§ 3 Listes

§ 3.0 Notion de liste

Les listes jouent un rôle important en *Mathematica*. Lorsque nous aurons un problème à résoudre, nous allons fréquemment considérer que l'input est une liste et que l'output est une liste. Le calcul à faire consiste à transformer la liste donnée - l'input - pour obtenir la liste voulue - l'output .

Pour définir des listes, on utilise des accolades { } et les éléments sont séparés par des virgules. Une liste est ordonnée et peut contenir des éléments répétés, il s'agit donc d'un objet plus général qu'un ensemble :

```
chiffresPairs = {0, 2, 4, 6, 8}
```

```
{0, 2, 4, 6, 8}
```

```
i = {1, 0}
```

```
{1, 0}
```

```
ensembleVide = {}
```

```
{}
```

```
notes = {4, 4.5, 4.5, 4}
```

```
{4, 4.5, 4.5, 4}
```

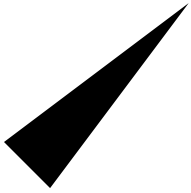
Un triangle peut être représenté par les coordonnées de ses trois sommets, c'est-à-dire une liste contenant trois listes de deux éléments:

```
sommets = {{0, 0}, {3, 4}, {-1, 1}}
```

```
{{0, 0}, {3, 4}, {-1, 1}}
```

On utilise les listes dans la plupart des commandes, p. ex.:

```
Graphics[Triangle[sommets], ImageSize → {200, 100}]
```
graphique triangle taille d'image



Les éléments d'une liste peuvent être de natures différentes:

$$\left\{4, \frac{\pi}{12}, x^2 + 1, \{a, \{b, c\}\}, \text{"Un texte"}\right\}$$

$$\left\{4, \frac{\pi}{12}, 1 + x^2, \{a, \{b, c\}\}, \text{Un texte}\right\}$$

Une liste peut représenter diverses structures de données : un vecteur, un ensemble, un tableau, une matrice, un arbre, etc. Dans le cas où une liste représente un vecteur ou une matrice, celle - ci peut être affichée sous cette forme grâce à la fonction **MatrixForm**:

```
MatrixForm[i]
```
apparence matricielle

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

S'il s'agit d'un tableau, on utilse la fonction **TableForm.** Parmi les options possibles, citons **TableHeadings** qui permet de mettre des en-têtes aux lignes et/ou aux colonnes:

```
TableForm[sommets, TableHeadings → {None, {"x", "y"}}]
```
forme de table en-têtes de table aucun

x	y
0	0
3	4
-1	1

```
Clear["Global`*"]
```
efface

§ 3.1 Construction de listes

3.1.1 Construction avec Table

La fonction **Table** permet de créer des listes:

- **Table[expr, imax]** produit une liste de imax “copies” de expr.
- **Table[expr, {i, imax}]** produit une liste des valeurs de expr lorsque i va de 1 à imax.
- **Table[expr, {i, imin, imax}]** commence à i = imin.
- **Table[expr, {i, imin, imax,di}]** utilise des incréments di.
- **Table[expr, {i, imin, imax}, {j, jmin, jmax}, ...]** donne une liste imbriquée. La liste associée à i est la liste la plus extérieure.

Voici quelques exemples:

```
Table[RandomInteger[10], 5]
```
table entier aléatoire

```
{8, 7, 7, 5, 4}
```

```
Table[3 + 5 i, {i, 10}]
```
table

```
{8, 13, 18, 23, 28, 33, 38, 43, 48, 53}
```

```
Table[3 + 2^k, {k, 0, 10}]
```
table

```
{4, 5, 7, 11, 19, 35, 67, 131, 259, 515, 1027}
```

L’exemple suivant génère un tableau de valeurs:

```
f[x_] := x^2 - 5
ptsF = Table[{x, f[x]}, {x, -2, 2, 0.5}];
```
table

```
TableForm[ptsF, TableHeadings → {None, {"x", "f(x)"}}]
```
forme de table en-têtes de table aucun

x	f(x)
-2.	-1.
-1.5	-2.75
-1.	-4.
-0.5	-4.75
0.	-5.
0.5	-4.75
1.	-4.
1.5	-2.75
2.	-1.

Voici un exemple de listes imbriquées:

```
Table[i + j, {i, 1, 3}, {j, 1, 5}]
```
table

$$\{\underbrace{\{2, 3, 4, 5, 6\}}_{\substack{i=1\\ j=1,2,3,4,5}}, \underbrace{\{3, 4, 5, 6, 7\}}_{\substack{i=2\\ j=1,2,3,4,5}}, \underbrace{\{4, 5, 6, 7, 8\}}_{\substack{i=3\\ j=1,2,3,4,5}}\}$$

```
Clear[f, ptsF]
```
efface

3.1.2 Construction avec Range

La fonction Range génère des listes dont la différence entre deux éléments est constante (c’est-à-dire des listes dont les éléments forment une suite arithmétique):

- **Range[imax]** produit la liste {1, 2, ... , imax}.
- **Range[imin, imax]** produit la liste {imin, ... , imax}.
- **Range[imin, imax, di]** utilise l'incrément di.

Toutes les listes obtenues avec **Range** peuvent donc être obtenues avec **Table**, l’avantage de **Range** étant une notation plus courte.

```
Range[0, 9]
```
plage

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
Range[99, 10, -5]
```
plage

```
{99, 94, 89, 84, 79, 74, 69, 64, 59, 54, 49, 44, 39, 34, 29, 24, 19, 14}
```

3.1.3 Construction avec NestList

Dans certaines listes, on passe d'un élément à l'autre en appliquant toujours la même fonction f. On crée ce genre de liste grâce à la fonction **NestList**: **NestList[f, expr, n]** renvoie la liste des résultats de l'application de **f** à **expr** de 0 à **n** fois.

Par exemple, dans une suite arithmétique, on passe d'un élément au suivant en lui ajoutant un terme constant dénommé "raison". Voici la construction des 13 premiers termes d'une suite arithmétique de premier terme 2, de raison 3:

```
successeur[x_] := x + 3;
NestList[successeur, 2, 12]
```
liste d'imbrication

```
{2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38}
```

Une suite géométrique est caractérisée par le fait que l'on passe d'un terme au suivant en le multipliant par un facteur constant dénommé "raison". Voici par exemple les 11 premiers termes d'une suite géométrique de premier terme 2, de raison 3:

```
successeur[x_] := 3 x;
NestList[successeur, 2, 10]
```
liste d'imbrication

```
{2, 6, 18, 54, 162, 486, 1458, 4374, 13 122, 39 366, 118 098}
```

```
Clear[successeur]
```
efface

3.1.4 Construction avec FixedPointList

Les principes de construction de listes avec **FixedPointList** et **NestList** sont similaires: dans les deux cas, on donne le premier élément de la liste et la fonction qui est appliquée afin de calculer l'élément suivant. Ce qui change, c'est la longueur de la liste : pour **NestList** la longueur est fixée par l'utilisateur tandis que pour **FixedPointList** la construction s'arrête lorsque le résultat demeure inchangé. La syntaxe est donc la suivante: **FixedPointList[f,expr]**. Voici un exemple:

```
a = FixedPointList[Sqrt, 1.5]
```
liste de point fixe racine carrée

```
{1.5, 1.22474, 1.10668, 1.05199, 1.02567, 1.01275, 1.00636, 1.00317, 1.00159, 1.00079, 1.0004,
 1.0002, 1.0001, 1.00005, 1.00002, 1.00001, 1.00001, 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.}
```

Pour mieux comprendre la situation, il faut afficher toutes les décimales calculées par *Mathematica*:

```
NumberForm[a, 16]
```
apparence numérique

```
{1.5, 1.224744871391589, 1.106681919700321, 1.051989505508644, 1.025665396466432,
 1.012751399143162, 1.006355503360101, 1.003172718608367, 1.001585103028378,
 1.000792237693907, 1.000396040422945, 1.000198000609352, 1.000098995404631,
 1.000049496477365, 1.000024747932452, 1.00001237388967, 1.000006186925696,
 1.000003093458063, 1.000001546727835, 1.000000773363619, 1.000000386681735,
 1.000000193340849, 1.00000009667042, 1.000000048335209, 1.000000024167604,
 1.000000012083802, 1.000000006041901, 1.00000000302095, 1.000000001510475,
 1.000000000755237, 1.000000000377619, 1.000000000188809, 1.000000000094405,
 1.000000000047202, 1.000000000023601, 1.000000000011801, 1.0000000000059, 1.00000000000295,
 1.000000000001475, 1.000000000000738, 1.000000000000369, 1.000000000000184,
 1.000000000000092, 1.000000000000046, 1.000000000000023, 1.000000000000012,
 1.000000000000006, 1.000000000000003, 1.000000000000001, 1.000000000000001, 1.}
```

Il est possible de modifier le critère d'arrêt. Par exemple, on lieu d'exiger que ≪deux éléments successifs sont égaux≫, on peut demander que ≪la valeur absolue de la différence entre deux éléments consécutifs est plus petite que 10^{-5}≫ :

