TD/TP Maple : fonctions et procédures

Exercice 1

1. Ecrire une procédure Maple ProcTrie(L) qui trie la liste L (on pourra utiliser sort).
2. Ecrire une procédure Maple ProcAjout(x,L) qui ajoute un élément x à la fin de la liste L.
3. Ecrire une procédure Maple ProcInserer(x,pos,L) qui insère l'élément x dans L à la position pos.

La liste L est dans ces trois procédures un paramètre d'entrée-sortie.
On testera à chaque fois les procédures, pour vérifier que la procédure a effectivement **modifié la liste** placée en paramètre.

Exercice 2

Ecrire une procédure myMember(x,L,p) fonctionnant comme la fonction member de Maple.
Remarque : Member à la fois renvoie un résultat ($true/false$ suivant si x est dans L), et modifie p. Elle est donc à la fois une fonction (renvoie un résultat) et une procédure (modifie un paramètre). On parle de fonction procédurale.

Exercice 3

Exécutez les commandes Maple suivantes :

> g :=proc(L : :name) RETURN(nops(eval(L))) : end : L :=[4,5,6] : g(L) ; > g('L') ;	> g2 :=proc(L : :name) RETURN(nops(L)) : end : L :=[4,5,6] : g2('L') ;

puis les commandes :

> g :=proc(x : :name) x :=x+1 ; end : x :=3 ;g('x') :x ;	> g2 :=proc(x : :name) x :=eval(x)+1 ; end : x :=3 ;g2('x') :x ;

Expliquez les résultats obtenus.

Exercice 4

Dans une entreprise de service, un tarif est le prix de l'heure et compte tenu des changements de tarif, ce prix est accompagné d'une date d'effet. Ainsi un tarif peut s'implanter par la liste [jour, mois, année, prix].
Pour conserver les modifications, on considère la liste Tarifs qui contient tous les tarifs utilisés depuis la création de l'entreprise, rangés du plus récent au plus ancien.
Exemple : Tarifs :=[[1,6,2002,200],[11,5,1998,160],[15,1,1997,120],[1,1,1995,100]]

1. Quel est le prix horaire le 15/09/1996 ? Ecrire en Maple une fonction **prix :=proc(Tarifs, date)** qui détermine le tarif en cours à la date donnée, date étant du type [jour, mois, année].
2. Ecrire en Maple **nouveauT :=proc(...)** ayant comme paramètre d'entrée la liste Tarifs et permettant d'ajouter un nouveau tarif dont les valeurs seront saisies au clavier. On vérifiera que le nouveau tarif a une date d'effet postérieure au tarif en cours.
3. Chaque Intervention est implantée comme une liste avec la date de prestation, le nombre d'heures correspondant à la prestation et le nom du client, soit

   intervention :=[jour, mois, année, nombre_heures, nom_client].

   Quel est le montant de [12,7,1999,6, "Dupont"] en prenant pour tarifs l'exemple donné ? Ecrire une fonction **montant** ayant pour premier paramètre la liste des tarifs et comme deuxième paramètre une intervention, et calculant le montant de la prestation au moment ou elle a été effectuée.

Exercice 5

On relève certains jours de l'année le niveau des précipitations. Une observation sera une liste de 3 valeurs [< numéro du jour >,< numéro du mois >, < niveau >]. Par exemple, 10mm d'eau observés le 13 février correspond à la liste [13,2,10]. On suppose qu'une variable **Observations** contient une liste de telles observations. C'est donc une liste de listes.

