UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

FLMA607 – Calcul formel – 2009-2010

TP 0 : introduction à Maple

MAPLE est un logiciel de calcul formel. Il permet de faire des calculs symboliques, c'est-à-dire en conservant des variables inconnues. La programmation en MAPLE est de type impérative.

La version graphique du logiciel MAPLE peut être lancée en entrant la commande `xmaple` dans un terminal. Une version « en ligne de commande » est aussi disponible *via* la commande `maple`.

1 Premiers pas

- Avant de commencer un calcul, il est fortement conseillé d'exécuter la commande `restart;` de façon à oublier les valeurs qui ont pu être assignées à certaines variables lors de calculs précédents.
- Toutes les fonctions MAPLE ne sont pas accessibles dès le lancement de MAPLE. Il sera parfois nécessaire de charger certaines librairies de fonctions à l'aide de la commande `with` .
- Il est essentiel d'apprendre à utiliser l'aide de Maple. Elle est accessible *via* le menu Help ou en tapant un point d'interrogation (par exemple `?evalf`).

Exercice 1

1. Que fait la commande *Pi* ?
2. Que fait la commande *evalf* appliquée à *Pi* ?
3. Attribuer la valeur 50 à la variable *Digits*. Calculer l'aire d'un cercle de rayon 3.
4. Attribuer la valeur 200 à la variable *Digits*. Recalculer l'aire d'un cercle de rayon 3. Que remarque t on ?

Exercice 2

1. Entrer l'instruction `L1:=[2,4,6,8];` Que fait elle ?
2. Entrer la commande `L2:=seq(2*i,i=1..4);` Comparer avec L1.
3. Que représentent les valeurs retournées par les instructions `L1[2];` et `nops(L1);` ?

Exercice 3

1. Entrer l'instruction `test:=0=1;`
2. Appliquer la fonction `evalb` à test.
3. Assigner une valeur à la variable `x` puis entrer l'instruction
   `if (x=2) mod 6 then print(vrai); else print(faux); end if;`
   Commenter.
4. Que font les deux instructions suivantes ?
   ```
   a:=0; for i from 1 to 100 do a:=a+i: end do: print(a);
   i:=50; while not(isprime(i)) do i:=i+1: end do: print(i);
   ```

2 Procédures

Voici un exemple de procédure[1].

```
pgcd:=proc(a::integer, b::integer)::integer;
local r::integer;
   if b=0 then
      return a:          # Ceci est un commentaire.
   else
      r:=irem(a,b):      # irem(a,b) renvoie le reste de a modulo b.
      return pgcd(b,r):
   end if:
end proc;
```

Remarques :

- Si la procédure ne rencontre pas la commande `return`, elle retourne le résultat de la dernière évaluation.
- Il n'y a pas de ; après `nom:=proc(parametres)` si vous choisissez de ne pas déclarer le type du résultat.

Prenons tout de suite de bonnes habitudes :

- *Indentez* vos programmes.
- Utilisez abondamment les commentaires. Une procédure devrait toujours commencer par un commentaire expliquant brièvement ce qu'elle fait, et précisant ce que sont les variables d'entrée et ce que renvoie la procédure.
- Les erreurs relatives aux notions de variable globale ou locale sont fréquentes. Déclarez toujours les variables locales, même si MAPLE ne l'exige pas explicitement.
- De même déclarez autant que possible les types des variables, même si MAPLE ne l'exige pas explicitement.
- Lorsque vous en serez au stade où Maple accepte de compiler votre procédure mais renvoie une erreur d'exécution ou un résultat incorrect, la commande `debug` sera très utile...
- Lors des parties codage, il est parfois conseillé (surtout pour des programmes un peu longs) d'écrire son programme dans un éditeur de texte, de le sauvegarder puis d'utiliser la commande `read` afin de faire appel au fichier ainsi créé.

Exercice 4

Écrire une procédure qui, étant donné un entier naturel n et une base b, retourne la liste des coefficients du développement de n en base b (sans utiliser la commande `convert`...).

Exercice 5

Écrire une procédure qui étant donné deux polynômes P et Q, retourne le produit PQ. On représentera un polynôme sous forme d'une liste de coefficients.

Exercice 6

Écrire une procédure qui étant donné une base b et deux entiers naturels n, m écrits en base b, retourne le produit nm écrit en base b, et calculé à l'aide de l'écriture des entiers comme polynômes suivant la base b.

Exercice 7

Écrire une procédure qui étant donné une base b et un entier naturel n écrit en base b retourne l'écriture en base b de $n!$.

[1] Celle-ci est même *récursive*, c'est-à-dire qu'elle s'appelle elle-même.