```
arret[p_, q_] := Abs[p - q] < 10^-5;
                 valeur absolue
FixedPointList[Sqrt, 1.5, SameTest → arret]
liste de point fixe  racine carrée  critère d'égalité
```

```
{1.5, 1.22474, 1.10668, 1.05199, 1.02567, 1.01275, 1.00636, 1.00317,
 1.00159, 1.00079, 1.0004, 1.0002, 1.0001, 1.00005, 1.00002, 1.00001, 1.00001}
```

Attention, la commande **FixedPoint[Sqrt, $\frac{3}{2}$]** produit la liste

$$\frac{3}{2},\ \sqrt{\frac{3}{2}},\ \sqrt{\sqrt{\frac{3}{2}}},\ \sqrt{\sqrt{\sqrt{\frac{3}{2}}}},\ \ldots$$

dont les termes successifs sont tous différents. Pour éviter d'obtenir une liste infinie, il faut effectuer les calculs, non avec des valeurs exactes, mais avec des valeurs numériques approchées. C'est pourquoi nous sommes partis du nombre réel 1.5 écrit en virgule flottante.

§ 3.2 Opérations arithmétiques sur les listes

Considérons les listes suivantes:

```
sArith = Table[3 + 5 i, {i, 0, 10}]
         table
```

```
{3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53}
```

```
suite = Table[3 + 2^k, {k, 0, 10}]
        table
```

```
{4, 5, 7, 11, 19, 35, 67, 131, 259, 515, 1027}
```

```
sCube = Table[n^3, {n, -2, 5}]
        table
```

```
{-8, -1, 0, 1, 8, 27, 64, 125}
```

Mathematica autorise d'appliquer les opérations arithmétiques à des listes. On peut additionner/soustraire/multiplier/diviser deux listes de même longueur, c'est-à-dire additionner/soustraire/multiplier/diviser les éléments de même position:

```
sArith + suite
```

```
{7, 13, 20, 29, 42, 63, 100, 169, 302, 563, 1080}
```

```
sArith / suite
```

$$\left\{\frac{3}{4}, \frac{8}{5}, \frac{13}{7}, \frac{18}{11}, \frac{23}{19}, \frac{4}{5}, \frac{33}{67}, \frac{38}{131}, \frac{43}{259}, \frac{48}{515}, \frac{53}{1027}\right\}$$

```
sArith + sCube (* les suites n'ont pas la même longueur *)
```

Thread::tdlen : Objects of unequal length in {3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53} + {-8, -1, 0, 1, 8, 27, 64, 125} cannot be combined. >>

```
{-8, -1, 0, 1, 8, 27, 64, 125} + {3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53}
```

On peut ajouter ou soustraire un nombre à une liste, c'est-à-dire à chaque élément de la liste:

```
sCube - 10
```

```
{-18, -11, -10, -9, -2, 17, 54, 115}
```

On peut multiplier/diviser une liste par un nombre, c'est-à-dire multiplier/diviser chaque élément de la liste par un nombre:

```
sCube * 5
```

```
{-40, -5, 0, 5, 40, 135, 320, 625}
```

Les opérations arithmétiques définies par *Mathematica* sont donc une extension des opérations sur les vecteurs vues dans le cadre du cours de géométrie vectorielle.

```
Clear[sArith, suite, sCube]
efface
```

§ 3.3 Fonctions appliquées à des listes

3.3.1 Fonctions listables

Certaines fonctions, lorsqu'elles sont appliquées à des listes, sont distribuées aux éléments de la liste. On dit qu'elles sont "**Listable**" En voici quelques exemples:

```
sAngles = Table[π/4 + k π/6, {k, -3, 3}];
```
table

```
Cos[sAngles]
```
cosinus

$$\left\{\frac{1}{\sqrt{2}}, \frac{1+\sqrt{3}}{2\sqrt{2}}, \frac{1+\sqrt{3}}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{-1+\sqrt{3}}{2\sqrt{2}}, -\frac{-1+\sqrt{3}}{2\sqrt{2}}, -\frac{1}{\sqrt{2}}\right\}$$

```
f[x_] := 12 x / Pi;
```
nom

```
f[sAngles]
```

```
{-3, -1, 1, 3, 5, 7, 9}
```

Les exponentielles sont aussi des fonctions listables:

$$2^{\{3,4,5\}}$$

```
{8, 16, 32}
```

```
Clear[f, sAngles]
```
efface

3.3.2 Fonctions de base

Considérons la liste suivante:

```
a = {6, {1, 5}, 5, 3, {2, {7, 5}}};
```

La fonction **Length** donne le nombre d'éléments d'une liste:

```
Length[{4, 5, 5}]
```
longueur

```
3
```

```
Length[a]
```
longueur

```
5
```

Les fonctions **Max** et **Min** donnent le plus grand et le plus petit nombre contenu dans une liste et ses sous-listes:

```
Max[a]
```
maximum

```
7
```

Pour savoir si un élément appartient à une liste, on peut utiliser la fonction **MemberQ**:

```
MemberQ[a, 3]
```
est membre?

```
True
```

Contrairement aux fonctions **Min** et **Max**, **MemberQ** n'examine que le premier niveau de la liste:

```
MemberQ[a, 7]
```
est membre?

```
False
```

Pour déterminer les positions d'un élément dans une liste, à n'importe quel niveau de celle-ci, on utilise la fonction **Position**:

```
Position[a, 5]
```
position

```
{{2, 2}, {3}, {5, 2, 2}}
```

Il est possible de limiter la recherche jusqu'à un certain niveau:

```
Position[a, 5, 2]
position
```

```
{{2, 2}, {3}}
```

Pour déterminer le nombre d'occurrences d'un élément dans une liste, on utilise la fonction **Count**:

```
Count[a, 6]
compte
```

```
1
```

Comme **MemberQ**, **Count** n'examine que le premier niveau de la liste:

```
Count[a, 5]
compte
```

```
1
```

Exemple: simulation de 100 jets de dé

Comme application de la fonction **Count**, simulons 100 lancés successifs d'un dé à 6 faces et comptons les occurences de chacun des résultats possibles:

```
jets = RandomInteger[{1, 6}, 100]
       entier aléatoire
frequence = Table[{i, Count[jets, i]}, {i, 1, 6}];
            table     compte
TableForm[frequence,
forme de table
 TableHeadings → {None, {"Face", "Nombre d'occurences"}},
 en-têtes de table    aucun
 TableAlignments → Center]
 alignements de table   centre
```

```
{3, 6, 3, 6, 3, 5, 2, 3, 3, 1, 1, 4, 4, 2, 6, 5, 3, 5, 3, 4, 4, 3, 5, 4, 6, 2, 4, 4, 6, 2, 2, 1, 1, 4, 5,
 5, 4, 3, 2, 1, 3, 5, 4, 1, 2, 2, 6, 4, 1, 6, 1, 3, 5, 6, 6, 5, 1, 2, 5, 6, 5, 6, 4, 6, 1, 2, 5, 4,
 5, 6, 1, 1, 3, 4, 4, 4, 4, 4, 1, 6, 3, 4, 5, 3, 3, 5, 2, 6, 1, 3, 5, 6, 3, 5, 3, 6, 4, 6, 3, 2}
```

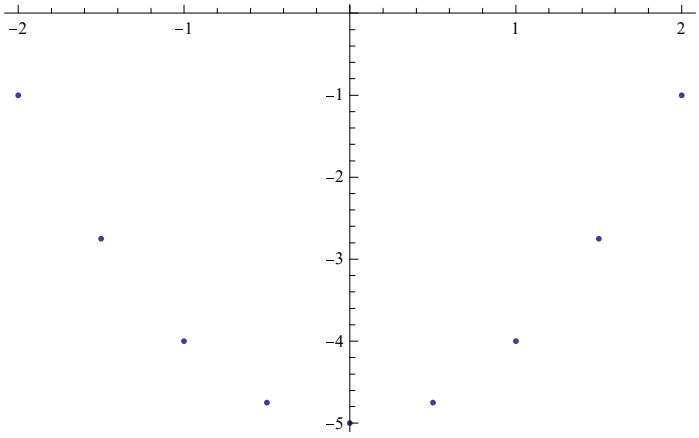
Face	Nombre d'occurences
1	14
2	12
3	19
4	20
5	17
6	18

