# Les nombres

## 1 Les entiers

**Exercice 1** (écrire un entier).
\$123456789\$ donne 123456789. On aimerait obtenir ceci : 123 456 789.

1. Essayer en utilisant les espaces mathématiques. Vous trouverez la liste ci-dessous par exemple dans « Une courte (?) introduction à LaTeX $2_\varepsilon$ » (faire une recherche sur « espacement en mode math » dans le fichier pdf flshort).

| Espaces positives | Espaces négatives |
|---|---|
| \, ou **thinspace** | \! ou **negthinspace** |
| \: ou \medspace | \negmedspace |
| \; ou \thickspace | \negthickspace |
| **\enskip** | |
| **\quad** | |
| **\qquad** | |

   Plus général : \mspace{0.5mu} (\usepackage{amsmath} en préambule). La quantité 0.5 peut être remplacé par une autre quantité, éventuellement négative.
2. Essayer avec les instructions du package numprint. On trouvera la doc de ce package par exemple à l'adresse :
   `http://texcatalogue.sarovar.org/entries/numprint.html`
   ou dans les dossiers de votre distribution LaTeX.

**Exercice 2** (poser une opération).
A l'aide du package xlop, obtenir :

$$\begin{array}{r} 4\,5\,1 \\ +\,7\,1\,1\,9 \\ \hline 7\,5\,7\,0 \end{array} \qquad \begin{array}{r|l} 2\,5 & 6 \\ \hline 1 & 4 \end{array}$$

## 2 Les décimaux

**Exercice 3** (écrire un décimal).

1. Coder : $5 \cdot 10^{-7}$
   Un exposant s'obtient avec ^ et un « groupe » se délimite en LaTeX avec des accolades.
2. Coder : 12,783 895
   L'écriture \$12,783895\$ laisse un espace derrière la virgule qui ne convient pas. Résolver ce problème :
   (a) en écrivant la virgule entre accolades,
   (b) en utilisant le package numprint .

# 3 Pourcentages et unités

**Exercice 4.**
Coder : 12%
Le symbole % sert à écrire des commentaires dans le fichier source. Comme d'autres symboles interprétés, on l'obtiendra en le faisant précéder d'un \ .

**Exercice 5.**
Coder : 12 $\mathrm{mm}^2$
Dans un champ mathématique, les lettres d'un texte sont interprétées comme étant des symboles mathématiques...

1. L'instruction \mathrm{} est une solution (voir des exemples d'utilisation de \mathrm{} dans « Une courte (?) introduction à LaTeX $2_\varepsilon$ » ).
2. L'utilisation du package numprint en est une autre.

# 4 Les rationnels, les fractions

**Exercice 6.**
Coder : $\frac{1}{2}$ puis $\dfrac{1}{2}$. *Pour connaître les instructions, utiliser les raccourcis proposés par l'éditeur de textes.*

**Exercice 7.**
Donnez le code de : $\dfrac{\dfrac{a}{b}}{\dfrac{c}{d}} = \dfrac{a}{b} \times \dfrac{d}{c}$

1. On pourra utiliser des espaces pour augmenter la longueur du trait de fraction principal.
2. Les opérateurs usuels (× par exemple) se trouvent dans les menus de l'éditeur de textes.

**Exercice 8.**

$$\frac{4}{\pi} = 1 + \cfrac{1}{2 + \cfrac{3^2}{2 + \cfrac{5^2}{2 + \cfrac{7^2}{2 + \cfrac{}{2 + \ldots}}}}}$$

On pourra utiliser l'instruction \cfrac{} du package amsmath.

