Marcel Délèze
Edition 2017

§ 3 Listes

Exercice 3-0

Lisez attentivement le présent cahier et exécutez-en tous les inputs.

§ 3.0 Notion de liste

La liste est une structure de données très courante. Pour définir des listes, on utilise des accolades { }. Une liste est ordonnée et peut contenir des éléments répétés :

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

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

```
eVide = {}
```

```
{}
```

```
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 :

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

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

Les éléments d'une liste peuvent être de natures différentes et être eux-mêmes des listes:

```
Clear[a, b, c, x];
```
efface

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

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

Une liste peut représenter diverses structures de données: un vecteur, un ensemble, un tableau, une matrice, un arbre, etc.

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 -.

§ 3.1 Construction de listes

Construction de listes avec Table[expr, {i, imin, imax}]

La fonction Table[] permet de créer des listes:

```
? Table
```

Table[*expr*, *n*] generates a list of *n* copies of *expr*.
Table[*expr*, {*i*, i_{max}}] generates a list of the values of *expr* when *i* runs from 1 to i_{max}.
Table[*expr*, {*i*, i_{min}, i_{max}}] starts with $i = i_{min}$.
Table[*expr*, {*i*, i_{min}, i_{max}, *di*}] uses steps *di*.
Table[*expr*, {*i*, {i_1, i_2, …}}] uses the successive values i_1, i_2, ….
Table[*expr*, {*i*, i_{min}, i_{max}}, {*j*, j_{min}, j_{max}}, …] gives a nested list. The list associated with *i* is outermost. >>

Voici quelques exemples:

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

table

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

$$\texttt{suite = Table}\left[3 + 2^k, \{k, 0, 10\}\right]$$

table

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

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

table

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

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

table

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

$$\texttt{f[x\_] := } x^2 - 5$$

```
ptsF = Table[{x, f[x]}, {x, -3, 3}]
```

table

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

Dans une suite arithmétique, "incrément" est un synonyme de "raison" : c'est le nombre constant que l'on ajoute à un terme pour obtenir le terme suivant.
L'itérateur {x, -3, 5, $\frac{3}{2}$} donne à x les valeurs successives de la suite arithmétique de premier terme -3, d'incrément $\frac{3}{2}$ et comprises entre -3 et 5 inclusivement.
L'itérateur {x, -3, 5} incrémente x de 1 à chaque pas (par défaut, l'incrément est égal à 1).

$$\texttt{ptsF = Table}\left[\{x, f[x]\}, \left\{x, -3, 5, \frac{3}{2}\right\}\right]$$

table

$$\left\{\{-3, 4\}, \left\{-\frac{3}{2}, -\frac{11}{4}\right\}, \{0, -5\}, \left\{\frac{3}{2}, -\frac{11}{4}\right\}, \{3, 4\}, \left\{\frac{9}{2}, \frac{61}{4}\right\}\right\}$$

Présentation de listes sous la forme de tableaux

Une liste peut être affichée à l'écran sous différentes formes. Présenter une liste sous la forme d'un tableau permet souvent d'augmenter la lisibilité

```
TableForm[ptsF]
```

forme de table

-3	4
$-\frac{3}{2}$	$-\frac{11}{4}$
0	-5
$\frac{3}{2}$	$-\frac{11}{4}$
3	4
$\frac{9}{2}$	$\frac{61}{4}$

Il est possibe d'y adjoindre des en-têtes de colonnes. Découvrez d'autres possibilités en consultant l'aide.

```
TableForm[ptsF, TableHeadings → {None, {"x", "f(x)"}}]
```

forme de table en–têtes de table aucun

x	f(x)
-3	4
$-\frac{3}{2}$	$-\frac{11}{4}$
0	-5
$\frac{3}{2}$	$-\frac{11}{4}$
3	4
$\frac{9}{2}$	$\frac{61}{4}$

Exercice 3-1-1

Créez les listes suivantes au moyen de la fonction Table[] :

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

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

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

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

$$\left\{\{1, 1\}, \left\{2, \frac{1}{2}\right\}, \left\{3, \frac{1}{3}\right\}, \left\{4, \frac{1}{4}\right\}, \left\{5, \frac{1}{5}\right\}, \left\{6, \frac{1}{6}\right\}, \left\{7, \frac{1}{7}\right\}, \left\{8, \frac{1}{8}\right\}, \left\{9, \frac{1}{9}\right\}, \left\{10, \frac{1}{10}\right\}\right\}$$

puis affichez le tableau

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	$\frac{1}{10}$

Construction de listes avec Range[imin, imax]

```
? Range
```

Range[i_{max}] generates the list {1, 2, …, i_{max}}.
Range[i_{min}, i_{max}] generates the list {i_{min}, …, i_{max}}.
Range[i_{min}, i_{max}, di] uses step di. »

La fonction **Range[initial, final]** équivaut à **Table[i, {i, initial, final}]** :

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

plage

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

Range[initial, final, pas] permet de construire n'importe quelle suite arithmétique :

```
Range[99, 9, -5]
```

plage

```
{99, 94, 89, 84, 79, 74, 69, 64, 59, 54, 49, 44, 39, 34, 29, 24, 19, 14, 9}
```

Construction de listes avec NestList[f, x, n] = {x, f[x], f[f[x]], ...}

```
? NestList
```

NestList[f, $expr$, n] gives a list of the results of applying f to $expr$ 0 through n times. »

Dans certaines listes, on passe d'un élément à l'élément suivant en appliquant toujours la même fonction **f**. Cette propriété peut être mise à profit pour créer des listes.

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

```
Clear[successeur];
```
efface
```
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 une suite géométrique de premier terme 2, de raison 3, qui comporte 11 termes:

```
Clear[successeur];
```
efface
```
successeur[x_] := 3 x;
NestList[successeur, 2, 10]
```
liste d'imbrication
```
{2, 6, 18, 54, 162, 486, 1458, 4374, 13122, 39366, 118098}
```

Exercice 3-1-2

Créez les listes suivantes au moyen de 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).