```
Clear[a, jet, frequence]
efface
```

3.3.3 ListPlot

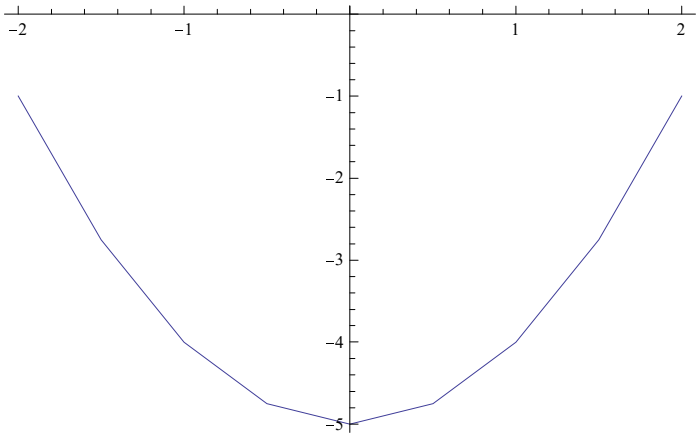
La commande **ListPlot dessine** une liste de points:

```
f[x_] := x^2 - 5
ptsF = Table[{x, f[x]}, {x, -2, 2, 0.5}];
       table
ListPlot[ptsF]
tracé de liste
```



Si on désire relier les points par une ligne il faut modifier la valeur l'option **Joined**:

```
ListPlot[ptsF, Joined → True]
tracé de liste   joint   vrai
```




```
Clear[f, ptsF]
efface
```

3.3.4 Flatten

Considérons la liste suivante:

```
liste = Table[{i, j}, {i, 1, 5}, {j, 1, 5}]
        table
```

```
{{{1, 1}, {1, 2}, {1, 3}, {1, 4}, {1, 5}},
 {{2, 1}, {2, 2}, {2, 3}, {2, 4}, {2, 5}}, {{3, 1}, {3, 2}, {3, 3}, {3, 4}, {3, 5}},
 {{4, 1}, {4, 2}, {4, 3}, {4, 4}, {4, 5}}, {{5, 1}, {5, 2}, {5, 3}, {5, 4}, {5, 5}}}
```

On aimerait réorganiser la liste précédente de telle manière qu'elle devienne une liste de 25 points dont le premier élément est {1, 1}. Pour ce faire, il faut supprimer les accolades de niveau 1. Ceci est possible grâce à la fonction **Flatten**:

- **Flatten[list]** aplatit complètement les listes imbriquées.
- **Flatten[list, n]** aplatit jusqu'au niveau n.
- D'autres utilisations de **Flatten** sont développées dans l'aide.

```
Flatten[liste, 1]
aplatis
```

```
{{1, 1}, {1, 2}, {1, 3}, {1, 4}, {1, 5}, {2, 1}, {2, 2},
 {2, 3}, {2, 4}, {2, 5}, {3, 1}, {3, 2}, {3, 3}, {3, 4}, {3, 5}, {4, 1},
 {4, 2}, {4, 3}, {4, 4}, {4, 5}, {5, 1}, {5, 2}, {5, 3}, {5, 4}, {5, 5}}
```

Par défaut, la fonction **Flatten** enlève toutes les accolades intérieures:

```
Flatten[liste]
aplatis
```

```
{1, 1, 1, 2, 1, 3, 1, 4, 1, 5, 2, 1, 2, 2, 2, 3, 2, 4, 2, 5, 3, 1, 3,
 2, 3, 3, 3, 4, 3, 5, 4, 1, 4, 2, 4, 3, 4, 4, 4, 5, 5, 1, 5, 2, 5, 3, 5, 4, 5, 5}
```

Il est possible d'éliminer que les accolades des n premiers niveaux:

```
Flatten[{11, {22, {33, {44, {55, {66}}}}}}, 2]
aplatis
```

```
{11, 22, 33, {44, {55, {66}}}}
```

Exemple: sélection des éléments positifs d'une liste

Sélectionnons les valeurs positive de la liste suivante:

```
a = Cos[Range[0, 6]]
    co··· plage
```

```
{1, Cos[1], Cos[2], Cos[3], Cos[4], Cos[5], Cos[6]}
```

La fonction **Sign** retourne 1 si l'argument est positif, 0 si l'argument est nul et -1 si l'argument est négatif. Appliquons cette fonction à la liste **a** pour afin d'identifier ses éléments positifs:

```
Sign[a]
signe
```

```
{1, 1, -1, -1, -1, 1, 1}
```

Nous pouvons maintenant déterminer la position des éléments de la liste **a** qui sont positifs:

```
pos = Position[Sign[a], 1]
      position   signe
```

```
{{1}, {2}, {6}, {7}}
```

Malheureusement les résultats est une liste de listes, il faut donc utiliser la fonction **Flatten** et les éléments à retenir sont

```
a[[Flatten[pos]]]
   aplatis
```

```
{1, Cos[1], Cos[5], Cos[6]}
```

```
Clear[liste, a, pos]
efface
```

Exemple: coïncidences de deux listes

Une coïncidence de deux listes signifie non seulement qu'un élément se trouve dans les deux listes mais encore qu'il est situé à la même position dans chaque liste. Pour déterminer les coïncidences entre deux listes, on peut observer la position des zéros dans la différence des deux listes:

```
a = {86, 34, 45, 97, 13, 23};
b = {45, 34, 39, 45, 13, 97};
a - b
```

```
{41, 0, 6, 52, 0, -74}
```

```
Position[a - b, 0]
position
```

```
{{2}, {5}}
```

Remarquez que l'élément 45 est dans l'intersection des deux listes mais n'est pas une coïncidence car il ne se trouve pas à la même position dans les deux listes.

```
pos = Flatten[Position[a - b, 0]]
      aplatis   position
```

```
{2, 5}
```

```
a[[pos]]
```

```
{34, 13}
```

On peut aussi déterminer le nombre de coïncidences entre deux listes:

```
Count[a - b, 0]
compte
```

```
2
```

```
Clear[a, b, pos]
efface
```

3.3.5 Transpose

Quelle relation y a-t-il entre les deux listes suivantes ?

```
sPts = {{x1, y1}, {x2, y2}, {x3, y3}, {xn, yn}};
sao = {{x1, x2, x3, xn}, {y1, y2, y3, yn}};
```

Pour la mettre en évidence, affichons ces listes sous la forme de tableaux appelés matrices:

```
MatrixForm[sPts]
apparence matricielle
```

$$\begin{pmatrix} x1 & y1 \\ x2 & y2 \\ x3 & y3 \\ xn & yn \end{pmatrix}$$

```
MatrixForm[sao]
apparence matricielle
```

$$\begin{pmatrix} x1 & x2 & x3 & xn \\ y1 & y2 & y3 & yn \end{pmatrix}$$

On observe que les lignes de la première sont les colonnes de la deuxième. Dans un tel cas, on dit que la deuxième matrice est la transposée de la première (et la première est la transposée de la deuxième). Pour passer d'une forme à une autre, on utilise la fonction **Transpose**:

```
Transpose[sPts]
transposée
```

```
{{x1, x2, x3, xn}, {y1, y2, y3, yn}}
```

```
Transpose[sao]
transposée
```

```
{{x1, y1}, {x2, y2}, {x3, y3}, {xn, yn}}
```

Utilisons la transposition pour former une liste de points:

```
a = Range[0, 9];
    plage
f[x_] := x^2 - 3;
sPts = Transpose[{a, f[a]}]
       transposée
```

```
{{0, -3}, {1, -2}, {2, 1}, {3, 6}, {4, 13}, {5, 22}, {6, 33}, {7, 46}, {8, 61}, {9, 78}}
```

Réciproquement, utilisons la transposition pour extraire de **sPts** la liste des ordonnées:

```
Transpose[sPts][[2]]
transposée
```

```
{-3, -2, 1, 6, 13, 22, 33, 46, 61, 78}
```

```
Clear[sPts, sao, a, f]
efface
```

3.3.6 Transformation d'une liste en arguments d'une fonction

La fonction **Plus** est une fonction à plusieurs arguments qui donne la somme des arguments:

```
Plus[2, 3, 4, 5]
plus
```

```
14
```

Cependant, lorsqu'on applique **Plus** à une liste, on n'obtient pas la somme des éléments car la fonction **Plus** n'est appliquée qu'à un seul argument:

```
Plus[{2, 3, 4, 5}]
plus
```

```
{2, 3, 4, 5}
```

La fonction **Apply** permet de transformer les éléments d'une liste en arguments d'une fonction. Par conséquent, **Apply[f,{a,b,c}]** équivaut à **f[a,b,c]**; par exemple,

```
Apply[Plus, {2, 3, 4, 5}]
rempl··· plus
```

```
14
```

Times est l'analogue de **Plus** pour la multiplication:

```
Times[2, 3, 4, 5]
multiplication
```

```
120
```

```
Apply[Times, {2, 3, 4, 5}]
rempl··· multiplication
```

```
120
```

Exemple: factorielle d'un nombre

Un ensemble de 3 éléments distincts $\{a, b, c\}$ peut être ordonné de 6 manières différentes :

abc, acb, bac, bca, cab, cba

On dit que la factorielle de 3 vaut 6, ce qu'on note $3! = 6$. On la calcule comme suit : $3! = 1 \cdot 2 \cdot 3$. D'une manière générale, à un ensemble de n éléments correspond $n!$ listes de longueur n. On construit la fonction "factorielle de n" qui, pour un naturel n donné, calcule le produit

$$1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots \cdot n$$

```
factorielle[n_] := Apply[Times, Range[n]]
```
rempl··· multipl··· plage

```
Table[{i, factorielle[i]}, {i, 10}]
```
table

```
{{1, 1}, {2, 2}, {3, 6}, {4, 24}, {5, 120},
 {6, 720}, {7, 5040}, {8, 40 320}, {9, 362 880}, {10, 3 628 800}}
```

```
Clear[factorielle]
```
efface

Exemple: moyenne arithmétique d'une liste

Pour calculer la moyenne arithmétique d'une liste, on divise la somme par le nombre d'éléments:

```
moyenne[x_List] := Apply[Plus, x] / Length[x]
```

```
moyenne[{1, 2, 3, 4}]
```

$$\frac{5}{2}$$

Le symbole `x_List` spécifie que la fonction **moyenne** doit être appliqué à une liste `x`.

