Travaux Pratiques
de Programmation Fonctionnelle avec Racket

S. PIECHOWIAK

L'interpréteur utilisé dans les TP est DrRacket.

Consignes ...

Travail demandé : 2 fichiers doivent être envoyés dans le même message.

Le premier fichier est le compte rendu de TP sous la forme d'un seul document au format PDF. On y expliquera les parties délicates des programmes. On présentera des exemples de résultats.

Le nom du fichier est impérativement de la forme :

<NOM DES ETUDIANTS DU GROUPE>-<TP-SCHEME-2019-2020>.pdf

Le 2[ème] fichier contient tous les sources des programmes documentés. Le nom de ce fichier est impérativement de la forme :

<NOM DES ETUDIANTS DU GROUPE>-<TP-SCHEME-2019-2020>.txt

Les 2 documents doivent être envoyés ensembles à l'adresse :

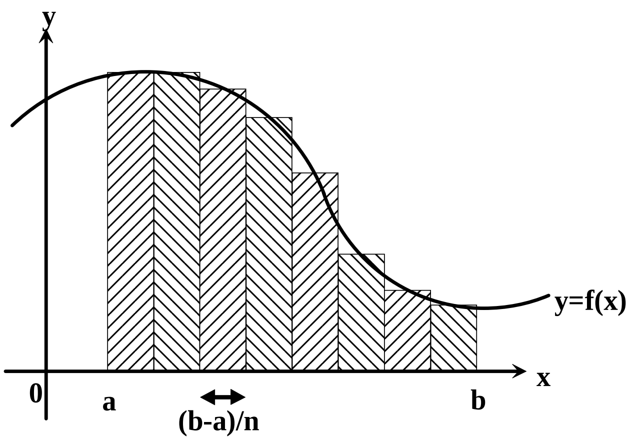
sylvain.piechowiak@univ-valenciennes.fr avant le 22/11/2019 minuit.
Après cette date, les documents seront automatiquement mis à la corbeille (note = 00/20).

PARTIE 1 : Calculs d'intégrales ...

On souhaite calculer une valeur approchée de l'intégrale d'une fonction réelle f supposée intégrable sur un intervalle [a,b] en calculant l'aire algébrique de la surface limitée par l'axe des abscisses, la courbe qui représente f et les droites y = a et y = b.

On découpe l'intervalle [a,b] en n segments de même longueur $\frac{b-a}{n}$ et on calcule la somme des surface des rectangles de largeur $\frac{b-a}{n}$ et de hauteur la valeur de f pour le point du milieu de chaque segment.

L'abscisse du milieu du kième segment vaut $m_k = a + (k-1)\frac{b-a}{n} + \frac{1}{2}\frac{b-a}{n}$. L'aire du kième rectangle A_k vaut donc $A_k = \frac{b-a}{n} \times f(m_k)$ et finalement pour un nombre n de rectangles la valeur approchée $\int_a^b f$ vaut donc : $\sum_{k=1}^{n} A_k = \frac{b-a}{n} \times \sum_{k=A}^{n} f(m_k)$



Définir une fonction **`integrale`** qui pour le quadruplet (f n a b) renvoie la valeur de l'intégrale $\int_a^b f$ obtenue par la méthode précédente avec n donné. f étant la fonction à intégrer, a et b étant les bornes de l'intégrale et n étant le pas de découpage. On supposera que la fonction f est continue sur [a,b].

PARTIE 2 : Calculs matriciels …

On souhaite faire du calcul matriciel numérique.

Par exemple, voici une matrice carrée A de dimension 3 × 3 :

$$A = \begin{bmatrix} 1.2 & 6.0 & -1.3 \\ 2.5 & -1.1 & 2.9 \\ 3.8 & -4.0 & -2.7 \end{bmatrix}$$

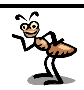
On se propose de représenter les matrices par des listes. Ainsi la matrice $A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,1} & a_{3,3} \end{bmatrix}$ est représentée par la liste LA = ($(a_{1,1}, a_{1,2}, a_{1,3})$,($a_{2,1}, a_{2,2}, a_{2,3}$), ($a_{3,1}, a_{3,1}, a_{3,3}$)) dans laquelle chaque sous-liste représente une ligne de la matrice.