Construction de listes avec FixedPointList

```
? FixedPointList
```

FixedPointList[*f*, *expr*] generates a list giving the results
of applying *f* repeatedly, starting with *expr*, until the results no longer change. >>

Le principe de construction de listes (l'expression indique quel est le premier élément; la fonction f indique comment calculer le successeur de chaque élément) est le même pour les commandes **NestList[...]** et **FixedPointList[...]**. Ce qui change, c'est la longueur de la liste.

Dans **NestList[f, expr, n]**, on doit spécifier à l'avance la longueur de la liste; elle sera de $(n + 1)$ éléments.

Dans **FixedPointList[f, expr]**, on ne précise pas à l'avance la longueur de la liste. Les calculs sont effectués jusqu'à ce que deux éléments successifs soient égaux.

Sqrt signifie "**Square root**" (= racine carrée : Sqrt[x] = $\sqrt{x}$).

```
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., 1.}
```

Pour mieux comprendre la situation, il faut afficher tous les chiffres significatifs:

```
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, 1.000000000094404, 1.000000000047202, 1.000000000023601,
 1.000000000011801, 1.0000000000059, 1.00000000000295, 1.000000000001475,
 1.000000000000737, 1.000000000000369, 1.000000000000184, 1.000000000000092,
 1.000000000000046, 1.000000000000023, 1.000000000000011, 1.000000000000006,
 1.000000000000003, 1.000000000000001, 1.000000000000001, 1., 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

Le calcul avec des valeurs exactes produit une liste

```
FixedPointList[Sqrt, 3/2, 5]
```
liste de point fixe | racine carrée

$$\left\{\frac{3}{2}, \sqrt{\frac{3}{2}}, \left(\frac{3}{2}\right)^{1/4}, \left(\frac{3}{2}\right)^{1/8}, \left(\frac{3}{2}\right)^{1/16}, \left(\frac{3}{2}\right)^{1/32}\right\}$$

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.

Length

```
? Length
```

Length[*expr*] gives the number of elements in *expr*. ≫

```
Length[sArith]
```
longueur

```
11
```

```
Length[FixedPointList[Sqrt, 1.5]]
```
longueur | liste de point fixe | racine carrée

```
52
```

Exercice 3-1-3

Dans le but de résoudre l'équation

$x = \cos(x)$ où x est en radians

on construit la liste

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

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

b) Démontrez (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". Reprenons maintenant l'exemple donné plus haut:

```
FixedPointList[Sqrt, 1.5]
```

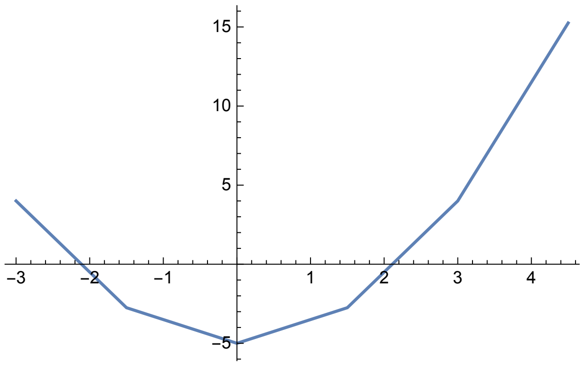
c) Ecrivez sous la forme mathématique la suite de nombres ainsi construite.

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

Représentation graphique d'une liste de points

La commande **ListLinePlot[list]** trace une ligne polygonale passant par une liste de points:

```
ListLinePlot[ptsF, ImageSize → {300, 200}]
```
tracé de liste de ligne | taille d'image



Problème résolu : dessiner un polygone régulier à n côtés

Représentons le polygone par une liste de points dont le dernier coïncide avec le premier pour exprimer que la ligne polygonale correspondante est fermée :

$\{\{x_1, y_1\}, \{x_2, y_2\}, \ldots \{x_1, y_1\}\}$

Les points sont régulièrement disposés sur le cercle trigonométrique: les abscisses sont les cosinus des angles, les ordonnées sont les sinus des angles.

Il faut donc d'abord dresser la liste des angles: un tour vaut 2π radians, un angle au centre du polygone à n côtés vaut $\frac{2\pi}{n}$ et la liste des angles à considérer est

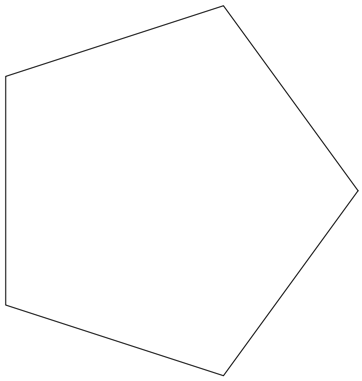
$$\left\{0,\ \frac{2\pi}{n},\ 2*\frac{2\pi}{n},\ 3*\frac{2\pi}{n},\ \ldots,\ n*\frac{2\pi}{n}\right\}$$