```
Clear[moyenne];
```
efface

3.3.7 Fonctions non-listables

Certaines fonctions, lorsqu'elles sont appliquées à une liste, ne sont pas automatiquement distribuées à chaque élément de cette liste. On dit qu'elles ne sont pas "listables". La fonction `Circle` en est un exemple. Pour illustrer le problème, nous considérons l'exercice consistant à tracer des cercles de rayons 1 dont les coordonnées des centres sont répertoriées dans une liste :

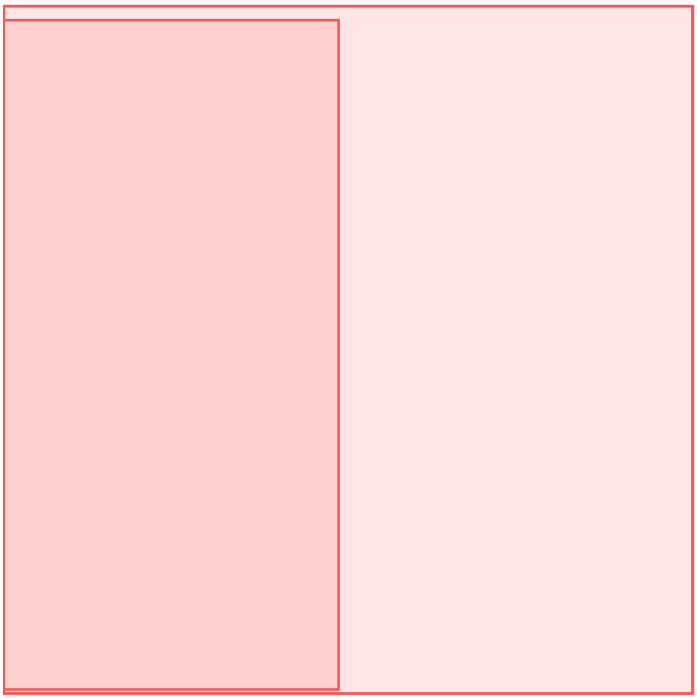
```
centres = Table[{x^2, x}, {x, 0, 5}];
```
table

```
Circle[centres]
```
cercle

```
Circle[{{0, 0}, {1, 1}, {4, 2}, {9, 3}, {16, 4}, {25, 5}}]
```

Nous constatons que la fonction `Circle` n'a pas été appliquée à chaque coordonnées. Il ne va donc pas être possible de dessiner ces cercle en procédant ainsi :

```
Graphics[%]
```
graphique

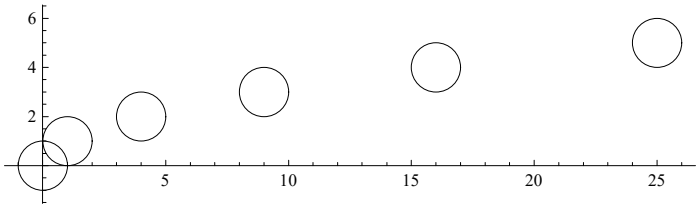


Pour remédier à ce problème, il existe l'opérateur `Map` : il permet d'appliquer une fonction à chaque élément d'une liste. Par défaut, la fonction est appliquée aux éléments du premier niveau de la liste. Utilisons `Map` pour tracer nos cercles :

```
Map[Circle, centres]
```
app· cercle

```
{Circle[{0, 0}], Circle[{1, 1}], Circle[{4, 2}],
 Circle[{9, 3}], Circle[{16, 4}], Circle[{25, 5}]}
```

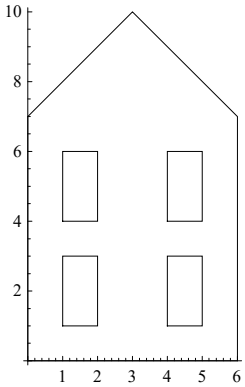
```
Graphics[%, Axes → True, AspectRatio → Automatic]
```
graphique axes vrai rapport d'aspect automatique




```
Clear[centres]
```
efface

- **Exemple: utilisation de `Map` pour faire une translation**

Nous voulons faire le dessin suivant:



On définit d'abord les coordonnées du contour et d'une fenêtre de la maison:

```
c = {{0, 0}, {0, 7}, {3, 10}, {6, 7}, {6, 0}, {0, 0}};
f1 = {{1, 1}, {1, 3}, {2, 3}, {2, 1}, {1, 1}};
```

Pour définir les sommets des trois autres fenêtres de la maison, on applique une translation à chaque sommet de la première fenêtre. Pour translater d'un vecteur t la ligne polygonale f1, on additionne t à chaque sommet x de f1:

```
trans[t_][x_] := x + t;
f2 = Map[trans[{0, 3}], f1];
```
applique

```
f3 = Map[trans[{3, 0}], f1];
```
applique

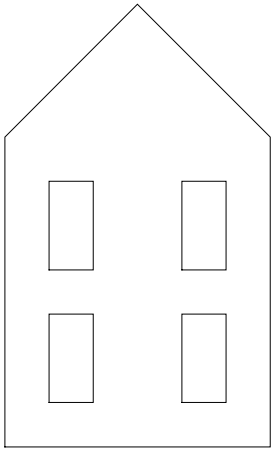
```
f4 = Map[trans[{3, 3}], f1];
```
applique

Expliquons ce qui précède

```
Map[trans[{3, 0}], f1]
    = Map[trans[{3, 0}], {{1, 1}, {1, 3}, ...}]
    = {trans[{3, 0}][{1, 1}], trans[{3, 0}][{1, 3}], ... }
    = {{1, 1} + {3, 0}, {1, 3} + {3, 0}, ...}
    = {{4, 1}, {4, 0}, ...}
```

Maintenant nous pouvons dessiner notre maison:

```
Graphics[Line[{c, f1, f2, f3, f4}]]
```
graphique ligne



```
Clear["Global`*"]
```
efface

Distinction entre Map et Apply

L'opérateur **Map** permet d'appliquer une fonction à chaque élément d'une liste tandis que **Apply** permet considérer les éléments d'un liste comme les arguments d'une fonction :

```
Apply[f, {a, b, c}]
```
remplace tête

```
Map[f, {a, b, c}]
```
applique

```
f[a, b, c]
```

```
{f[a], f[b], f[c]}
```

§ 3.4 Sélection d'un élément ou d'une liste d'éléments

Dans les exemples qui suivent, nous allons travailler avec les listes suivantes:

```
sArith = Table[3 + 5 i, {i, 0, 10}]
```
table

```
{3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53}
```

```
suite = Table[3 + 2^k, {k, 0, 10}]
```
table

```
{4, 5, 7, 11, 19, 35, 67, 131, 259, 515, 1027}
```

$$\text{sAngles} = \text{Table}\left[\frac{\pi}{4} + \frac{k\,\pi}{6}, \{k, -3, 3\}\right]$$
table

$$\left\{-\frac{\pi}{4}, -\frac{\pi}{12}, \frac{\pi}{12}, \frac{\pi}{4}, \frac{5\,\pi}{12}, \frac{7\,\pi}{12}, \frac{3\,\pi}{4}\right\}$$

$$\text{sCube} = \text{Table}\left[n^3, \{n, -2, 5\}\right]$$
table

```
{-8, -1, 0, 1, 8, 27, 64, 125}
```

3.4.1 Sélection avec Part

Nous avons vu que pour sélectionner l'élément numéro **i** d'une liste, on utilise le symbole **[[i]]** placé après le nom de la liste. Ceci peut aussi être fait avec la fonction **Part** comme le montrent les exemples suivants:

```
sArith[[4]]
```

```
18
```

```
Part[sArith, 4]
```
partie

```
18
```

Sélectionner un élément ne signifie pas l'ôter de la liste. Par exemple, la liste **sArith** n'a pas été modifiée:

```
?sArith
```

```
Global`sArith
```

```
sArith = {3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53}
```

On peut aussi sélectionner une sous-liste quelconque; par exemple, pour sélectionner les éléments numéros 3,5,6 d'une liste:

```
suite[[{3, 5, 6}]]
```

```
{7, 19, 35}
```

```
Part[suite, {3, 5, 6}]
partie
```

```
{7, 19, 35}
```

En utilisant **Range**, on peut facilement sélectionner plusieurs éléments consécutifs:

```
sCube[[Range[3, 6]]]
        plage
```

```
{0, 1, 8, 27}
```

On peut utiliser des indices négatifs:

-1 représente le dernier élément,
-2 l'avant-dernier élément, etc.

