Université Paris 6
Licence d'informatique

Module de Programmation
ANNÉE 2000-2001

Travaux Pratiques n° 0
FAMILIARISATION AVEC CAML

1 Les modes de compilation d'un programme CAML

Il existe plusieurs façons de compiler un programme CAML. Tout d'abord, il est possible d'écrire un programme CAML dans un fichier et de le compiler pour obtenir un fichier exécutable. Ce fichier doit obligatoirement avoir l'extension `.ml`. La compilation du programme s'effectue en exécutant depuis un shell la commande :

`camlc` *`fichier`*`.ml -o` *`fichier_executable`*

1.1 Le mode interactif

CAML est également fourni avec un mode interactif (dit boucle d'interaction), qui est activé depuis un shell par la commande `camllight`. Le prompt `#` vous invite à saisir les déclarations et expressions CAML. Les phrases CAML sont analysées, compilées et exécutées. À l'issue de l'exécution, le prompt réapparait, vous invitant à une nouvelle saisie.

L'éditeur du mode interactif est des plus rudimentaires : il ne vous permet de corriger que la ligne que vous êtes en train de taper en supprimant tous les caractères jusqu'au point de modification souhaitée. On utilise donc la plupart du temps un éditeur de texte pour taper ses programmes dès que leur taille dépasse la ligne.

1.2 L'éditeur de texte `emacs` et son mode CAML

`emacs` est un éditeur de texte aux possibilités très étendues. On dispose d'une interface permettant d'éditer des programmes CAML. L'éditeur se lance soit au moyen d'un des menus de votre interface graphique, soit par la commande `emacs` exécutée depuis un shell.

Pour faire fonctionner le mode Caml sous `emacs`, vous devez exécuter la première fois la commande Unix suivante :

```
cp /Infos/caml-ens/99-00/tuareg-mode/2.emacs ~/.emacs
```

qui vous crée un fichier de configuration `~/.emacs` pour `emacs`.

À partir de cet instant, si vous ouvrez un fichier dont le nom se termine par l'extension `.ml`, alors vous êtes automatiquement dans ce mode, le texte que vous tapez se colorie au fur et à mesure que des mots-clefs du langage sont reconnus, vous disposez d'un menu `Tuareg` qui vous propose d'évaluer la phrase courante, tout le contenu du buffer ou bien d'ouvrir un deuxième buffer où Caml est lancé interactivement.

2 Exercices

Exercice 1 *(Fonctions réelles ou entières)*

1 Écrire une fonction prenant un entier x et calculant la valeur de $2x^2 + 3x - 2$.

2 Écrire une fonction prenant un réel x et calculant la valeur de $2x^2 + 3x - 2$.

3 Écrire une fonction affichant les racines réelles (si elles existent) de l'équation $ax^2 + bx + c = 0$ (a, b et c sont entiers).

Exercice 2 *(Premières fonctions récursives)*

1 Écrire une fonction calculant la factorielle d'un entier.

2 Écrire une fonction donnant le n-ième terme de la suite de Fibonacci. On rappelle que cette suite est définie par récurrence de la façon suivante :

$$\begin{cases} F_0 = F_1 = 1 \\ F_{n+2} = F_n + F_{n+1}, \text{ pour tout } n \in \mathbb{N} \end{cases}$$

Exercice 3 *(Les fonctions à plusieurs arguments)*

1 Écrire la fonction entière `power2` qui associe à un entier n sa puissance de 2, 2^n.

2 Écrire la fonction entière `power` qui prend deux entiers x et y et leur associe x^y.

3 Réécrire la fonction `power` pour qu'elle prenne des arguments de type `float`.

Exercice 4 *(Les listes simples, filtrage et itérateurs)*

1 Écrire une fonction retournant le deuxième élément d'une liste.

2 Écrire une fonction sommant les valeurs contenues dans une liste d'entiers.

3 Écrire une fonction affichant tous les éléments d'une liste d'entiers.

4 Écrire une fonction incrémentant d'une valeur entière, passée en paramètre, tous les éléments d'une liste d'entiers.

5 Écrire une fonction incrémentant d'une valeur entière, passée en paramètre, toutes les premières composantes d'une liste de couples d'entiers.

6 Écrire une fonction qui prend en argument une liste de couples et retourne la liste de leur première composante.

7 Écrire une fonction incrémentant d'une valeur entière, passée en paramètre, tous les premiers éléments d'une liste de listes d'entiers. Que se passe-t-il lorsque l'on passe une liste d'entiers en argument ? Et une liste de listes de listes d'entiers ?

Exercice 5 *(Traits impératifs : la séquence, les boucles, l'affectation)*

1 Écrire une fonction permutant le contenu de deux variables entières passées en paramètre.

2 Écrire une fonction affichant les nombres inférieurs à 1000 et dont la valeur est égale à la somme du cube du chiffre des centaines, du cube du chiffre des dizaines et du cube du chiffre des unités.

Exercice 6 *(Représentation et manipulation de matrices)*

1 Écrire une fonction calculant le produit de deux vecteurs de taille n.

2 Écrire une fonction calculant le produit de deux matrices de taille $n * n$.

Exercice 7 *(Manipulation de tableaux)*

1 Écrire une fonction permutant deux éléments d'un tableau `T`, d'indices respectifs `i` et `j`.

2 Écrire une fonction de tri des éléments d'un tableau basé sur l'algorithme de tri par sélection-échange. Le tri par sélection-échange procède en deux étapes pour placer chaque élément à sa place définitive :

- la recherche de l'élément à placer à l'endroit courant parmi l'ensemble des éléments non encore placés
- la permutation de cet élément avec l'élément placé actuellement à l'endroit courant. La fonction de tri doit pouvoir s'appliquer aux tableaux d'entiers ou de chaînes de caractères, et l'on doit pouvoir spécifier lors de l'appel si le tri s'effectue par ordre croissant ou décroissant.

3 Peut-on utiliser la fonction de tri que vous avez définie précédemment pour trier un tableau de listes selon la longueur des listes ? Si ce n'est pas le cas, modifiez vos fonctions réalisant le tri afin de pouvoir trier les éléments du tableau selon n'importe quel critère (admettant un ordre) associé à chaque élément du tableau (par exemple la longueur de la liste, sa profondeur, le nombre d'occurrences du caractère 'i', le nombre de 0 dans la représentation binaire associée ...)

Exercice 8 *(Les fonctions polymorphes)*

1 Définir la fonction `identité`, qui à une valeur associe elle-même.

2 Écrire une fonction `applique` prenant deux arguments, et appliquant le premier sur le second. Commentez le type de cette fonction, et trouvez un contre-exemple mal typé de son utilisation.

3 La fonction prédéfinie `print_int : int -> int` prend un argument entier, l'affiche à l'écran et retourne la valeur `()`. Définissez la fonction `id_print` qui prend un entier et le rend, en l'ayant préalablement affiché à l'écran.