```
n = 5;
sPts = Table[{Cos[k 2 π / n], Sin[k 2 π / n]}, {k, 0, n}]
```
table cosinus sinus

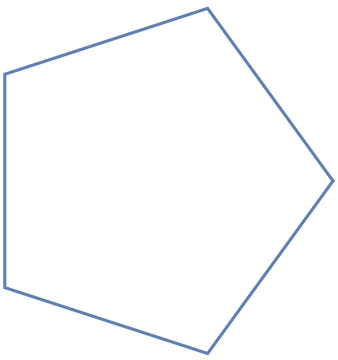
$$\left\{\{1, 0\}, \left\{\frac{1}{4}\left(-1+\sqrt{5}\right), \sqrt{\frac{5}{8}+\frac{\sqrt{5}}{8}}\right\}, \left\{\frac{1}{4}\left(-1-\sqrt{5}\right), \sqrt{\frac{5}{8}-\frac{\sqrt{5}}{8}}\right\}, \left\{\frac{1}{4}\left(-1-\sqrt{5}\right), -\sqrt{\frac{5}{8}-\frac{\sqrt{5}}{8}}\right\}, \left\{\frac{1}{4}\left(-1+\sqrt{5}\right), -\sqrt{\frac{5}{8}+\frac{\sqrt{5}}{8}}\right\}, \{1, 0\}\right\}$$

Cette liste de points - considérée comme une ligne polygonale fermée - peut être dessinée au moyen de plusieurs fonctions graphiques. Nous donnons deux méthodes, la première traduisant le point de vue de la géométrie, la deuxième celui de l'analyse.

```
Graphics[Line[sPts], AspectRatio → Automatic, ImageSize → {200, 200}]
```
graphique ligne rapport d'aspect automatique taille d'image



```
ListLinePlot[sPts, Axes → False, AspectRatio → Automatic, ImageSize → {200, 200}]
```
tracé de liste de ligne axes faux rapport d'aspect automatique taille d'image



Consultez l'aide pour vous informer sur les options telles que **AspectRatio→Automatic** et **ImageSize**.

Exercice 3-1-4

a) Représentez graphiquement la ligne polygonale (ouverte) qui passe par les points suivants:

```
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}}
```

b) Représentez graphiquement la ligne polygonale (ouverte) de sommets:

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

§ 3.2 Extraction d'éléments ou de sous-listes

Dans les exemples qui suivent, nous repartons des listes définies dans le § 3.1

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

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

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

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

Extraction de sous-listes avec Part[...]

Pour extraire le premier ou le dernier élément d'une liste:

```
First[sAngles]
```
premier

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

```
Last[sAngles]
```
dernier

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

Pour extraire l'élément numéro i d'une liste, on utilise le symbole **[[i]]** comme le montrent les exemples suivants :

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

18

```
suite[[3]]
```

7

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

```
? suite
```

```
Global`suite
```

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

Les doubles crochets **[[..]]** sont réservés à l'extraction de sous-listes.

```
? Part
```

expr[[*i*]] or Part[*expr*, *i*] gives the i^{th} part of *expr*.
expr[[–*i*]] counts from the end.
expr[[*i*, *j*, …]] or Part[*expr*, *i*, *j*, …] is equivalent to *expr*[[*i*]][[*j*]] ….
expr[[{i_1, i_2, …}]] gives a list of the parts i_1, i_2, … of *expr*.
expr[[*m* ;; *n*]] gives parts *m* through *n*.
expr[[*m* ;; *n* ;; *s*]] gives parts *m* through *n* in steps of *s*.
expr[["*key*"]] gives the value associated with the key "*key*" in an association *expr*.
expr[[Key[*k*]]] gives the value associated with an arbitrary key *k* in the association *expr*. »

On peut extraire plusieurs éléments consécutifs; par exemple, pour extraire les éléments numéros 3, 4, 5, 6

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

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

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

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

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

On peut utiliser des indices négatifs:

-1 représente le dernier élément,

-2 l'avant-dernier élément, etc.

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

```
1027
```

Test d'appartenance à une liste avec MemberQ[list, form]

Pour savoir si un élément appartient à une liste:

```
MemberQ[Table[i^2, {i, 1, 9}], 64]
est mem··· table
```

```
True
```

```
MemberQ[Table[i^2, {i, 1, 9}], 65]
est mem··· table
```

```
False
```

Recherche d'éléments d'après leurs valeurs

Pour la valeur maximale ou minimale des éléments d'une liste:

```
Max[suite]
maximum
```

```
1027
```

```
Min[sCube]
minimum
```

-8

Pour déterminer le nombre d'occurrences d'un élément dans une liste, on peut utiliser la fonction **Count[liste, element]**:

```
Count[{147, 36, 24, 36, 21, 36, 12}, 36]
compte
```

3

Pour déterminer les positions d'un élément d'une valeur donnée à n'importe quel niveau

```
a = {147, 36, {24, 36}, 21, 36, 12};
pos = Position[a, 36]
      position
```

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

```
? Position
```

Position[*expr*, *pattern*] gives a list of the positions at which objects matching *pattern* appear in *expr*.
Position[*expr*, *pattern*, *levelspec*] finds only objects that appear on levels specified by *levelspec*.
Position[*expr*, *pattern*, *levelspec*, *n*] gives the positions of the first *n* objects found.
Position[*pattern*] represents an operator form of Position that can be applied to an expression. >>

Pour déterminer les positions d'un élément d'une valeur donnée en limitant la recherche au premier niveau

```
pos = Position[a, 36, 1]
      position
```

{{2}, {5}}

Pour déterminer le nombre d'occurrences d'un élément dans une liste