```
suite[[-1]]
```

```
1027
```

```
First[sAngles]
premier
```

$$-\frac{\pi}{4}$$

```
Last[sAngles]
dernier
```

$$\frac{3\pi}{4}$$

3.4.2 Sélection avec Select

La fonction **Select** permet de sélectionner les éléments d'une liste remplissant un certain critère: **Select[liste,critQ]** extrait de **liste** les éléments remplissant le critère **critQ**. Illustrons le fonctionnement de cette fonction à l'aide d'un exemple: supposons que nous voulons sélectionner les éléments positifs d'une liste. Pour cela, nous commençons par définir une fonction à valeur booléenne (**True** ou **False**) **positifQ** qui retourne **True** lorsque son argument est strictement positif:

```
positifQ[x_] := x > 0
```

```
positifQ[-7]
```

```
False
```

```
positifQ[8]
```

```
True
```

Sélectionnons maintenant les nombres positifs de **sCube** à l'aide de **Select**:

```
Select[sCube, positifQ]
sélectionne
```

```
{1, 8, 27, 64, 125}
```

Sont sélectionnés tous les éléments de la liste **sCube** pour lesquels **positifQ** est vrai.

Pour sélectionner tous les angles de **sAngles** dont le cosinus est négatif, nous définissons un critère **cosNeg**:

```
cosNegQ[x_] := Cos[x] < 0;
               cosinus
```

```
Select[sAngles, cosNegQ]
sélectionne
```

$$\left\{\frac{7\pi}{12}, \frac{3\pi}{4}\right\}$$

Une liste de critères est disponible dans *Mathematica*. Il s'agit de fonctions dont le nom se termine toujours par **Q** (ce qui signifie Question). Voici quelques exemples:

- **OddQ** teste si l'expression est un nombre impaire;
- **EvenQ** teste si l'expression est un nombre paire;
- **PrimeQ** teste si l'expression est un nombre premier;
- **IntegerQ** teste si l'expression est un nombre entier.

```
Select[sArith, OddQ]
sélectionne  nombre
```

```
{3, 13, 23, 33, 43, 53}
```

```
Select[sArith, EvenQ]
sélectionne  nombre p
```

```
{8, 18, 28, 38, 48}
```

```
Select[sArith, PrimeQ]
sélectionne  nombre pre
```

```
{3, 13, 23, 43, 53}
```

```
Select[sAngles, IntegerQ]
sélectionne  entier?
```

```
{}
```

Exemple: extraction des multiples de 3

Comme pour les équations, la condition d'égalité s'écrit **==** (à ne pas confondre avec symbole d'affectation immédiate **=** et le symbole d'affectation différée **:=**).

```
2 == 1 + 1
```

```
True
```

```
3 == 1 + 1
```

```
False
```

Le critère "x est un multiple de 3" peut s'écrire:

```
mult3Q[x_] := Mod[x, 3] == 0
              modulo mod
```

```
mult3Q[14]
```

```
False
```

```
mult3Q[15]
```

```
True
```

Les multiples de 3 de **sArith** sont:

```
Select[sArith, mult3Q]
sélectionne
```

```
{3, 18, 33, 48}
```

Dans un critère,

le "et" logique se note && au clavier, ce qui équivaut au ∧ de la palette;
le "ou" logique se note || au clavier (touches [Alt Gr]+[7])
ce qui équivaut au ∨ de la palette.

Le critère "x est un nombre pair supérieur à 20" peut s'écrire:

```
critQ[x_] := EvenQ[x] ∧ x > 20
             nombre pair?
```

```
Select[sArith, critQ]
sélectionne
```

```
{28, 38, 48}
```

```
Clear["Global`*"]
efface
```

Exemple: triplets de Pythagore (version avec `Select`)

On veut dresser une partie de la liste des triplets de Pythagore, plus précisément la liste des triplets (a, b, c) tels que

a, b, c sont des entiers positifs,

$a^2 + b^2 = c^2$ et

$1 \leq a < b \leq 12$.

Nous allons effectuer le calcul en deux étapes. Dans une première partie, en n'utilisant qu'une partie des conditions à remplir, nous allons former une liste de candidats :

```
candidats = Flatten[Table[{a, b, Sqrt[a^2 + b^2]}, {a, 1, 11}, {b, a + 1, 12}], 1];
            aplatis  table
```

Les candidats vérifient certaines conditions requises, mais pas toutes. La deuxième partie du calcul est un filtrage. Nous ne retenons que les candidats dont la troisième composante est entière:

```
critQ[x_] := IntegerQ[x[[3]]]
             entier?

TableForm[
forme de table
 Select[candidats, critQ],
 sélectionne
 TableHeadings -> {None, {"a", "b", "c"}}
                  aucun
]
```

a	b	c
3	4	5
5	12	13
6	8	10
9	12	15

La dernière condition peut aussi s'écrire sous la forme

```
Clear[critQ]
efface

critQ[{a_, b_, c_}] := IntegerQ[c]
                       entier?

Select[candidats, critQ]
sélectionne

{{3, 4, 5}, {5, 12, 13}, {6, 8, 10}, {9, 12, 15}}

Clear[candidats, critQ]
efface
```

3.4.3 Sélection de sous-listes avec `Cases` (Facultatif)

Les intéressés peuvent se rapporter aux annexes se trouvant sur la page internet du cours (rubrique “Documents de cours au format Mathematica, annexes”)

§ 3.5 Listes comme ensembles (facultatif)

Une liste (ordonnée) peut être utilisée pour décrire un ensemble (non ordonné). Les intéressés peuvent se rapporter aux annexes se trouvant sur la page internet du cours (rubrique “Documents de cours au format Mathematica, annexes”)

§ 3.6 Modification d’une liste (facultatif)

Il est également possible de modifier une liste existante, par exemple en changeant l'un de ses éléments ou en ajoutant des éléments.

```
maliste = Range[5]
          plage

maliste[[3]] = 10
maliste

{1, 2, 3, 4, 5}

10

{1, 2, 10, 4, 5}
```

```
Clear[maliste]
```
efface

Pour ajouter des éléments à une liste on dispose des commandes suivantes: **Prepend** pour un ajout au début de la liste, **Append** pour un ajout à la fin et Insert pour insérer un élément à un endroit spécifique.

§ 3.7 Exercices

■ Exercice 3-1

Créer les listes suivantes au moyen de la fonction **Table**:

$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

$\left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}, 2\pi\right\}$

$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}, \frac{1}{49}, \frac{1}{64}, \frac{1}{81}, \frac{1}{100}\right\}$

$\left\{\frac{2}{3}, \frac{4}{5}, \frac{6}{7}, \frac{8}{9}, \frac{10}{11}, \frac{12}{13}, \frac{14}{15}\right\}$

$\{\{0, x, 2x, 3x, 4x\}, \{0, x^2, 2x^2, 3x^2, 4x^2\}, \ldots, \{0, x^{10}, 2x^{10}, 3x^{10}, 4x^{10}\}\}$

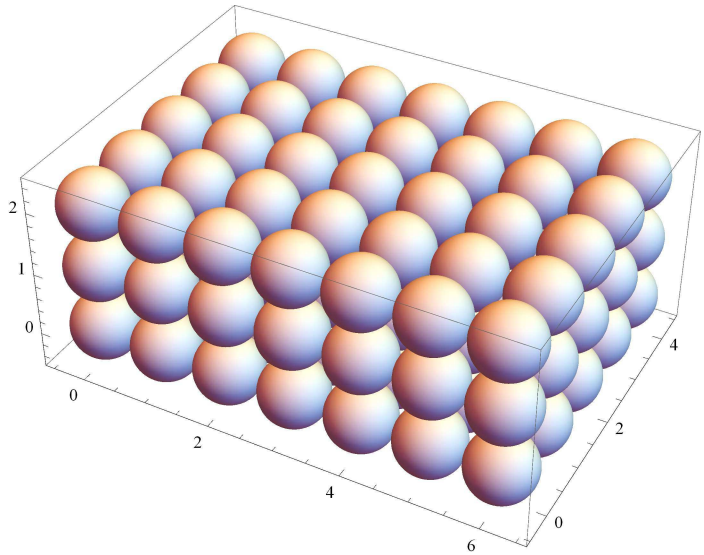
Créer le tableau ci-dessous.

x	y
1	1
2	$\frac{1}{2}$
3	$\frac{1}{3}$
4	$\frac{1}{4}$
5	$\frac{1}{5}$
6	$\frac{1}{6}$
7	$\frac{1}{7}$
8	$\frac{1}{8}$
9	$\frac{1}{9}$
10	$\frac{1}{10}$

■ Exercice 3-2

Dessiner, le graphique ci-dessous.

Indication: commencer par créer un liste de sphères à l'aide de la fonction **Sphere**.



Exercice 3-3

Créer les listes suivantes au moyen de la fonction **NestList**:

a) {1, -1, 1, -1, 1, -1, ...} (14 éléments);

b) {3, 9, 81, 6561, 43046721, ...} (8 éléments);

c) $\{16, 4, 2, \sqrt{2}, ...\}$ (12 éléments).

Exercice 3-4

Dans le but de résoudre l'équation

$x - \cos(x) = 0$ où x est exprimé en radians

on construit la liste

$$x_0 = \frac{\pi}{4.};\ x_1 = \cos(x_0);\ x_2 = \cos(x_1);\ x_3 = \cos(x_2);\ ...$$

a) Calculer les termes successifs de cette suite et montrer (numériquement) qu'elle atteint un point fixe.

b) Démontrer (mathématiquement) que tout point fixe de la suite est une solution de l'équation.

Cette méthode qui permet de résoudre certaines équations est appelée "méthode d'itération du point fixe". On peut vérifier aisément que les commandes Reduce et Solve ne permettent pas de résoudre l'équation cos(x) = x. Dans le chapitre "Résolution d'équations" nous allons nous intéresser de manière plus approfondie aux méthodes de résolution alternatives.

Reprenons maintenant l'exemple **FixedPointList[Sqrt, 1.5]** donné dans la théorie:

c) Sans utiliser l'ordinateur, écrire sous la forme mathématique la suite de nombres ainsi construite.

d) Quelle est l'équation dont la méthode d'itération donnée produit une solution ?
L'équation possède-t-elle d'autres solutions ?

Exercice 3-5

Sans utiliser *Mathematica*, déterminer le résultat de l'évaluation du code ci-dessous en effectuant, si nécessaire, des corrections.

