4. En supplément :
   On peut noter un tableau de 3 colonnes et 3 lignes en une tableau de 9 chiffre. Ecrivez une procédure testant si un tableau 3x3 est un carré magique (toutes les sommes sur lignes, colonnes et les 2 diagonales sont identiques).
   Généralisez pour un tableau nxn.
   Généralisez pour les carrés antimagiques (toutes les sommes sont différentes).

Transformations et simplifications

Documentez-vous dans l'aide Maple sur les fonctions suivantes, et utilisez-les avec des variantes des exemples proposés dans cette aide.

- expand
- combine
- factor
- collect
- convert
- normal
- simplify
- sort

fonctions, et surtout, elles sont très utiles lorsqu'on fait les mêmes opérations en modifiant la valeur des variables.
La syntaxe type est la suivante :

```
nomproc :=proc (paramètres)
  local variableslocales ;
  global variablesglobales ;
  options optionsutilisées ;
  instructions ;
end ;
```

- *nomproc* : nom de cette procédure ; on l'utilisera avec l'instruction nomproc(….), qui nous renverra le résultat des instructions de la procédure.
- *paramètres* : la liste des variables dont dépend directement la procédure, et qui doivent être spécifiées à chaque appel à la procédure.
- *variableslocales* : liste des variables qui n'apparaissent que dans la procédure ; pour cet usage, vous pouvez utiliser les mêmes noms que d'autres variables existant déjà à l'extérieur de la procédure, sans que cela pose de problème, à condition de garder les idées claires.
- *variablesglobales* : liste des variables que le procédure devra aller chercher 'en-dehors' de sa définition.
- *optionsutilisées* : plusieurs options sont possibles ; on n'utilisera queb l'option remember, qui accélère les calculs en conservant les résultats déjà obtenus.

Le résultat de la procédure peut être affiché par un print (la dernière variable évaluée est affichée), ou bien stocké dans une variable globale.

Quelques exemples :

```
> fact :=proc(n)
   option remember;
   if (n=0 or n=1) then 1 else n*fact(n-1)  fi;
   end;
> nbpremiers:=proc(n)
   local i,N,L;
   N:=0; L:=[];
   for i from 1 to n do
    N:=nextprime(N);
    L:=[op(L),N] od;
   L;
   end;
```

De la même manière, vous pouvez réécrire en procédure les blocs d'instructions élaborés lors du TD précédent, afin de les réutiliser à volonté.

Exercices :

1. Ecrire une procédure approchant 3^1/5, à n décimales près, n étant le paramètre de la procédure.
   - pensez à fixer Digits :=n+1, au début de la procédure, puis à remettre Digits :=10 à la fin.
   - réutilisez les instructions écrites la semaine dernière, ou bien copiez celles du corrigé des exercices, en cherchant tout d'abord à comprendre comment elles fonctionnent.
2. Ecrire une procédure semblable, pour approcher x^1/y à n décimales près, x, y et n étant trois paramètres de la procédure.
3. Crible d'Eratosthène (détermination de nombres premiers)
   Ecrire une procédure qui prend n comme argument, crée un tableau (fonctionne comme une liste, sauf que le $n^{ième}$ élément du tableau T est T[n]) de n fois le chiffre 1, puis remplace ces 1 par des 0 pour le $4^{ème}$,$6^{ème}$, $8^{ème}$ élément, et ensuite pour les $6^{ème}$, $9^{ème}$ puis pour les multiples de 4, 5, 6…
   Il ne reste à la fin que des 1 qui correspondent à des nombres premiers (pourquoi ?). Les afficher.
   Vous pouvez éventuellement chercher ensuite à améliorer la rapidité de la méthode.

Maple TD 3

Infos diverses
Reprise de la fin du TD 2 :
-Fonctions
-Procédures
-Transformations et simplifications

Infos diverses

Ce TD consistera en une (dernière) reprise du TD2, afin de pouvoir faire dès la semaine prochaine des choses vraiment liées à vos cours de mathématiques.
Au fur et à mesure de l'avancée des TD, des corrigés des exercices seront disponibles.
L'examen partiel aura lieu dans la semaine du 20 novembre ; vu la nécessité d'1 poste par personne, il s'agira probablement d'un examen d'1h, et le groupe sera dédoublé.
A priori, la note de ce cours sera égale au maximum de la note de l'examen final et de la moyenne du final et du partiel.
Remarque : pour obtenir votre diplôme, la compensation entre les modules est automatique, mais vous devez absolument vous être présentés à tous vos examens, partiels, finaux, et éventuellement session de septembre…

Reprise TD2

La partie programmation (boucles et tests) doit être vue ; dans le cas contraire, vous disposez des horaires de logithèque en libre-accès, ainsi que du corrigé, et des livres de la bibliothèque, pour reprendre si nécessaire.
Les parties fonctions et procédures vont être traitées ci-après ; les transformation et simplifications sont à lire ensuite, et à chercher dans l'aide. Il est IMPORTANT de savoir les retrouver…

Fonctions

Maple permet de définir 'en général' une fonction de la même manière qu'en mathématiques, c'est-à-dire une manière de transformer des objets mathématiques en d'autres objets mathématiques. On peut définir des fonctions transformant un nombre en un autre, transformant plusieurs nombres en un nombre, ou un en plusieurs, mais aussi des fonctions agissant par exemple sur des ensembles, etc…
Pour définir une fonction (qui, comme tout objet créé dans Maple, disparaît avec restart), il faut indiquer son nom, les variables qui y 'entrent', ainsi que ce qui en 'sort'. Ainsi, on peut définir des polynômes d'une ou plusieurs variables:

```
> P1:=x->x^3+x^2+x+1;
> P2:=(x,y)->x*y^2-x^2*y;
> P3:=(alpha,beta,z)->alpha*z^2+beta*z+1;
> P4:=(beta,z)->P3(beta^2,beta,z);
> P4(b,z);
```

Mais il est aussi possible de définir des fonctions avec des tests, ou des fonctions faisant référence à ellesmêmes :

```
> f:=(n)->if type(n,integer) then sum(1/'i',i=1..n) else ERROR(`n non entier`) fi;
> facto:=(n)->if n=0 then 1 elif  n=1 then 1 else nfacto(n-1) fi;
```

Dans ce cas, Maple voit ici des procédures, que nous étudierons plus loin, mais cela est transparent pour nous.
Ensuite, il suffit d'utiliser fonction(…), avec ses variables, pour remplacer tout ce que vous avez mis dans la définition de la fonction. Ainsi, l'objet 'fonction' permet de simplifier l'utilisation de Maple, en évitant les répétitions. Il en est de même pour les procédures…

Procédures

Les procédures consistent en des petits programmes, regroupant plusieurs instructions, auxquels on donne un nom, et qui restent en mémoire (jusqu'au prochain restart). Elles sont un peu plus larges et plus souples que les