```
Count[a, 36]
compte
```

2

Problème résolu: tableau de fréquences

On lance 100 fois un dé à jouer et on note les résultats obtenus:

```
jets = Table[Random[Integer, {1, 6}], {100}]
       table aléatoire nombre entier
```

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

Calculons la fréquence de chaque face:

```
frequence = Table[{i, Count[jets, i]}, {i, 1, 6}];
            table     compte
```

```
TableForm[frequence, TableHeadings → {None, {"Face", "Nombre d'occurences"}}]
```
forme de table — en-têtes de table — aucun

Face	Nombre d'occurences
1	14
2	21
3	10
4	20
5	17
6	18

Exercice 3-2-1

Considérons les listes

$$ls = \mathrm{Table}\left[\mathrm{Sin}\left[\frac{k\,\pi}{5} + \frac{\pi}{10}\right], \{k, -5, 15\}\right];$$

table — sinus

$$lt = \mathrm{Table}\left[(-1)^{i}, \{i, -5, 15\}\right];$$

table

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

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

Exercice 3-2-2

Etant donné la liste

$$liste = \mathrm{Table}\left[\{i, (-1)^{i}\}, \{i, -6, 6\}\right]$$

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) extrayez le 3-ème élément;
extrayez l'avant-dernier élément;

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

Problème résolu

Dans le chapitre "Equations", nous avons résolu la famille de systèmes d'équations

$$\frac{\alpha - \sin(\alpha)}{2\,\pi} = t \qquad \text{pour } t \in \left\{0, \frac{1}{10}, \frac{2}{10}, \ldots, \frac{9}{10}, 1\right\}$$

$$h = r\left(1 - \cos\left(\frac{\alpha}{2}\right)\right) \qquad \text{pour } r = \frac{1}{2}$$

Nous voulons maintenant montrer comment l'utilisation de listes permet d'écrire des programmes plus courts et plus souples :

$$\texttt{es} = \texttt{Table}\left[\texttt{FindRoot}\left[\frac{\alpha - \texttt{Sin}[\alpha]}{2\,\pi} == \texttt{t},\ \left\{\alpha, \frac{\texttt{t}}{2\,\pi}\right\}\right], \left\{\texttt{t}, 0, 1, \frac{1}{10}\right\}\right]$$

table — trouve racine

... FindRoot: The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances.

```
{{α → 0.}, {α → 1.62675}, {α → 2.11314}, {α → 2.49078}, {α → 2.8248}, {α → 3.14159},
 {α → 3.45839}, {α → 3.7924}, {α → 4.17005}, {α → 4.65643}, {α → 6.28321}}
```

Comme demandé, augmentons la précision, c'est-à-dire le nombre de chiffres significatifs:

$$\texttt{es} = \texttt{Table}\left[\texttt{FindRoot}\left[\frac{\alpha - \texttt{Sin}[\alpha]}{2\,\pi} == \texttt{t},\ \left\{\alpha, \frac{\texttt{t}}{2\,\pi}\right\}, \texttt{WorkingPrecision} \to 30\right], \left\{\texttt{t}, 0, 1, \frac{1}{10}\right\}\right]$$

table — trouve racine — précision de travail

```
{{α → 0}, {α → 1.62675334523314654067956124826}, {α → 2.11313895890097965950434132919},
 {α → 2.49078486654830751533479837656}, {α → 2.82479720235026428021392702669},
 {α → 3.14159265358979323846264338328}, {α → 3.45838810482932219671135973986},
 {α → 3.79240044063127896159048838999}, {α → 4.17004634827860681742094543737},
 {α → 4.65643196194643993624572551829}, {α → 6.28318530753322409041145812683}}
```

$$\texttt{r} = \frac{1}{2};\ \texttt{h} = \texttt{r}\left(1 - \texttt{Cos}\left[\frac{\alpha}{2}\right]\right)\ \texttt{/. es}$$

cosinus

```
{0, 0.156475586933729721076815449582, 0.254069083618145037126175003959,
 0.340154245104748063810961100674, 0.42113190309992098869356632127,
 0.500000000000000000000000000000, 0.578868096900007901130643367873,
 0.659845754895251936189038899326, 0.745930916381854962873824996041,
 0.843524413066270278923184550418, 0.9999999999999999999921837773954878295033}
```

§ 3.3 Opérations sur les listes

Dans les exemples qui suivent, nous repartons des listes définies dans le § 3.1

$$\texttt{sArith} = \texttt{Table}[3 + 5\,\texttt{i}, \{\texttt{i}, 0, 10\}];$$

table

$$\texttt{suite} = \texttt{Table}\left[3 + 2^{\texttt{k}}, \{\texttt{k}, 0, 10\}\right];$$

table

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

table

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

table

Opérations arithmétiques sur les listes

Mathematica autorise d'appliquer les opérations courantes à des listes.

On peut additionner deux listes de même longueur c'est-à-dire additionner les éléments de même position:

```
sArith + suite
```

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

```
sArith + sCube
```

Thread: 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 une liste par un nombre c'est-à-dire multiplier chaque élément de la liste par un nombre:

```
5 * sCube
```

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

On peut appliquer une fonction à une liste c'est-à-dire à tous les éléments de la liste:

```
Cos[sAngles]
```
cosinus

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

On peut aussi élever un nombre à une liste d'exposants:

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

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

Exercice 3-3-1:

Construisez les suites *sArith, suite, sCube* et *sAngles* au moyen de **Range[imin, imax]** au lieu de **Table[]**. C'est une occasion d'effectuer des opérations arithmétiques sur une liste.