```
Clear["Global`*"]
efface
f[x_] = x^2
NestList[f[x], 2, 3]
liste d'imbrication
FixedPointList[f[x], 1/2]
liste de point fixe
```

Exercice 3-6

Construire les suites **sArith**, **suite**, **sCube** et **sAngles** ci-dessous au moyen de la fonction **Range** au lieu de **Table**.
Exemple: la liste **Table[3 k, {k, 0, 15}]** peut aussi être générée avec **Range[0, 45, 3]**

```
sArith = Table[3 + 5 i, {i, 0, 10}];
         table
sCube = Table[n^3, {n, -2, 5}];
        table
sAngles = Table[π/4 + k π/6, {k, -3, 3}];
          table
suite = Table[3 + 2^k, {k, 0, 10}];
        table
```

Exercice 3-7

Dans la liste suivante, déterminer la première expression dont la valeur est négative

$$\mathrm{Cos}\left[k\,\frac{2\pi}{17}\right],\ k \in [0, 17].$$

Indication : former une liste de valeurs : Cos[0], $\mathrm{Cos}\left[\frac{2\pi}{17}\right]$, $\mathrm{Cos}\left[2\cdot\frac{2\pi}{17}\right]$,...
déterminer les signes de chaque terme à l'aide de la fonction **Sign**;
déterminer la position des -1;
déterminer la position du premier -1;
écrire l'expression correspondante.

Exercice 3-8

Considérons les listes

```
lt = Table[(-1)^i, {i, -5, 15}];
     table
```

```
ls = Table[Sin[k π/5 + π/10], {k, -5, 15}];
     table  sinus
```

a) Calculer le nombre d'occurrences de l'élément **1** dans les listes **lt** et **ls**.

b) Déterminer la position de chaque **1** dans **lt** et **ls**.

Exercice 3-9

Simuler 100 jets d'une pièce de monnaie à l'aide de la fonction **RandomChoice** et calculer la fréquence relative (nombre d'apparitions divisé par nombre total de jets) de pile et de face. Procéder de telle manière qu'un changement du nombre de jets ne doive s'effectuer qu'en un seul endroit.

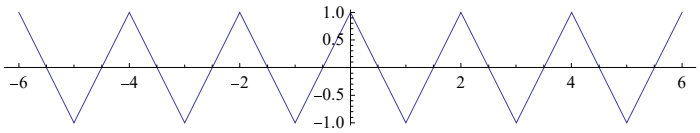
Exercice 3-10

a) Représenter graphiquement la ligne polygonale (ouverte) de sommets:

```
Table[{t Cos[t], t Sin[t]}, {t, 0, 30, 0.5}];
table        cosinus    sinus
```

b) Reproduire la ligne polygonale ci-dessous sans le système d'axe.

Indication: commencer par générer une liste contenant les sommets de la ligne.



Exercice 3-11

On jette deux fois un dé en forme de dodécaèdre régulier dont les faces sont numérotées de 1 à 12 et on somme les deux nombres obtenus.

Dresser un tableau contenant toutes les sommes possibles ainsi que le nombre de combinaisons permettant d'y arriver. Indication: commencer par générer la liste contenant la somme obtenue pour chacun des 24 jets possibles.

Exercice 3-12

Afficher la liste des 10 premiers nombres premiers:

a) en un tableau sur deux lignes:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 & 23 & 29 \end{pmatrix}$$

b) en un tableau sur deux colonnes.

Exercice 3-13

En fonction de la température en degrés Celsius 0, 1, ..., 119, on donne les **pressions de saturation de la vapeur d'eau** correspondantes, en pascals (voir "*Formulaires et tables*"):

```
psat = {611, 657, 706, 758, 813, 872, 935, 1002, 1073, 1148, 1228, 1312, 1402, 1497, 1598, 1705,
   1818, 1937, 2063, 2197, 2338, 2487, 2643, 2809, 2983, 3167, 3360, 3564, 3780, 4005,
   4243, 4492, 4755, 5030, 5319, 5623, 5941, 6275, 6625, 6992, 7375, 7778, 8199, 8639,
   9101, 9583, 10086, 10612, 11160, 11735, 12334, 12959, 13611, 14292, 15000, 15737,
   16505, 17308, 18143, 19012, 19916, 20856, 21834, 22849, 23906, 25003, 26143,
   27326, 28554, 29828, 31157, 32517, 33944, 35424, 36957, 38543, 40183, 41877,
   43636, 45463, 47343, 49289, 51316, 53409, 55569, 57809, 60115, 62488, 64941,
   67474, 70096, 72801, 75592, 78474, 81477, 84513, 87675, 90935, 94295, 97757,
   101325, 105000, 108772, 112673, 116665, 120799, 125046, 129403, 133912, 138511,
   143263, 148148, 153153, 158310, 163620, 169050, 174644, 180378, 186275, 192335};
```

La liste ci-dessus se trouve dans le fichier **pSatVapeurEau.nb qui** est téléchargeable depuis la page internet du cours dans la rubrique “Documents de cours au format Mathematica/Annexes” (http://applmaths.collegedusud.ch).

a) Former la liste des points de mesures {{0, 611}, ..., {119, 192335}} et dessiner le graphique de la pression de saturation de l’eau en fonction de la température. Annoter les axes.

b) Former la liste des points de mesures {{611, 0}, ..., {192335, 119}} et dessiner le graphique de la température d’ébullition de l’eau en fonction de la pression en annotant les axes.

c) A partir de **psat**, former un tableau dans lequel chaque ligne est constituée de 10 éléments