1. La transposée de la matrice $A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,1} & a_{3,3} \end{bmatrix}$ est la matrice $A^T = \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} \\ a_{1,2} & a_{2,2} & a_{3,1} \\ a_{1,3} & a_{2,3} & a_{3,3} \end{bmatrix}$ dans laquelle $a_{i,j}^T = a_{j,i}$.

   Définir la fonction **`transposee`** qui renvoie la transposée de la matrice passée en argument.

2. Une matrice est dite symétrique si $\forall i \forall j \; a_{i,j} = a_{j,i}$. Définir la fonction **`estSymetrique`** qui renvoie **`#t`** si la matrice passée en argument est symétrique et renvoie **`#f`** sinon.
3. On définit la somme de 2 matrices de mêmes dimensions par : soient $A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,1} & a_{3,3} \end{bmatrix}$ et $B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & q_{3,1} & b_{3,3} \end{bmatrix}$, la somme de A et B donne S :
$$S = A + B = \begin{bmatrix} s_{1,1} = a_{1,1} + b_{1,1} & s_{1,2} = a_{1,2}+b_{1,2} & s_{1,3} = a_{1,3}+b_{1,3} \\ s_{2,1} = a_{2,1}+b_{2,1} & s_{2,2} = a_{2,2}+b_{2,2} & s_{2,3} = a_{2,3}+b_{2,3} \\ s_{3,1} = a_{3,1}+b_{3,1} & s_{3,2} = a_{3,1+}b_{3,1} & s_{3,3} = a_{3,3}+b_{3,3} \end{bmatrix}$$
Plus généralement, la somme de deux matrices A et B de mêmes dimensions $n \times m$ est une matrice S telle que $s_{i,j} = a_{i,j} + b_{i,j}$.
Définir les fonctions **`somme`** qui renvoie la somme des 2 matrices passées en arguments et représentées par des listes.
4. Le produit des 2 matrices A et B de dimensions respectives $n_A \times m$ et $m \times n_B$ est une matrice P de dimension $n_A \times n_B$ telle que :$p_{i,j} = \sum_{k=1}^{m} a_{i,k} \times b_{k,j}$
$$P = A \times B = \begin{bmatrix} p_{1,1} = \sum_{k=1}^{m} a_{1,k} \times b_{k,1} & p_{1,2} = \sum_{k=1}^{m} a_{1,k} \times b_{k,2} & p_{1,3} = \sum_{k=1}^{m} a_{1,k} \times b_{k,3} \\ p_{2,1} = \sum_{k=1}^{m} a_{2,k} \times b_{k,1} & p_{2,2} = \sum_{k=1}^{m} a_{2,k} \times b_{k,2} & p_{2,3} = \sum_{k=1}^{m} a_{2,k} \times b_{k,3} \\ p_{3,1} = \sum_{k=1}^{m} a_{3,k} \times b_{k,1} & p_{3,2} = \sum_{k=1}^{m} a_{3,k} \times b_{k,2} & p_{3,3} = \sum_{k=1}^{m} a_{3,k} \times b_{k,3} \end{bmatrix}$$
$$= \begin{bmatrix} a_{1,1} \times b_{1,1} + a_{1,2} \times b_{2,1} + a_{1,3} \times b_{3,1} & p_{1,2} & p_{1,3} \\ a_{2,1} \times b_{1,1} + a_{2,2} \times b_{2,1} + a_{2,3} \times b_{3,1} & p_{2,2} & p_{2,3} \\ a_{3,1} \times b_{1,1} + a_{3,2} \times b_{2,1} + a_{3,3} \times b_{3,1} & p_{3,2} & p_{3,3} \end{bmatrix}$$
Définir la fonction **`produit`** qui renvoie le produit des 2 matrices passées en arguments et représentées par des listes.