– On définit la variable **Mois :=[31,28,31,30,31,30,31,31,30,31,30,31]** correspondant aux nombres de jours des mois d'une année non bissextile. Définir une fonction **numjour** qui, à une observation donnée, associe le numéro du jour dans l'année. Par exemple, **numjour**([12,2,10]) vaut 43 (31 jours de janvier + 12 jours de février).
– Définir une fonction à valeurs booléennes **correct**(x) qui retourne true si l'observation x correspond à un mois compris entre 1 et 12, un jour entre 1 et le nombre de jours du mois donné, et un niveau de précipitations positif ou nul.
– Donner une ou des instructions permettant de supprimer les observations incorrectes (si elles existent) de la variable **Observations**.
– Donner une fonction booléenne **Avant**(x,y) prenant la valeur true si l'observation x a lieu avant l'observation y.

Exercice 6

Une association reçoit des dons. Un donateur peut effectuer plusieurs dons. Chaque don est implanté sous la forme d'une liste de deux éléments, le premier étant le nom du donateur en chaîne de caractères, le deuxième le montant en franc (sans centime). L'ensemble des dons est implanté sous forme d'une liste de dons. On a donc une liste de listes. Par exemple, on pourra avoir une liste de dons égale à :
[["ELF",10000],["BNP",10000],["MARTIN",500],["DUPONT",300],["MARTIN",200]]

1. Ecrire une procédure ajout(L : : name,Nom : : string,Montant : : integer) qui ajoute à la fin de la liste L le nouveau don (dont le nom et le montant sont les deux autres paramètres).
2. Ecrire en Maple une fonction `Moy` donnant à partir d'une liste L de dons le montant moyen des dons.
3. En fin d'année on veut une liste des dons où chaque donateur apparaît une seule fois avec le montant total de tous ses dons. Ecrire la fonction `Cloture` qui à partir de L construit et renvoie cette liste.
4. Donner la ou les commande(s) Maple permettant d'obtenir le don moyen par donateur.

Exercice 7

Soit $S_n(x) = \sum_{k=0}^{n} \frac{x^k}{k!}$.
- Trouver la formule de récurrence entre S_{n+1} et S_n.
- Ecrire en Maple une fonction Somme récursive ayant pour paramètres x et n et renvoyant $S_n(x)$.

Exercice 8

Ecrire une fonction récursive permettant de calculer le n° terme de la suite de Fibonacci définie par $u_0 = u_1 = 1$ et, pour tout $n \geq 2$, $u_n = u_{n-1} + u_{n-2}$.

Exercice 9

Jeu du pendu
On veut réaliser un jeu du pendu. On a donc un mot stocké dans une variable S de type string (par exemple S :="acapulco"). Nous allons créer une chaîne de caractères Mot correspondant à la situation où en est le joueur (avec les lettres qu'il a déjà trouvées). Supposons que l'on ait perdu au bout de 10 tentatives. Il faut donc demander une lettre au joueur. Si la lettre est dans le mot S à trouver, on la remplace dans Mot. Si la lettre n'y est pas, alors le joueur a une chance de moins. Si le mot est trouvé, alors le joueur a gagné. S'il ne reste plus de tentative, le joueur a perdu.
Pour cela, nous allons procéder comme suit.

1. Créer une fonction `MotVide` prenant en paramètre une chaîne de caractères et créant une chaîne de la même taille formée de caractères -. Par exemple, `MotVide`("acapulco") doit renvoyer "- - - - - - - -"
2. Créer une fonction `EstDedans` prenant en paramètres une chaîne de caractères S et une lettre (sous forme d'une chaîne d'un caractère) et renvoyant *true* si lettre est dans S, et *false* sinon. Par exemple `EstDedans`("acapulco","b") renvoie *faux*.
3. Créer une fonction `Remplace` prenant en paramètres deux chaînes de caractères S1 et S2 et une lettre (sous forme de chaîne d'un caractère). Cette fonction renvoie une chaîne formée à partir de S2 où l'on a mis la lettre à la bonne place (ou aux bonnes places), i.e. la où elle apparaît dans S1. Par exemple, `Remplace`("acapulco","-c-pu-c-","a") renvoie "acapu-c-"
4. Créer une fonction `Pendu` prenant en paramètre une chaîne de caractères et simulant un jeu du pendu.