si il atteint un nombre entre 17 et 21, on récupère le double de la somme mise.

3 Boucles while, programmes élaborés

Question 8 : Résolution de $f(x)=0$

Étant donné une fonction continue $f : \mathbb{R} \rightarrow \mathbb{R}$, et deux réels a et b tels que $f(a) < 0$ et $f(b) > 0$, écrire un programme `dichotomie` qui trouve une racine de l'équation $f(x) = 0$. On utilisera la méthode dichotomique, qui calcule $f(c)$, où $c = \frac{a+b}{2}$, et qui remplace a ou b par c avant de recommencer. Essayer sur la fonction suivante :

```
let yo x = x *. x +. x -. 1. ;;
dichotomie yo 0. 2. ;;
```

Question 9 : Factorisation

À l'aide de la fonction `mod`, faire un programme qui décompose un entier en facteurs premiers. L'optimiser autant que possible, et le tester sur 100, 1001, $10^8 + 1$ puis $10^9 + 7$. Le programme est censé tourner en un temps raisonnable, même sur ces derniers exemples!

4 Enregistrements, nombres complexes

Question 10 : Enregistrements

On peut définir de nouveaux types en caml, notamment des juxtapositions de plusieurs types déjà connus. Essayer par exemple :

```
type fiche = { nom:string; taille:int; poids:float };;
```

```
let ma_fiche = { nom="moi"; taille=171; poids=62.5 };;
ma_fiche.taille;;
```

Faire une fonction qui prend une fiche en argument et écrit un petit message du style "Bonjour untel, votre indice de masse corporel est de xxx". (l'indice de masse corporel est égal à $500 * \text{poids}/\text{taille}^2$)

Question 11 : Nombres complexes

Définir un type `complexe` pour coder un nombre complexe. Écrire les deux fonctions de base `add` et `mul` effectuant la somme et la multiplication de deux nombres complexes.

Bonus Définir aussi `sub`, `div`, `conj`

Question 12 : Exponentiation rapide

Écrire une fonction simple `puiss` calculant la puissance n-ième d'un nombre complexe.

À présent, optimiser cette fonction, en remarquant que si $n = 19$ par exemple, comme 19 s'écrit 10011 en binaire, alors

$$x^{19} = x^{10011_b} = x * x^2 * x^{16} = x * x^2 * \left(\left((x^2)^2 \right)^2 \right)^2$$

Une fois que la fonction optimisée `expo` est codée, comparer les temps d'exécution :

```
let i = {re=0.0; im=1.0};;  
puiss i 1000003;;  
expo i 1000000003;;
```