```
{3, 8, 13, 18, 23, 28, 33, 38, 43, 48, 53}
{4, 5, 7, 11, 19, 35, 67, 131, 259, 515, 1027}
{-8, -1, 0, 1, 8, 27, 64, 125}
```

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

Flatten

La fonction **Flatten** sert à supprimer certaines paires d'accolades.
Prenons un exemple.

```
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}}}
```

```
liste[[1]]
```

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

```
Length[liste]
```
longueur

5

Observez attentivement les accolades : il s'agit d'une liste de listes de listes !

```
liste[[1]][[3]]
```

{1, 3}

```
liste[[1]][[3]][[2]]
```

3

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.

```
?Flatten
```

Flatten[*list*] flattens out nested lists.
Flatten[*list*, *n*] flattens to level *n*.
Flatten[*list*, *n*, *h*] flattens subexpressions with head *h*.
Flatten[*list*, {{s_{11}, s_{12}, …}, {s_{21}, s_{22}, …}, …}] flattens *list* by combining all levels s_{ij} to make each level *i* in the result. >>

```
tableau = 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}}

```
tableau[[1]]
```

{1, 1}

```
Length[tableau]
```
longueur

25

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

```
liste = {11, {22, {33, {44, {55, {66}}}}}};
Flatten[liste]
```
aplatis

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

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

```
Flatten[liste, 2]
```
aplatis

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

Problème résolu : extraction des éléments positifs d'une liste

Etant donné la liste

```
a = Table[Cos[Range[0, 6]]]
    table  co… plage
{1, Cos[1], Cos[2], Cos[3], Cos[4], Cos[5], Cos[6]}
```

on peut en extraire les éléments positifs en utilisant la fonction

Sign[x] qui signifie signe de x

de la manière suivante:

```
Sign[a]
signe
{1, 1, -1, -1, -1, 1, 1}
```

```
pos = Position[Sign[a], 1]
      position  signe
{{1}, {2}, {6}, {7}}
```

Les éléments à retenir sont

```
a[[Flatten[pos]]]
   aplatis
{1, Cos[1], Cos[5], Cos[6]}
```

Coïncidences de deux listes de nombres

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

Pour déterminer les coïncidences entre deux listes, on peut observer la position des zéros dans la différence des deux listes:

```
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. Une coïncidence 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.

```
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
```

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[...]** est un format d'affichage):

```
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):

```
Transpose[sPts]
```

transposée

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

```
Transpose[sao]
```

transposée

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

```
MatrixForm[Transpose[frequence]]
```

apparence m·· transposée

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 14 & 21 & 10 & 20 & 17 & 18 \end{pmatrix}$$

Utilisons la transposition pour former une liste de points:

```
x = Range[0, 9];
```

plage

```
f[x_] := x^2 - 3;
sPts = Transpose[{x, f[x]}]
```

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:

```
y = Transpose[sPts][[2]]
```

transposée

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

Exercice 3-3-2

Affichez la liste des 10 premiers nombres premiers

a) en un tableau sur deux lignes:

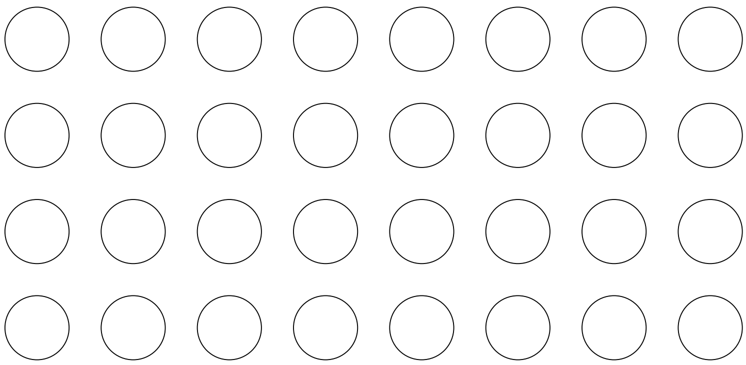
$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 & 23 & 29 \end{pmatrix}$$

b) en un tableau sur deux colonnes:

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

Exercice 3-3-3

Dessinez le graphique suivant :



Indications:

la liste des cercles est de la forme **{Circle[{x1, y1}, *r*], Circle[{x2, y2}, *r*], ...}**;
ne pas dactylographier la liste des cercles, mais utiliser la fonction **Table[...]** pour la créer;
utiliser **Graphics[listeDesCercles]** pour la dessiner.

Exercice 3-3-4 [facultatif]

On donne la fonction

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

a) Dressez les trois listes suivantes

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

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

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

b) Calculez les valeurs maximale et minimale de la liste y
puis déterminez les abscisses correspondantes.

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

d) Extrayez de *y* les valeurs qui sont positives
puis formez la liste des points correspondants
et affichez ces points sous la forme d'un tableau de nombres en virgule flottante.

Appliquer à une liste une fonction ayant un nombre quelconque d'arguments: Apply[f, {a1, a2, ...}] = f[a1, a2, ...]

