```
du coté d'un cube) et qui renvoie le volume du cube.
\end{Exofn}

\begin{Exofn}
Ecrire une fonction \texttt{prix\_TTC} qui a pour paramètres \texttt{prix\_
HT} (prix d'un article HT), \texttt{tva} (montant de la TVA en pourcent) et
qui renvoie le prix TTC de l'article.
\end{Exofn}

\begin{Exofn}
Ecrire une procédure qui a pour paramètres un entier n non nul, et qui affiche
n fois la phrase "Je dois être sage en classe !".
\end{Exofn}

\begin{Exofn}
Écrire une fonction \texttt{volumeSphere} qui calcule le volume d'une sphère
de rayon r fourni
en argument.
\end{Exofn}

\begin{Exofn}
Écrire une fonction \texttt{somme} avec trois entiers en paramètres, et qui
renvoie leur somme.
\end{Exofn}

\begin{Exofn}
Écrire une fonction \texttt{contient\_e} qui détermine si une chaîne contient
ou non le caractère « e ».
\end{Exofn}

\begin{Exomn}
Écrire une fonction \texttt{asterisque} qui recopie une chaîne (dans une
nouvelle variable), en insérant des astérisques entre les caractères.
Ainsi par exemple, « gaston » devra devenir « g*a*s*t*o*n ».
\end{Exomn}

\begin{Exomn}
Écrire une fonction \texttt{inverse} qui recopie une chaîne (dans une nouvelle
variable) en l'inversant.\\ Ainsi par exemple, \texttt{inverse('zorglub')} va
renvoyer \texttt{'bulgroz'}.
\end{Exomn}


\begin{Exomn}
Écrire une fonction \texttt{est\_palindrome} qui détermine si un mot est un
palindrome (\ding{42} utilisez l'exercice précédent).
Par exemple, \texttt{est\_palindrome('kayak')} va renvoyer \texttt{True}.
\end{Exomn}

\begin{Exomn}
Écrire une procédure \texttt{etoiles} qui a pour paramètre un entier
\texttt{n} non nul et qui affiche (n lignes):
\begin{center}
\includegraphics[scale=1]{python3_2.png}
\end{center}
\end{Exomn}


\begin{Exofn}
Ecrire une procédure \texttt{entiers} qui prend en paramètre un entier et
affiche tous les entiers entre 0 et n.
\end{Exofn}
```

```
\begin{Exomn}
Ecrire une fonction \texttt{somme} qui prend en paramètre un entier \texttt{n}
    et qui retourne la somme 0+1+2+...+n.\\
Par exemple, \texttt{somme(100)} va renvoyer \texttt{5050}.
\end{Exomn}

\begin{Exofn}
Ecrire une fonction \texttt{moyenne} qui prend en paramètres 4 réels et
renvoie la moyenne des nombres.
\end{Exofn}

\begin{Exofn}
\' Ecrire une fonction \texttt{de} qui simule le lancer d'un dé à 6 faces
\end{Exofn}


\begin{Exomn}
\' Ecrire une fonction \texttt{simul\_de} qui lance un dé à 6 faces 100 fois
et calcule la somme des faces obtenues
\end{Exomn}

\begin{Exomn}
\' Ecrire une fonction \texttt{simul\_de\_n} qui lance un dé à 6 faces n fois
et calcule la somme des faces obtenues.
\end{Exomn}


\begin{Exofn}
\' Ecrire une fonction qui prend en argument un entier $n$ et qui retourne
$1^2+2^2+3^2+...+n^2$
\end{Exofn}

\begin{Exomn}
Cette fonction prend comme arguent un réel positif $x$, et retourne le plus
petit entier supérieur à $x$.\\
   A réaliser sans fonction particulière :interdiction ici d'utiliser floor,
round...
\end{Exomn}


\begin{Exofn}
Écrire une fonction \texttt{nature} qui détermine le type de la valeur entrée
en argument (int,float,bool, str).\\
Par exemple, \texttt{nature('voiture')} va renvoyer \texttt{<class int>}.
\end{Exofn}


\begin{Exofn}
\' Ecrire une fonction \texttt{maximal\_1} qui prend deux entiers en arguments
et qui donne le plus grand des deux.
\end{Exofn}

\begin{Exomn}
\' Ecrire une fonction \texttt{maximal\_2} qui prend trois entiers en
arguments et qui donne le plus grand des trois.