# 5 Irrationnels et ensembles de nombres

**Exercice 9.**
Coder :

$$\begin{aligned}|x| &= x\,\mathbb{1}_{[0;+\infty[}(x) - x\,\mathbb{1}_{]-\infty;0]}(x)\\ &= \max(x;-x)\\ &= \max(x;0) - \min(x;0)\\ &= \sqrt{x^2}\\ &= \begin{cases} x \text{ si } x \in [0;+\infty[\\ -x \text{ si } x \in ]-\infty;0]\end{cases}\end{aligned}$$

1. L'instruction \mathbb{} du package mathbbol permet les notations « ajourées ».
2. \begin{split}...\ **end**{split} du package amsmath se gère comme un tableau (utilisation de & pour séparer les cellules et de \\ pour changer de ligne) dans un champ mathématique et permet les alignements sur les symboles d'égalité. Bien d'autres solutions sont envisageables pour ce résultat.
3. \begin{cases}...\\...\ **end**{cases} pour une présentation accolade gauche.

**Exercice 10.**
Coder :

| | $\mathbb{N}$ | $\mathbb{Z}$ | $\mathbb{D}$ | $\mathbb{Q}$ | $]0;+\infty[$ | $\mathbb{R}$ |
|---|---|---|---|---|---|---|
| $\cos(2\pi)$ | $\in$ | | | | | |
| 1,2 | $\notin$ | | | | | |
| $-\pi$ | | | | | | |
| $\sqrt{\frac{3}{2}}$ | | | | | | |
| $\ln\left(\frac{5}{3}\right)$ | | | | | | |
| $\exp\left(\frac{1}{2}\right) = \sqrt{e}$ | | | | | | |

Pour les symboles $\in$ et $\notin$, consulter les menus de l'éditeur de textes.
Une convention typographique à connaître : la lettre e pour le nombre e est à écrire en droit.

# 6 La calculatrice

**Exercice 11.**
Définir une nouvelle commande \calculatrice telle que :

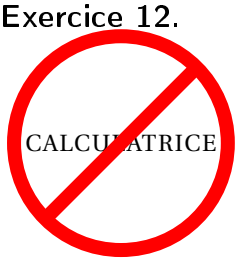
* \calculatrice affichera :
  La calculatrice n'est pas autorisée.
* et \calculatrice[est] affichera :
  La calculatrice est autorisée.

On cherchera dans « Une courte (?) introduction à LaTeX 2$\varepsilon$ » ce qui est dit sur \newcommand .

```
\newcommand{\NomCommande}[Nombre de paramètres][valeur par défaut du paramètre 1]
{définition de la commande, le paramètre numéro i est référencé par #i}
```

Le nombre de paramètres va de 0 à 9.

**Exercice 12.**




Dans la doc de pgf–tikz : consulter le paragraphe « Shape Library ».

**Exercice 13.**

Pour un travail sur la calculatrice, on veut représenter les touches d'une calculatrice, par exemple : [5] [×] [3] [=] ...

Comment peut-on procèder ?

**Exercice 14** (triangle affichage casio).

Pour écrire les instructions d'un programme de calculatrice casio, définir une commande \trianglecasio qui affichera ◢

# 7 Feuille tableur

**Exercice 15.** 1. Pour représenter une feuille de tableur dans un texte, on peut évidemment coller une copie écran d'une partie de la feuille (la copie décran peut se faire avec n'importe quel logiciel de copie d'écran, on pourra par exemple essayer le logiciel gratuit capturino que l'on trouvera facilement sur le web) :

| | A | B |
|---|---|---|
| 1 | 5 | 5 |
| 2 | 54 | 3 |
| 3 | 6 | 6 |
| 4 | | |

avec les instructions suivantes :

```
\documentclass[12pt,a4paper]{article}
\usepackage{graphicx}
\begin{document}
\includegraphics[scale=0.8]{imagefeuilletableur}
\end{document}
```

Une petite gêne : lorsqu'on compile avec latex–dvips le format de l'image doit être .eps et lorsqu'on compile avec pdflatex le format d'image devra être jpg ou png par exemple. Il faut donc toujours

avoir deux formats de ses images. Un outil simple lorsqu'on n'a que peu d'images : open office draw, logiciel de dessin que l'on télécharge avec la suite libre et gratuite open office, exporte facilement une image jpg ou png en une image eps par exemple.