```
tabl={{611, ..1148}, {1228, ...}, ..., {..., 192335}}
```

Présenter les pressions de saturation sous la forme du tableau donné ci-dessous.

	0	1	2	3	4	5	6	7	8	9
0	611	657	706	758	813	872	935	1002	1073	1148
10	1228	1312	1402	1497	1598	1705	1818	1937	2063	2197
20	2338	2487	2643	2809	2983	3167	3360	3564	3780	4005
30	4243	4492	4755	5030	5319	5623	5941	6275	6625	6992
40	7375	7778	8199	8639	9101	9583	10086	10612	11160	11735
50	12334	12959	13611	14292	15000	15737	16505	17308	18143	19012
60	19916	20856	21834	22849	23906	25003	26143	27326	28554	29828
70	31157	32517	33944	35424	36957	38543	40183	41877	43636	45463
80	47343	49289	51316	53409	55569	57809	60115	62488	64941	67474
90	70096	72801	75592	78474	81477	84513	87675	90935	94295	97757
100	101325	105000	108772	112673	116665	120799	125046	129403	133912	138511
110	143263	148148	153153	158310	163620	169050	174644	180378	186275	192335

Exercice 3-14

Calculer la somme et le produit de chacune des listes **sArith**, **suite**, **sCube** et **sAngles** de l’exercice 3-6.

Exercice 3-15

Définir une fonction **moyenneGeo** qui retourne la moyenne géométrique, calculée avec 5 chiffres significatifs, d’une liste de nombres (la moyenne géométrique de n nombres $x_1, x_2, ..., x_n$ est égale à $\sqrt[n]{x_1 \cdot x_2 \cdot ... \cdot x_n}$), puis calculer la moyenne géométrique de chacune des listes **sArith**, **suite**, **sCube** et **sAngles** de l’exercice 3-6.

Exercice 3-16

a) Calculer les valeurs de la fonction $f(x) = \dfrac{x-1}{2x+1}$ aux abscisses entières entre -10 et 10.

Prescription d’exercice: utiliser **Map**.

b) Afficher la liste des points correspondants.

c) Représenter le graphe de la fonction f en y ajoutant les points dont les coordonnées ont été créées dans la partie b).

Exercice 3-17

Sans utiliser *Mathematica*, déterminer les résultats retournés lors de l'évaluation des instructions suivantes:

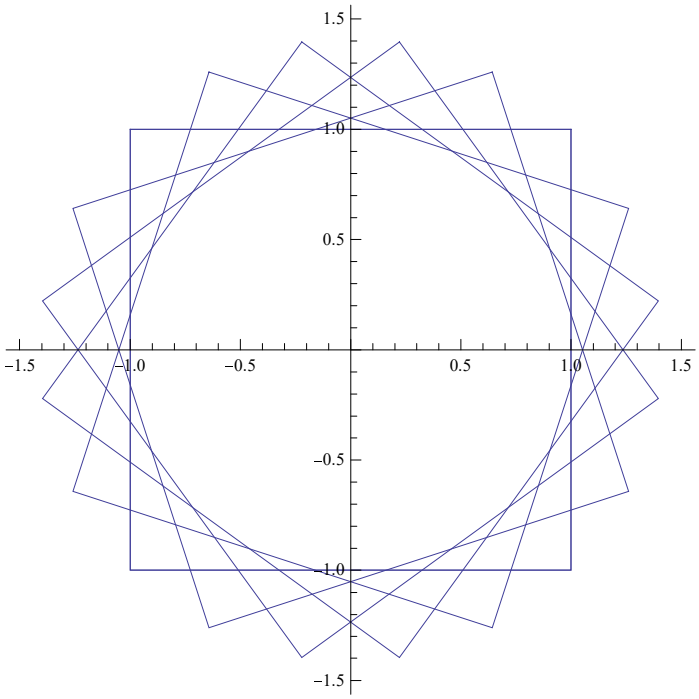
```
Clear["Global`*"]
```
efface

```
liste = {{2, 3}, {7, 11}};
f[x_, y_] := x + y
f[{x_, y_}] := x * y
f[liste]
Map[f, liste]
```
applique

```
Apply[f, liste]
```
remplace tête

Exercice 3-18

La figure suivante peut être obtenue à partir d'un carré dont les côtés sont parallèles aux axes et auquel on a fait subir des rotations de 18°, 36°, 54°, 72° et 90°.



Effectuer les opérations suivantes dans le but de reproduire cette figure:

a) Définir un liste **carre** contenant les sommets du carré initial.

b) Dessiner le carré initial ayant subi une rotation de 18°. Pour ce faire, utiliser la fonction **rot** définie ci-dessous qui fait effectuer à un point de coordonnée (x, y) une rotation d'angle α autour de l'origine.

```
rot[α_][{x_, y_}] := {x Cos[α] - y Sin[α], x Sin[α] + y Cos[α]}
```
cosinus sinus sinus cosinus

```
rot[90 °][{1, 0}]
```

```
{0, 1}
```

```
rot[180 °][{1, 3}]
```

```
{-1, -3}
```

c) Faire une liste des cinq dessins de carré et les faire s'afficher dans le même repère.

Exercice 3-19

Étant donnée la liste

```
Table[{i, (-1)^i}, {i, -6, 6}]
table
```

```
{{-6, 1}, {-5, -1}, {-4, 1}, {-3, -1}, {-2, 1},
 {-1, -1}, {0, 1}, {1, -1}, {2, 1}, {3, -1}, {4, 1}, {5, -1}, {6, 1}}
```

a) sélectionner le 3e élément;
sélectionner l'avant-dernier élément;

b) à quelle(s) position(s) se trouve(nt) les éléments {-1,-1} et {1,-1} ?

Exercice 3-20

On donne la liste

```
liste = Table[{x, Cos[x]}, {x, -6, 6}];
        table      cosinus
```

Former les sous-ensembles suivants:

lp1 = ensemble des points à l'intérieur du premier quadrant;
lp2 = ensemble des points à l'intérieur du deuxième quadrant;
lp3 = ensemble des points à l'intérieur du troisième quadrant;
lp4 = ensemble des points à l'intérieur du quatrième quadrant.

Exercice 3-21

Parmi les nombres naturels inférieurs à 1000, quels sont ceux qui sont des carrés parfaits et qui sont de la forme $4\,n + 1$ avec n entier?

Indication: former la liste des carrés parfaits inférieurs à 1000; puis, dans cette liste, sélectionner les nombres dont le reste de la division par 4 donne 1.

§ 3.8 Exercices de récapitulation

Les exercices qui suivent sont des exemples de questions d'examen sur le chapitre "Listes".

Exercice 3-22

Construire les listes **a**,**b**,**c**,**d** et les sommes **e**, **f** suivantes:

$$a = \left\{x,\ x^2,\ x^3,\ \ldots,\ x^{36}\right\}$$

$$b = \left\{1,\ \frac{1}{2},\ \frac{1}{3},\ \ldots,\ \frac{1}{36}\right\}$$

$$c = \{1,\ -1,\ 1,\ -1,\ \ldots\} \qquad \text{(liste de 36 termes)}$$

$$d = \left\{x,\ -\frac{x^2}{2},\ \frac{x^3}{3},\ -\frac{x^4}{4},\ \ldots,\ -\frac{x^{36}}{36}\right\}$$

$$e = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots - \frac{x^{36}}{36}$$

$$f = 0.8 - \frac{0.8^2}{2} + \frac{0.8^3}{3} - \frac{0.8^4}{4} + \ldots - \frac{0.8^{36}}{36}$$

Exercice 3-23

On considère la suite de valeurs numériques en virgule flottante

$$\left\{0,\ \frac{1.}{2.+0},\ \frac{1.}{2.+\frac{1.}{2.+0}},\ \frac{1.}{2.+\frac{1.}{2.+\frac{1.}{2.+0}}},\ \frac{1.}{2.+\frac{1.}{2.+\frac{1.}{2.+\frac{1.}{2.+0}}}},\ \ldots\right\}$$

a) Définir la fonction qui, étant donné un terme x de la suite, donne le terme suivant "successeur de x".

b) Construire la liste des 36 premiers termes de la suite.

c) Calculer le point fixe de la fonction "successeur".

Exercice 3-24

a) Construire la liste

$$\text{rayons} = \left\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots, \frac{1}{512}\right\}$$

b) A partir de la liste précédente et de la fonction

```
cercle[r_] := Circle[{1-r, 0},r]
```

construire la famille de cercles

$$\text{famCercles} = \left\{\text{Circle}[\{0, 0\}, 1], \text{Circle}\left[\left\{1 - \frac{1}{2}, 0\right\}, \frac{1}{2}\right], \ldots, \text{Circle}\left[\left\{1 - \frac{1}{512}, 0\right\}, \frac{1}{512}\right]\right\}$$

c) Dessiner la figure correspondante.

Exercice 3-25

Dessiner une vue de profil d'un gratte-ciel à base carré de **n=20** étages, chaque étage ayant sur chaque face **m=10** fenêtres. Les fenêtres auront une largeur **l=1** unité et une hauteur **h=1.5** unité. L'écart horizontale entre les fenêtres sera **l** et l'écart vertical **h**. Les fenêtres seront noires tandis que le reste de la façade sera gris (pour cela travailler avec la fonction **Polygon**).

Exercice 3-26

On donne la fonction

$$f(x) = \frac{x^3 - 5}{x}$$

a) Dresser les trois listes suivantes

les abscisses $x = \left\{-\frac{3}{4}, -\frac{1}{4}, \frac{1}{4}, \frac{3}{4}, \ldots, 10 + \frac{1}{4}\right\}$;

les ordonnées $y = \left\{f\left[-\frac{3}{4}\right], f\left[-\frac{1}{4}\right], f\left[\frac{1}{4}\right], f\left[\frac{3}{4}\right], \ldots, f\left[10 + \frac{1}{4}\right]\right\}$;

les points $p = \left\{\left\{-\frac{3}{4}, f\left[-\frac{3}{4}\right]\right\}, \left\{-\frac{1}{4}, f\left[-\frac{1}{4}\right]\right\}, \ldots, \left\{10 + \frac{1}{4}, f\left[10 + \frac{1}{4}\right]\right\}\right\}$.

b) Calculer les valeurs maximale et minimale de la liste y
puis déterminer les abscisses correspondantes.

c) Dessiner la ligne polygonale qui joint les points de p.

d) Sélectionner de y les valeurs qui sont positives
puis former la liste des points correspondants
et afficher ces points sous la forme d'un tableau de nombres en virgule flottante.

Exercice 3-27

On jette 3 fois un dé cubique dont les faces sont numérotés de 1 à 6. On somme les trois nombres sortis.

a) Générer un liste **jets** contenant les 216 résultats possibles.

b) A partir de **jets**, construire un tableau contenant pour chaque somme le nombre de façons de l'obtenir.

c) Définir une fonction **somme[n_]** qui retourne une liste contenant toutes les façons d'obtenir une somme de n.

Exercice 3-28

Un nombre parfait est un entier naturel égal à la somme de ses diviseurs autres que lui-même. Ceci est équivalent à dire que le double d'un nombre parfait est égale à la somme de tous ses diviseurs. Ainsi 6 est un nombre parfait car $6 = 1 + 2 + 3$ (ou, de façon équivalente, car $2 \cdot 6 = 1 + 2 + 3 + 6$).

Dresser la liste des nombres parfaits inférieurs à 1000.

Indication: utiliser la fonction **Divisors**.

§3-9 Réponses

Exercice 3-4

a)