PARTIE 3 : Calculs de polynômes …



On se propose de manipuler formellement des polynômes, c'est à dire travailler sur leur expression. On rappelle qu'un polynôme est une expression de la forme :

$$P(X) = a_n X^n + a_{n-1} X^{n-1} + \ldots + a_1 X^1 + a_0 X^0 = \sum_{i=0}^{n} a_i X^i$$

où n est appelé le degré du polynôme et où chaque $a_i X^i$ est appelé monôme de degré i et de coefficient a_i

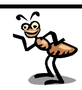
Chaque monôme $a_i X^i$ sera représenté par un couple (a_i, i) et chaque polynôme sera représenté par la liste de ses monômes $((a_0, 0), (a_1, 1), (a_2, 2), \ldots (a_n, n))$. <u>On notera que les monômes ne sont pas nécessairement rangés selon l'ordre des degrés</u>.

1) Comment sera représenté le polynôme : $P(X) = 7X^9 - 3X^7 + 2X^2 + 5$?
2) Donner une fonction **`degreMonome`** qui renvoie le degré du monôme qu'on lui passe en argument
3) Donner une fonction **`coefficientMonome`** qui renvoie le coefficient du monôme qu'on lui passe en argument
4) Donner une fonction **`degrePolynome`** qui renvoie le degré du polynôme qu'on lui passe en argument.
5) Donner une fonction qui renvoie la valeur d'un monôme pour une valeur de X précise. On appellera **`valeurMonome`** cette fonction. EX : **`(valeurMonome '(2 3) 2)`** renvoie 16.
6) Donner une fonction qui renvoie la valeur d'un polynôme pour une valeur de X précise. On appellera **`valeurPolynome`** cette fonction.
7) Donner une fonction qui renvoie la somme formelle de 2 polynômes.
8) Donner une fonction qui renvoie la dérivée formelle d'un polynôme. On rappelle que la dérivée du polynôme P(X) est P'(X) avec :

$$P'(X) = n \times a_n X^{n-1} + (n-1) \times a_{n-1} X^{n-2} + ... + 1 \times a_1 X^0$$

9) Donner une fonction qui renvoie la primitive formelle d'un polynôme.
10) Donner une fonction qui renvoie le produit formel de 2 polynômes.

PARTIE 4 : Gestion d'ne petite base de données ...



On se propose de gérer une petite base de données dans laquelle sont stockées des informations sur des personnes. Ces informations devront ensuite permettre de répondre aux requêtes posées par l'utilisateur.

Dans cette base, chaque personne est caractérisée par une identité, une adresse et un numéro de téléphone.

Chaque identité possède un nom, un prénom, une date de naissance et un identifiant unique.

Chaque adresse comporte un numéro, une rue, un code postal, une ville et un pays.

1) Définir les fonctions suivantes : **`creationAdresse`**, **`creationTelephonne`**, **`creationIdentite`** et **`creationPersonne`** qui renvoient une adresse, un numéro de téléphone , une identité ou une personne dont les champs sont remplis avec les données fournies en entré.

L'ensemble des personnes sera représenté par une liste de personnes dans laquelle chaque identifiant est unique. Donc lorsque l'utilisateur souhaite ajouter une nouvelle personne il conviendra de s'assurer que l'identifiant n'est pas déjà présent.

2) Définir les fonctions **`ajoutPersonne`** et **`detruirePersonne`** qui permettent d'ajouter ou détruire une personne dans la base.

L'utilisateur doit pouvoir faire des requêtes du type : **`(getPersonne maBase 'identifiant 234)`**, **`(getPersonne maBase 'nom 'Dupont),`** etc. dans lesquelles le 2ème argument est la base, le 3ème argument est le champ à considérer et le 4ème argument est la valeur du champ. Il peut y avoir une seule réponse ou plusieurs.

Par exemple pour la requête **`(getPersonne maBase 'identifiant 234)`** il ne devrait y avoir que 0 ou 1 réponse. Pour la **`requête (getPersonne maBase 'nom 'Dupont)`** il peut y avoir 0, 1 ou plusieurs répnse puisque plusieurs personnes peuvent avoir le même nom.

3) Définir les fonctions de base pour permettre à l'utilisateur de faire ses requêtes.

4) Afin de faciliter la lecture des réponses aux requêtes, on définira une fonction de tri qui trie chaque réponse fournie.