

TP1 - 1 séance

Introduction au langage Caml Fonctions numériques

Vous pouvez trouver à cette adresse https://doc.ubuntu-fr.org/caml_light comment utiliser gedit ou emacs pour faire des programmes Caml Light.

Algorithmes de fonctions simples :

a) Fonctions à une variable :

- Écrire une fonction de calcul d'un prix TTC à partir d'un prix HT avec un taux de TVA de 20 %.
- Écrire une fonction qui détermine si une année est bissextile
- Écrire une fonction qui teste si un caractère est une lettre minuscule

b) Fonctions à deux variables :

- Écrire une fonction qui calcule la moyenne de 2 réels.
- Écrire une fonction qui détermine le quotient et le reste d'une division entière.

c) Fonctions avec déclarations locales :

- Définir une fonction puissance_4 pour les entiers utilisant une fonction carré locale.
- Écrire une fonction qui convertit une lettre minuscule en majuscule.

Algorithmes de fonctions récursives

a) Écrire une fonction qui calcule le n-ème terme de la suite de Fibonacci est définie par :

$$u_0=0; \quad u_1=1; \quad \text{et} \quad \forall n>1, \quad u_n=u_{n-1}+u_{n-2}.$$

b) Écrire une fonction calculant la somme des n premiers carrés.

Algorithmes de fonctions d'ordre supérieur

a) Écrire une fonction **sigma** qui calcule $\sum_{x=0}^n f(x)$. Puis utiliser cette fonction pour redéfinir la somme des n premiers carrés.

b) Écrire une fonction **rond** telle que $\text{rond } f \circ g = g \circ f$ (où $g \circ f(x) = g(f(x))$). Utiliser ensuite cette fonction en l'appliquant à différents arguments.

Calcul de la racine carrée par la méthode de Newton

La suite y_n définie par
$$\begin{cases} y_0 = x \\ y_{n+1} = \frac{(y_n + x/y_n)}{2} \end{cases}$$
 converge vers \sqrt{x} .

L'idée est donc de partir de y_0 et de calculer y_{n+1} , jusqu'à ce que $(y_{n+1})^2 = x \pm \epsilon$.

- Écrire la fonction (Newton x y eps) qui calcule les termes successifs de la suite de Newton, jusqu'à ce que $y^2 = x \pm \epsilon$. Utiliser cette fonction pour définir la fonction Racine.
- Redéfinir la fonction Racine, en utilisant deux fonctions locales, la première testant si la solution approchée est correcte ou non, la seconde calculant y_{n+1} à partir de y_n