```
{0.785398, 0.707107, 0.760245, 0.724667, 0.74872, 0.732561, 0.743464, 0.736128,
 0.741074, 0.737744, 0.739988, 0.738477, 0.739495, 0.738809, 0.739271, 0.73896,
 0.739169, 0.739028, 0.739123, 0.739059, 0.739103, 0.739073, 0.739093, 0.73908,
 0.739089, 0.739083, 0.739087, 0.739084, 0.739086, 0.739085, 0.739085, ...}
```

c) $x_n = \sqrt[2^n]{1.5}$

d) L'équation est $\sqrt{x} = x$; autre solution : $x = 0$.

Exercice 3-5

Après correction {2, 4, 16, 256} et {0.5,0.25,0.0625,...,0} (en réalité *Mathematica* ne retourne par 0 mais **Underflow[]** ce qui correspond à un nombre trop proche de zéro pour que *Mathematica* puisse calculer avec).

Exercice 3-7

$\cos\left(5 \cdot \frac{2\pi}{17}\right) \approx -0.273663$.

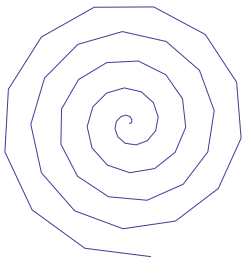
Exercice 3-8

a) pour **ls** 2 et pour **lt** 10

b) pour **ls** positions 8 et 18, pour **lt** positions 2, 4, 6, ...,20

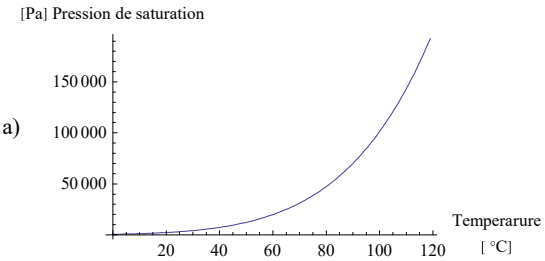
Exercice 3-10

a)

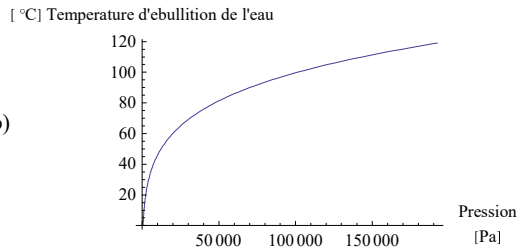


Exercice 3-13

a)



b)



Exercice 3-14

Pour **sArith**: 308, 496130658895872, 21.675

Pour **suite** : 2080, 1231305732007851500, 44.114

Pour **sCube** : 219, 0, 0

Pour **sAngles** : $\frac{7\pi}{4}$, $\frac{35\pi^7}{442368}$, 0.81506

Exercice 3-15

Pour **sArith**: 21.675

Pour **suite** : 44.114

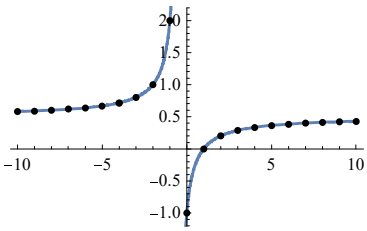
Pour **sCube** : 0

Pour **sAngles** : 0.81506

Exercice 3-16

$$\left\{\frac{11}{19}, \frac{10}{17}, \frac{3}{5}, \frac{8}{13}, \frac{7}{11}, \frac{2}{3}, \frac{5}{7}, \frac{4}{5}, 1, 2, -1, 0, \frac{1}{5}, \frac{2}{7}, \frac{1}{3}, \frac{4}{11}, \frac{5}{13}, \frac{2}{5}, \frac{7}{17}, \frac{8}{19}, \frac{3}{7}\right\}$$

$$\left\{\left\{-10, \frac{11}{19}\right\}, \left\{-9, \frac{10}{17}\right\}, \left\{-8, \frac{3}{5}\right\}, \left\{-7, \frac{8}{13}\right\}, \left\{-6, \frac{7}{11}\right\}, \left\{-5, \frac{2}{3}\right\}, \left\{-4, \frac{5}{7}\right\}, \left\{-3, \frac{4}{5}\right\}, \{-2, 1\}, \{-1, 2\}, \{0, -1\}, \{1, 0\}, \left\{2, \frac{1}{5}\right\}, \left\{3, \frac{2}{7}\right\}, \left\{4, \frac{1}{3}\right\}, \left\{5, \frac{4}{11}\right\}, \left\{6, \frac{5}{13}\right\}, \left\{7, \frac{2}{5}\right\}, \left\{8, \frac{7}{17}\right\}, \left\{9, \frac{8}{19}\right\}, \left\{10, \frac{3}{7}\right\}\right\}$$



Exercice 3-17

{14,33}, {6,77} et {9,14}

Exercice 3-19

a) {-4,1}, {5,-1}

b) 6, 8

Exercice 3-21

{1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441, 529, 625, 729, 841, 961}

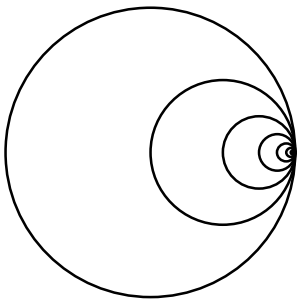
Exercice 3-23

b)

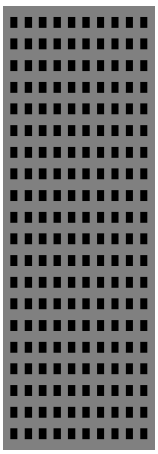
{0, 0.5, 0.4, 0.416667, 0.413793, 0.414286, 0.414201, 0.414216, 0.414213, 0.414214, ..., 0.414214}

c) 0.414214

Exercice 3-24



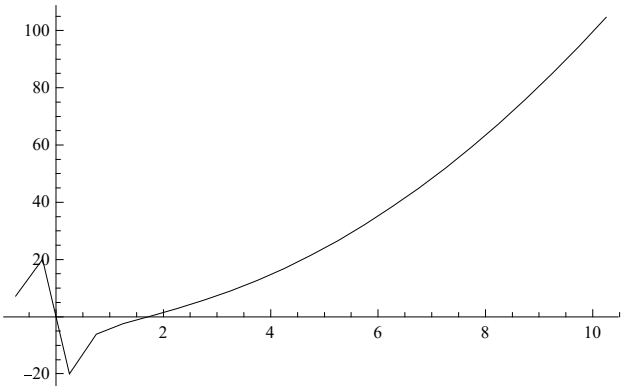
Exercice 3-25



Exercice 3-26

b) Maximum de $\frac{68\,601}{656}$ en l'abscisse $\frac{41}{4}$, minimum de $-\frac{319}{16}$ en l'abscisse $\frac{1}{4}$.

c)



d)

x	y
-0.75	7.22917
-0.25	20.0625
1.75	0.205357
2.25	2.84028
2.75	5.74432
3.25	9.02404
3.75	12.7292
4.25	16.886
4.75	21.5099
5.25	26.6101
5.75	32.1929
6.25	38.2625
6.75	44.8218
7.25	51.8728
7.75	59.4173
8.25	67.4564
8.75	75.9911
9.25	85.022
9.75	94.5497
10.25	104.575

Exercice 3-27

b)

somme	nb. possibilités
1	0
2	0
3	1
4	3
5	6
6	10
7	15
8	21
9	25
10	27
11	27
12	25
13	21
14	15
15	10
16	6
17	3
18	1

Exercice 3-28

6, 28 et 496