2. Une autre méthode pour reproduire une feuille de calcul est de coder un tableau. Reproduire et compiler le code suivant :

```
\documentclass[12pt,a4paper]{article}
\usepackage{xcolor}
\usepackage[table]{colortbl}
\begin{document}
\begin{tabular}{|>{\columncolor{lightgray}}c|c|c|c|c|c|}
\hline
\rowcolor{lightgray} &A & B & C & D & E \\
\hline
1&  $a$= &\textit{valeur de} $a$ &&&\\
\hline
2& &&&&\\
\hline
3 & Indices $n$& Termes $u_n$  &  &  &  \\
\hline
4  &  0& \textit{valeur de} $u_0$ &  &  &  \\
\hline
5 &  1&\textit{valeur de} $u_1$  &  &  &  \\
\hline
 6 &  2& \textit{valeur de} $u_2$ &  &  &  \\
\hline
\vdots &  \vdots &  &  &  &  \\
\hline
\end{tabular}
\end{document}
```

# 8 Des solutions

**Exercice 1** (écrire les entiers).
123 456 789 s'obtient par \$123\, 456\, 789\$ ou par \nombre{123456789}.
Cette instruction \nombre{} est une instruction du package frenchb de babel.
Le préambule contiendra donc \usepackage[frenchb]{babel} .
Ce package gérant aussi divers usages typographiques en usage en France, il est conseillé de le charger systématiquement.
Si l'on a chargé le package numprint, pour obtenir 123 456 789 on peut aussi écrire : \numprint{123456789} ou son abrégé \np{123456789} si l'on a chargé numprint avec l'option np (c'est à dire sous la forme : \usepackage[np]{numprint} ).

**Exercice 2** (poser une opération).

```
% dans le préambule :
\usepackage{xlop}
% dans le corps du texte :
 \opadd[voperator=bottom,carryadd=false]{451}{7119} \qquad
 \opidiv{25}{6}
```

**Exercice 3** (écrire un décimal).

1. \$5**cdot** 10^{−7}\$ (ou \$5.10^{−7}\$ si on veut un point sur la ligne de base).
   Avec le package numprint, on obtiendra $5 \cdot 10^{-7}$ avec le code \np{5E{−7}} .
   Le point marquant le produit n'est pas sur la ligne de base, si on veut changer cela on écrira dans le préambule : \npproductsign{.} et \np{5E{−7}} donnera alors : $5.10^{-7}$ .
   La convention typographique est celle de · et du package numprint (le point est au-dessus de la ligne de base).
   Si on veut éviter l'utilisation du point, on écrira dans le préambule : \npproductsign{**times**} et \np{5E{−7}} donnera alors : $5 \times 10^{-7}$ .
2. (a) \$12,783\, 895\$ donne 12,783 895 .
   (b) Avec numprint, on écrira \np{12.883895} qui donne 12,883 895.
   Avec \usepackage[frenchb]{babel}, \nombre{12,857895} donne 12,857 895
   Ces deux solutions semblent mieux adaptées aux conventions usuelles : pas d'espace après la virgule.
   (c) En mode mathématique, la virgule est toujours suivie d'une espace car elle est considérée comme une ponctuation et non comme une virgule décimale.
   (d) On peut aussi taper : \$12{,}75\$ qui donne 12,75 et non 12, 75 (suppression de l'espace après la virgule).

**Exercice 4.**
Le symbole % sert à écrire des commentaires dans un fichier LaTeX (texte non pris en compte par le compilateur). Pour écrire 12% , il faut en conséquence écrire : 12 \%

**Exercice 5.**
Le code `$12\ \mathrm{mm}^2$` donne $12\ \mathrm{mm}^2$.
Avec le package numprint, on obtient $12\,\mathrm{mm}^2$ par le code `\np[mm^2]{12}`.