Somme, produit et moyenne d'une liste

Plus[a1, a2, ...] est une fonction à plusieurs arguments qui donne la somme des arguments:

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

```
14
```

Par contre, lorsqu'on applique **Plus** à une liste, on n'obtient pas la somme des éléments car la fonction **Plus** n'a qu'un seul argument

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

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

```
Plus[sCube]
```
plus

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

La commande **Apply[...]** permet de transformer les éléments d'une liste en arguments d'une fonction

```
? Apply
```

Apply[*f*, *expr*] or *f* @@ *expr* replaces the head of *expr* by *f*.
Apply[*f*, *expr*, {1}] or *f* @@@ *expr* replaces heads at level 1 of *expr* by *f*.
Apply[*f*, *expr*, *levelspec*] replaces heads in parts of *expr* specified by *levelspec*.
Apply[*f*] represents an operator form of Apply that can be applied to an expression. >>

Apply[f,{a,b,c}] équivaut à **f[a,b,c]**; par exemple,

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

```
14
```

On peut ainsi obtenir la somme des éléments d'une liste

```
Apply[Plus, sCube]
```
remp··· plus

```
216
```

D'une manière analogue, on peut calculer le produit des éléments d'une liste:

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

```
120
```

```
Times[sCube]
multiplication
```

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

```
Apply[Times, sCube]
remp··· multiplication
```

```
0
```

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*2*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 * 2 * 3 * 4 * ... * n

```
Clear[factorielle];
efface
factorielle[n_] := Apply[Times, Range[n]]
                   remp··· 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}}
```

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

```
moyenne = Apply[Plus, sCube] / Length[sCube]
```

```
27
```

Mieux encore: on peut définir une fonction qui calcule la moyenne de n'importe quelle liste

```
Clear[moyenne];
efface
moyenne[x_List] := Apply[Plus, x] / Length[x]
moyenne[sCube]
```

```
27
```

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

Exercice 3-3-5

Calculez la somme, le produit et la moyenne arithmétique de chacune des listes suivantes: *sArith, suite, sCube* et *sAngles*.

Appliquer une fonction à chaque élément d'une liste: Map[f, {a1, a2, ...}] = {f[a1], f[a2], ...]

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

```
Tan[Range[-4, 4]]
```
ta··· plage

```
{-Tan[4], -Tan[3], -Tan[2], -Tan[1], 0, Tan[1], Tan[2], Tan[3], Tan[4]}
```

```
Clear[f]; f[x_] := x^2;
```
efface

```
f[Range[9]]
```
plage

```
{1, 4, 9, 16, 25, 36, 49, 64, 81}
```

Par contre, d'autres fonctions, lorsqu'elles sont appliquées à une liste, ne sont pas distribuées aux éléments de cette liste. On dit qu'elles ne sont pas "Listable". C'est en particulier le cas pour certaines fonctions telles que **Show**, **Graphics**, **Line**, **Polygon**, **Circle**, **Point**, **...**

```
a = Table[{x, f[x]}, {x, 1, 9}];
```
table

```
Point[a]
```
point

```
Point[{{1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}, {6, 36}, {7, 49}, {8, 64}, {9, 81}}]
```

L'opérateur **Map** sert à appliquer une fonction aux éléments d'une liste. Par défaut, la fonction est appliquée aux éléments du premier niveau de la liste

```
Map[Point, a]
```
app· point

```
{Point[{1, 1}], Point[{2, 4}], Point[{3, 9}], Point[{4, 16}],
 Point[{5, 25}], Point[{6, 36}], Point[{7, 49}], Point[{8, 64}], Point[{9, 81}]}
```

```
Clear[f]; f[Range[9]]
```
efface plage

```
f[{1, 2, 3, 4, 5, 6, 7, 8, 9}]
```

```
Map[f, Range[9]]
```
applique plage

```
{f[1], f[2], f[3], f[4], f[5], f[6], f[7], f[8], f[9]}
```

```
? Map
```

Map[*f*, *expr*] or *f* /@ *expr* applies *f* to each element on the first level in *expr*.
Map[*f*, *expr*, *levelspec*] applies *f* to parts of *expr* specified by *levelspec*.
Map[*f*] represents an operator form of Map that can be applied to an expression. >>

Exercice 3-3-6

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

Prescription d'exercice: utilisez Map.

b) Dressez la liste des points correspondants.

Graphique avec Map[Line, {listeDeLignesPolygonales}]

Pour tracer les 4 fenêtres de la maison, on applique une translation à chaque sommet de la première fenêtre:

```
SetOptions[Graphics, AspectRatio → Automatic];
alloue options graphique rapport d'aspect automatique
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 translater d'un vecteur t une ligne polygonale f1, on additionne t à chaque élément x de f1:

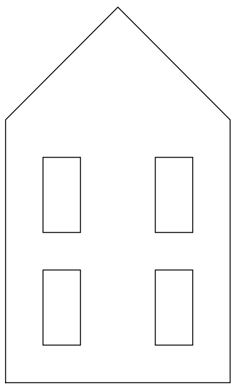
```
Clear[add]; add[t_][x_] := x + t;
efface
f2 = Map[add[{0, 3}], f1];
     applique
f3 = Map[add[{3, 0}], f1];
     applique
f4 = Map[add[{3, 3}], f1];
     applique
```

Expliquons ce qui précède

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

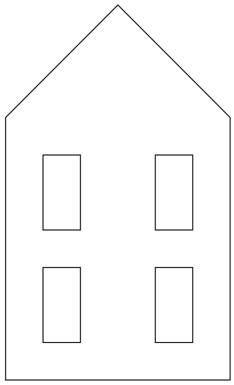
Comme il y a 5 lignes polygonales fermées à tracer, il faudra appeler 5 fois la fonction **Line[]**:

```
Graphics[{Line[c], Line[f1], Line[f2],
graphique ligne     ligne     ligne
          Line[f3], Line[f4]}, ImageSize → {200, 200}]
          ligne     ligne        taille d'image
```



Grâce à l'usage de l'opérateur Map, il est possible d'éviter d'écrire 5 fois la fonction **Line[]**:

```
maison = {c, f1, f2, f3, f4};
Graphics[Map[Line, maison], ImageSize → {200, 200}]
graphique   app· ligne             taille d'image
```



Expliquons ce qui précède :

```
       Map[Line, maison]
     = Map[Line, {c, f1, f2, f3, f4}]
 = {Line[c], Line[f1], Line[f2], Line[f3], Line[f4]}
```

Listes comme ensembles (facultatif)

Une liste (ordonnée) peut être utilisée pour décrire un ensemble (non ordonné). Les intéressés peuvent se rapporter aux annexes du § 3, plus précisément:

https://www.deleze.name/marcel/sec2/applmaths/csud/initiation_mathematica/annexes/3-s1-ensembles.nb

Produit scalaire de deux listes

Dans le cours de mathématiques, on étudie le produit scalaire de deux vecteurs du plan :

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = u_1\, v_1 + u_2\, v_2$$

Dans le langage Mathematica, cette opération peut être exécutée entre deux listes de même longueur.
Le symbole du produit scalaire est le point :

```
Clear[a, b, c, d, e, f]
efface
```

```
{a, b, c}.{d, e, f}
```

a d + b e + c f

Exemple 1 (géométrie analytique)

Pour calculer la norme d'un vecteur $\vec{u}$, on peut extraire la racine carrée du carré scalaire :

```
Clear[norme];
efface
```

```
norme[u_List] := √u.u ;
norme[{a, b}]
```

$\sqrt{a^2 + b^2}$

Pour calculer l'angle φ entre deux vecteurs $\vec{u}$, $\vec{v}$, vous connaissez la formule du produit scalaire :

$$\cos(\varphi) = \frac{\vec{u}.\vec{v}}{\|\vec{u}\| \, \|\vec{v}\|}$$

En Mathematica, définissons une fonction qui donne l'angle en radians entre deux vecteurs :

```
Clear[angle];
efface
angle[u_List, v_List] := ArcCos[u.v / (norme[u] norme[v])];
                         arc cosinus
```

```
angle[{1, 0}, {1, 1}]
```

$\frac{\pi}{4}$

Exemple 2 (polynômes)

En mathématiques, nous avons vu qu'un polynôme est de la forme

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

Les coefficients d'un polynôme étant donnés (dans l'ordre a_0, a_1, ...)

```
coeff = {9, -8, 7, -6, 5, -4}
```

{9, -8, 7, -6, 5, -4}

on peut définir la fonction polynôme comme étant le produit scalaire des deux listes

$$\{9, -8, 7, -6, 5, -4\}.\{1, x, x^2, x^3, x^4, x^5\}$$

```
Clear[p, x];
efface
p[x_] := coeff.Table[x^k, {k, 0, Length[coeff] - 1}];
               table           longueur
```

```
p[x]
```

$9 - 8x + 7x^2 - 6x^3 + 5x^4 - 4x^5$

Exercice 3-3-7

On donne les coordonnées des sommets d'un triangle ABC. Calculez

les longueurs des 3 côtés et

les angles intérieurs α, β, γ, exprimés en degrés.

A (-1; -2), B (1; 5), C (7; 3)

Exercice 3-3-8

Etant donné une liste de notes et la liste des coefficients correspondants, calculez la moyenne pondérée des notes

a) Données numériques (la note 4.5 a le coefficient 3; la note 5 a le coefficient 2; ...) :

```
notes = {4.5, 5, 3.5, 4, 6, 5.5};
coeff = {3, 2, 1, 1, 1/2, 1/2};
```

b) Généralisez : écrivez une fonction qui calcule la moyenne de la liste "notes" pondérée par les coefficients "coeff" (on suppose que les deux listes ont la même longueur).

Exercice 3-3-9

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*"). **Données numériques pour l'exercice 3-3-9 :**

https://www.deleze.name/marcel/sec2/applmaths/csud/initiation_mathematica/annexes/3-3-9-donnees-exercice.nb

```
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, 10 086, 10 612, 11 160, 11 735, 12 334, 12 959, 13 611, 14 292, 15 000,
   15 737, 16 505, 17 308, 18 143, 19 012, 19 916, 20 856, 21 834, 22 849, 23 906, 25 003, 26 143,
   27 326, 28 554, 29 828, 31 157, 32 517, 33 944, 35 424, 36 957, 38 543, 40 183, 41 877,
   43 636, 45 463, 47 343, 49 289, 51 316, 53 409, 55 569, 57 809, 60 115, 62 488, 64 941,
   67 474, 70 096, 72 801, 75 592, 78 474, 81 477, 84 513, 87 675, 90 935, 94 295, 97 757,
   101 325, 105 000, 108 772, 112 673, 116 665, 120 799, 125 046, 129 403, 133 912, 138 511,
   143 263, 148 148, 153 153, 158 310, 163 620, 169 050, 174 644, 180 378, 186 275, 192 335};
```

a) Formez la liste des points de mesures {{0, 611}, ..., {119, 192 335}} et dessinez le graphique de la pression de saturation de l'eau en fonction de la température.

b) Formez la liste des points de mesures {{611, 0}, ..., {192 335, 119}} et dessinez le graphique de la température d'ébullition de l'eau en fonction de la pression.

c) A partir de **psat**, formez un tableau dans lequel chaque ligne est constituée de 10 éléments
`tabl={{611, ..1148}, {1228, ...}, ..., {..., 192335}}`
Présentez les pressions de saturation sous la forme du tableau donné ci-dessous.
Les commandes de mise en page sont données.

En l'absence de précaution, le tableau à construire est trop large pour être imprimé. Afin d'éviter qu'il soit coupé, nous allons tourner le tableau d'un quart de tour.

Packages de l’auteur

Des packages ont été développés pour les cours *Applications des mathématiques.* Ici, nous utilisons le package Tableaux:

- On peut consulter le mode d’emploi:
  https://www.deleze.name/marcel/sec2/applmaths/packages/aide/Tableaux.pdf
- Avant d’utiliser le package, il faut le charger en donnant son adresse web:

```
Needs["Tableaux`",
nécessite
  "https://www.deleze.name/marcel/sec2/applmaths/packages/Tableaux.m"]
```

```
? tableauGraph
```

tableauGraph[t] affiche le tableau bidimensionnel t tourné d'un quart de tour;
tableauGraph[l, c, t] affiche le tableau {{Null, c}, {l, t}} tourné d'un quart de tour où
l est la liste des en–têtes de lignes,
c est la liste des en–têtes de colonnes et
t est un tableau bidimensionnel.

Afin que les nombres soient alignés dans les colonnes du tableau, définissons un format d'affichage **formatNum** de la liste de nombres **v** dans un champ de **nbChiffres**:

```
formatNum[nbChiffres_][v_] :=
 Map[PaddedForm[#, nbChiffres] &, v, {Length[Dimensions[v]]}]
```

app· forme de remplissage longueur dimensions

```
tableauGraph[formatNum[3][Range[0, 110, 10]], Range[0, 9], formatNum[6][tabl]]
```

plage plage

	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

§ 3.4 Extraction d'éléments ou de sous-listes avec Select

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

$$\text{suite} = \text{Table}\left[3 + 2^{k}, \{k, 0, 10\}\right];$$
table

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

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

```
? Select
```

Select[*list*, *crit*] picks out all elements e_i of *list* for which *crit*[e_i] is True.
Select[*list*, *crit*, *n*] picks out the first *n* elements for which *crit*[e_i] is True.
Select[*crit*] represents an operator form of Select that can be applied to an expression. >>

Pour extraire d'une liste tous les éléments vérifiant un certain critère, on peut utiliser la fonction **Select**. Par exemple, pour extraire tous les éléments positifs:

```
Clear[positifQ];
efface
positifQ[x_] := x > 0;
Select[sCube, positifQ]
sélectionne
```

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

Le critère de sélection **positifQ** est une fonction à valeur booléenne (vrai ou faux):

```
positifQ[-7]
```

```
False
```

```
positifQ[8]
```

```
True
```

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

Pour extraire tous les angles dont le cosinus est négatif:

```
Clear[cosNegQ];
efface
cosNegQ[x_] := Cos[x] < 0;
               cosinus
Select[sAngles, cosNegQ]
sélectionne
```

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

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

"odd" signifie "impair"; ("OddQ" pose la question "impair ?");
"even" signifie "pair";
"prime" signifie "premier";
"integer" signifie "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 pr
```

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

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

```
{}
```

Voyons maintenant comment l'utilisateur peut définir lui-même un critère de sélection.

La condition d'égalité s'écrit ==

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

```
True
```

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

```
False
```

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

```
Clear[mult3Q];
effface
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}
```

Il y a donc lieu de distinguer

le symbole d'affectation immédiate =

le symbole d'affectation différée :=

le symbole d'égalité logique ==

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" et "7")

ce qui équivaut au ∨ de la palette.

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

```
Clear[critQ];
efface
critQ[x_] := EvenQ[x] ∧ x > 20;
             nombre pair?
Select[sArith, critQ]
sélectionne
```

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

Exercice 3-4-1

On donne la liste

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

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

Formez 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.

Problème résolu : triplets de Pythagore (version avec Select)

Dresser une partie de la liste des triplets de Pythagore, plus précisément:
dressez la liste des triplets (a, b, c) tels que

a, b, c sont des entiers positifs

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

$2 \leq a < b \leq 12$

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, √(a² + b²)}, {a, 2, 11}, {b, a + 1, 12}], 1];
            aplatis  table
```

Enlevez les points-virgules afin de voir les résultats intermédiaires.

Les candidats vérifient certaines conditions requises, mais pas toutes. Une liste de candidats est une liste finie qui contient l'ensemble des solutions.
La deuxième partie du calcul est un filtrage. Nous ne retenons que les candidats dont la troisième composante est entière:

```
Clear[critQ];
efface
critQ[x_] := IntegerQ[x[[3]]];
             entier?
Select[candidats, critQ]
sélectionne
```

```
{{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}}

Exercice 3-4-2 (version avec Select)

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

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

Extraction de sous-listes avec Cases[list, pattern /; condition] (Facultatif)

Les intéressés peuvent se rapporter aux annexes du § 3, plus précisément:

https://www.deleze.name/marcel/sec2/applmaths/csud/initiation_mathematica/annexes/3-s2-cases.nb

Exercices de récapitulation § 3.1 à § 3.3

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

Exercice 3 - R1

Construisez 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 - R2

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

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

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

Exercice 3-R3

a) Construisez 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]

construisez 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) Dessinez la figure correspondante.

Liens

Vers les corrigés des exercices :

https://www.deleze.name/marcel/sec2/applmaths/csud/corriges/initiation_mathematica/3-listes-cor.pdf

Vers la page mère: Applications des mathématiques

https://www.deleze.name/marcel/sec2/applmaths/csud/index.html