**Exercice 6** (fraction displaystyle ou textstyle).
`$\frac{1}{2}$` puis `$\dfrac{1}{2}$`.
Mais la frappe de `\[ \frac{1}{2} \]` donne :

$$\frac{1}{2}$$

En fait l'encadrement par `$` et `$` sert à écrire des mathématiques dans une ligne de texte (petite fraction pour ne pas interférer avec la ligne du dessous, style textstyle) et l'encadrement par `\[` et `\]` sert à écrire des mathématiques « hors texte ».
Si l'on veut écrire une grande fraction dans une ligne de texte, on écrit `$\displaystyle\frac{a}{b}$` ou son raccourci : `$\dfrac{a}{b}$` (package amsmath).
Si à l'usage, on décide de ne jamais utiliser l'écriture "écrasée" pour les math en ligne de texte, on pourra écrire dans le préambule : `\everymath{\displaystyle}` . Ainsi, l'instruction `$\frac{a}{b}$` produira toujours $\frac{a}{b}$ et non $\frac{a}{b}$.
Si on veut avoir une petite fraction alors qu'on a déclaré `\everymath{\displaystyle}` en préambule, on pourra écrire `$\textstyle\frac{a}{b}$` ou son raccourci (nécessite le package amsmath) : `$\tfrac{a}{b}$`.

**Exercice 7** (quotient de fractions).
Avec un `\usepackage{amsmath}` dans le préambule, on pourra coder par :
`$\dfrac{\ \dfrac{a}{b}\ }{\dfrac{c}{d}}=\dfrac{a}{b}\times\dfrac{d}{c}$`
L'instruction `\ ` laisse un espace vide. On a placé ici ces deux espaces vides pour augmenter la taille du trait de fraction principal.
Si on veut ajuster comme on veut la longueur de la barre de fraction principale, on pourra utiliser l'instruction `\mpsace{...mu}` du package amsmath, par exemple :
`$\dfrac{\mspace{6mu} \dfrac{a}{b}\mspace{6mu} }{\dfrac{c}{d}}$`

On peut aussi vouloir mettre en évidence le trait de fraction principal en jouant sur son épaisseur : $\dfrac{\ \dfrac{a}{b}\ }{\dfrac{c}{d}}$

ce que l'on obtient par (package amsmath) :
`$\genfrac{}{}{0.8pt}{}{\mspace{6mu} \dfrac{a}{b}\mspace{6mu} }{\dfrac{c}{d}}$`

**Exercice 8** (fraction continue).

```
\[\dfrac{4}{\pi}
=1+\cfrac{1}{2+\cfrac{3^2}{2+\cfrac{5^2}{2+\cfrac{7^2}{2+\dots}}}}\]
```

**Exercice 9** (définitions de la valeur absolue).

```
% dans le préambule :
\usepackage{mathbbol}
% dans le corps du texte :
\[ \begin{split}
\vert x \vert &=
```

```
x\, \mathbb{1}_{\left[ 0;+\infty\right[ }(x)-x\, \mathbb{1}_{\left]-\infty; 0\right] }(x
    )\\
&=\max\left(x;-x \right)\\
&=\max\left( x;0\right) -\min\left( x; 0\right)\\
&= \sqrt{x^2\ }\\
&= \begin{cases} x\text{ si } x\in\left[ 0;+\infty\right[ \\
-x\text{ si } x\in\left]-\infty; 0\right]
\end{cases}
\end{split} \]
```

Exercice 10 (tableau ensemble de nombres).

```
\[
% tableau de 7 colonnes, texte centré
\begin{array}{|c|c|c|c|c|c|c|}
% trait horizontal : \hline
\hline
% séparation des colonnes par &
% fin d'une ligne du tableau par double slash :
    \\
&\mathbb{N}&\mathbb{Z}&\mathbb{D}
&\mathbb{Q}&\left] 0;+\infty\right[ &\mathbb{R}\\
\hline
\cos\left( 2\pi\right) &\in&&&&&\\
\hline
1,2&\notin&&&&&\\
\hline
-\pi&&&&&&\\
\hline
\sqrt{\dfrac{3}{2}}&&&&&&\\
\hline
\ln\left( \dfrac{5}{3}\right) &&&&&&\\
\hline
\end{array}
\]
```

Pour règler le problème des fractions ou parenthèses qui viennent gratouiller les bordures de cellule, on peut définir la commande suivante :

```
% dans le préambule
\usepackage{calc}
\newlength{\DimensionS} \newlength{\DimensionP}
\newsavebox{\Boite}
\newcommand{\etai}[2][3pt]%
{\savebox{\Boite}{\ensuremath{#2}}%
\setlength{\DimensionS}{\ht\Boite+\dp\Boite+#1+#1}%
\setlength{\DimensionP}{\dp\Boite+#1}%
\rule[-\DimensionP]{0pt}{\DimensionS}\ensuremath{#2}%
}
```

avec l'utilisation suivante :

```
$\begin{array}{|c|}
\hline
\etai{\sqrt{\dfrac{3}{2}}}\\
\hline
\end{array}$
```

qui donne $\boxed{\sqrt{\frac{3}{2}}}$ au lieu de $\boxed{\sqrt{\frac{3}{2}}}$

Ce problème de gratouillage des bordures par les fractions peut aussi être résolu par le package cellspace. Vous pourrez essayer l'exemple suivant :

```
\documentclass[10pt]{article}
\usepackage{cellspace}
\begin{document}
\begin{tabular}{|Sc|c|}
\hline $\sqrt{\frac{3}{2}}$ &  \\
\hline $\sqrt{\frac{3}{2}}$ &  \\
\hline  & $\sqrt{\frac{3}{2}}$ \\
\hline
\end{tabular}
\end{document}
```

**Exercice 11** (calculatrice autorisée ... ou non).

1. Syntaxe pour définir une nouvelle commande :

```
\newcommand{\NomDeLaCommande}
[Nb De Paramètres (entre 0 et 9) ]
[Si le paramètre 1 est optionnel, ici sa valeur par défaut]
{définition de la commande, #i désigne le paramètre i}
```

```
\newcommand{\calculatrice}[1][n'est pas]{\ovalbox{La calculatrice #1 autorisée.}}
```

   [1] signifie que la commande a un paramètre. [n'est pas] a pour conséquence le caractère optionnel de ce paramètre et lui donne par défaut la valeur "n'est pas". #1 est remplacé par la valeur du paramètre lorsque la macro est développée. Ainsi \calculatrice[glup] donnerait : La calculatrice glup autorise

2. On pourra essayer la définition suivante :

```
\usepackage{ifthen}
\newcommand{\calculatrice}[1][yes]{
\ifthenelse
{\equal{#1}{yes}}
{\ovalbox{La calculatrice est autorisée.}}
{\ifthenelse{\equal{#1}{no}}{\ovalbox{La calculatrice n'est pas autorisée
    .}}
{\ovalbox{La calculatrice #1 .}} }}
```

   avec les appels suivants : \calculatrice[no] \calculatrice[est un piège à cons] \calculatrice

   La syntaxe de \ifthenelse : \ifthenelse{test}{à faire si test=true}{à faire sinon} .
   \equal{#1}{yes} vaut true si le paramètre 1 a pour valeur "yes".
   Consulter la doc de "ifthen" pour plus de détails.

**Exercice 12** (panneau interdire).

```
\usepackage{tikz}
\usetikzlibrary{shapes}     % ou \usepackage{pgflibraryshapes}
\scalebox{0.5}
{
\tikz{ \node[forbidden sign,draw=red,line width=1ex]{\textsc{calculatrice}}; }
}
```

**Exercice 13** (touches de calculatrice).

1. L'utilisation de l'instruction node de Tikz peut se faire ainsi :

   ```
   \begin{tikzpicture}
   \node[draw=black!25,rounded corners,double=black,fill=black!10]{9};
    \end{tikzpicture}
   ```

   ce qui donne [9]

2. On intègre cela dans une commande pour l'utiliser à volonté. On écrit donc dans le préambule :

   ```
   \usepackage{tikz}
   \newcommand{\Touc}[1]{%
   \begin{tikzpicture}
   \node[draw=black!25,rounded corners,double=black,fill=black!10]
     { #1 };% fin node
    \end{tikzpicture}
   } % fin commande
   ```

   On utilisera cette commande dans le corps du document ainsi :

   \Touc{5} , \Touc{$\sqrt{\ }$} , \Touc{.} ce qui donne [5] , [√] , ⊙ .

3. On voit tout de suite un problème : la boîte grise a une hauteur et une largeur dépendant du symbole placé dedans.

   (a) Règlons tout d'abord le problème de la largeur.

   On peut utiliser \makebox[largeur][position du texte dans la boîte]{matériel} .

   Les arguments entre crochets sont optionnels. [position du texte] est [c] pour "centré", [l] pour un début à gauche, [r] pour alignement à droite de la boîte.

   On peut maintenant proposer ceci :

   ```
   \newcommand{\Touch}[1]{%
   \begin{tikzpicture}
   \node[draw=black!25,rounded corners,double=black,fill=black!10]
     {\makebox[1.2em]{#1}};
   \end{tikzpicture}
   } % fin commande
   ```

   \Touch{9}, \Touch{$\sqrt{\ }$} et \Touch{.} donnent alors [9] , [√] et ⊙

(b) On cherche maintenant à ramener toutes les boîtes à la hauteur de la boîte d'un chiffre.

i. Lorsque le symbole est plus petit qu'un chiffre (exemple du point donné ci-dessus), le problème se règle facilement : on ajoute un symbole invisible dans la boîte. Cela oblige la boîte à avoir au moins la hauteur de ce symbole invisible (par contre la largeur du symbole invisible est considérée comme nulle, ainsi la présence d'un symbole fantôme n'augmente pas la largeur de la touche de calculatrice).

Ainsi

```
\begin{tikzpicture}
\node[draw=black!25,rounded corners,double=black,fill=black!10] { . };
\end{tikzpicture}
```

donne ⊙

alors que

```
\begin{tikzpicture}
\node[draw=black!25,rounded corners,double=black,fill=black!10]
{\vphantom{9}.};
 \end{tikzpicture}
```

donne .

ii. Lorsque les symboles sont plus grands qu'un chiffre, on cherche à les réduire à la hauteur d'un chiffre.

Pour cela, on peut utiliser \scalebox{échelle}{matériel} du package graphicx .

On peut alors par exemple donner la définition suivante de la commande :

```
\newcommand{\Touche}[2][1]{%
\begin{tikzpicture}
\node[draw=black!25,rounded corners,double=black,fill=black!10]
  {\makebox[1.2em]{\scalebox{#1}{#2}}};
\end{tikzpicture}
} % fin commande
```

\Touche{9} , \Touche[0.65]{$\sqrt{\ }$} donnent alors 9 et √

Une explication : dans la ligne \newcommand{\Touche}[2][1]{ le [2] signifie qu'il y a deux arguments. Le [1] rend le premier argument (#1) optionnel, avec une valeur par défaut égale à 1. Ainsi pour un appel sans réduction, on ne donne pas de valeur à l'argument 1 et lors d'un appel avec réduction on fait l'appel avec une valeur explicite pour l'argument 1 (dans l'exemple : [0.65] ).

iii. Il nous reste à regrouper en une seule commande les deux cas de figure. Par exemple ainsi :

```
\newcommand{\ToucheC}[2][1]{%
\begin{tikzpicture}
\node[draw=black!25,rounded corners,double=black,fill=black!10]
  {\makebox[1.2em]{\vphantom{9}\scalebox{#1}{#2}}};
```

```
\end{tikzpicture}
} % fin commande
```

\ToucheC{9} , \ToucheC[0.65]{$\sqrt{\ }$} et \ToucheC{.} donnent alors  ,  et 

4. On aimerait maintenant ne pas être obligé de tâtonner pour trouver un coefficient de réduction satisfaisant. Il s'agit donc maintenant de demander à LaTeX de faire ce travail tout seul.

   (a) Pour cela, on peut utiliser les instructions suivantes :

   \usepackage{calc}\heightof{texte}

   renvoie la hauteur (au-dessus de la ligne de base) de 'texte'

   \usepackage{graphicx}\resizebox*{dimension horizontale}{dimension verticale}{matériel}

   redimensionne "matériel" avec les dimensions imposées.

   Pour conserver le rapport vertical/horizontal, on précise la dimension que l'on veut imposer et on remplace l'autre par un ! .

```
\newcommand{\ToucheCa}[1]{%
\begin{tikzpicture}
\node[draw=black!25,rounded corners,double=black,fill=black!10]
  {\makebox[1.2em]{\resizebox*{!}{\heightof{9}}{#1}}};% fin node
 \end{tikzpicture}
} % fin commande
```

   \ToucheCa{9} , \ToucheCa{$\sqrt{\ }$} donnent alors  et  .

   Mais : \ToucheCa{.} donne  ... ce qui est normal : on a redimensionné le symbole à afficher à la hauteur d'un chiffre, ce qui est ce que l'on recherchait pour les symboles plus grands qu'un chiffre mais pas pour les plus petits.

   (b) Il nous reste donc à règler le problème pour les symboles plus petits. La tâche se complique puisqu'il nous faut demander à LaTeX de tester si le symbole est plus petit ou plus grand et d'agir en fonction du résultat.

```
% package pour faire des tests
\usepackage{ifthen}
% package pour faciliter l'écriture de certains calculs sur les longueurs :
\usepackage{calc}
% on déclare deux nouvelles longueurs :
\newlength{\DimA}  \newlength{\DimB}

\newcommand{\ToucheCal}[1]{%
% DimA = hauteur + profondeur d'un chiffre :
\setlength{\DimA}{\heightof{9}+\depthof{9}}
% DimB = hauteur + profondeur du symbole à afficher :
\setlength{\DimB}{\heightof{#1}+\depthof{#1}}
% on teste maintenant qui est le plus grand :
\ifthenelse{\lengthtest{\DimB>\DimA}}
{ % instruction à appliquer si DimB>DimA
```

```
% dans ce cas on rapetisse le symbole aux dimensions d'un chiffre
\begin{tikzpicture}
\node[draw=black!25,rounded corners,double=black,fill=black!10]
  {\makebox[1.2em]{\resizebox*{!}{\heightof{9}}{#1}}};% fin node
\end{tikzpicture}
} % fin cas DimB>DimA
{ % instruction à appliquer si DimB<=DimA
% dans ce cas on ne touche pas au symbole mais on ajoute un
% symbole fantôme (invisible) dans la boîte de la hauteur d'un chiffre
\begin{tikzpicture}
\node[draw=black!25,rounded corners,double=black,fill=black!10]
  {\makebox[1.2em]{\vphantom{9}#1}};% fin node
\end{tikzpicture}
}% fin cas DimB<=DimA
} % fin commande
```

\ToucheCal{9} , \ToucheCal{$\sqrt{\ }$} et \ToucheCal{$\cdot$} donnent alors [9] , [√] et [·] .

5. Il existe un package keystroke qui peut convenir également. On écrira par exemple `\keystroke{5}` , `\keystroke{$x^2$}` qui donneront [5] , [$x^2$] .
   Pour les affichages de la machine, on pourra utiliser des polices du "style affichage digital sur écran pauvre en pixels". Par exemple `\usepackage{ifsym}` , `\textifsym{123.58}` .

**Exercice 14** (triangle affichage casio).

```
%\usepackage{tikz}  dans le préambule
\newcommand{\trianglecasio}
{
\begin{tikzpicture}[scale=0.2]
\draw[fill] (0,0)--(1,0)--(1,1)--cycle;
\end{tikzpicture}
}
```