\end{Exomn}

\begin{Exofn}
\' Ecrire une fonction qui prend un entier en argument et qui retourne
\texttt{True} si l'entier est un multiple de 10, \texttt{False} sinon.
\end{Exofn}
```

```
\begin{Exofn}
\' Ecrire une fonction qui prend un entier en argument et qui retourne
\texttt{True} si l'entier est pair, \texttt{False} sinon.
\end{Exofn}


\begin{Exofn}
\' Ecrire une fonction \texttt{maj} qui prend un chaîne de caractère en
argument et qui retourne la même chaine de caractère en majuscules.\\
Par exemple, \texttt{maj('Voiture')} va renvoyer \texttt{'VOITURE'}
\end{Exofn}


\begin{Exomn}
Un administrateur système veut assurer un maximum de sécurité pour les
utilisateurs de son site. Il décide de créer un fonction \texttt{force} qui
prend en argument une chaîne de caractères, et qui calcule la "force" d'un mot
de passe:
\begin{itemize}
\item Si le score est strictement inférieur à 20, Le mot de passe est "Très
faible"
\item Si le score est  supérieur à 20 et  strictement inférieur à 40, Le mot
de passe est "faible"
\item Si le score est  supérieur à 40 et  strictement inférieur à 80, Le mot
de passe est "fort"
\item Si le score est  supérieur à 80 , Le mot de passe est "très fort"
\end{itemize}
Le score est égale au quadruple du nombre de caractères du mot de passe.\\
\end{Exomn}

\begin{Exodn}
On reprend l'exercice précédent, mais on change les modalités pour calculer le
score, qui se calcule maintenant en additionnant des bonus :\\
\begin{itemize}
\item Nb de caractère * 4
\item (Nb total de caractère - nb de lettres majuscules ) * 2
\item Nb de caractères spéciaux * 4
\end{itemize}
\end{Exodn}

\begin{Exofn}
Ecrire une fonction qu'on nommera $f$ qui prend en entrée un nombre $x$ et qui
renvoie le résultat de $3x^2+4x-5$
\end{Exofn}

%\begin{Exofn}
%Ecrire une fonction qui incrémente de 2 la valeur passée en argument.
%\end{Exofn}


\begin{Exomn}
En 2018 la population mondiale est estimée à 7 577 millions (environ 7,6
milliards) . Le taux annuel de la croissance
démographique de la population mondiale est d'environ 1,2 \%. \\
\' Ecrire programme qui cherche à déterminer en quelle année, si cette
évolution se poursuit, la population mondiale
dépassera 10 milliards et quelle sera cette population
\end{Exomn}

\begin{Exofn}
Ecrire une fonction qui donne l'image de la fonction $f$ définie par :
```

```
\begin{itemize}
\item $f(x)=2x+3$ si $x<0$
\item $f(x)=3-x$ si $0 \leq x <2$
\item $f(x)=x^2-3$ si $x\geq2$
\end{itemize}
\end{Exofn}

\begin{Exomn}
Ecrire une fonction \texttt{moyenne} prenant en argument un entier\texttt{n}
positif.\\
Cette fonction demandera à l'utilisateur de rentrer une par une ses \texttt{n}
notes pour ensuite calculer la moyenne de ces notes.\\
Exple:
\begin{verbatim}
>>> moyenne(4)
Note 1 : 12
Note 2 : 14
Note 3 : 13
Note 4 : 16
13.75
\end{verbatim}
\end{Exomn}


\begin{Exofn}
Écrire une fonction \texttt{min\_str} qui prend en paramètre 2 chaînes de
caractères et qui renvoie la
plus petite des 2. Si les deux chaines ont la même longueur, on renvoie la
première.\\
Par exemple, \texttt{min\_str("voiture","bus")} va renvoyer \texttt{voiture}.
\end{Exofn}




\begin{Exofn}
Écrire une fonction \texttt{pair} qui prend en paramètre un nombre entier
naturel et qui affiche sa
parité (pair ou impair) et qui renvoie \texttt{True} si le nombre est pair et
\texttt{False} s'il est impair.
\end{Exofn}

\begin{Exomn}
Ecrire une fonction \texttt{factorielle} qui prend un entier \texttt{n} en
argument et qui renvoie le produit $1\times 2 \times 3...\times n$.\\
Par exemple, \texttt{factorielle(30)} renvoie
265252859812191058636308480000000.
\end{Exomn}



\begin{Exofn}
Écrire un programme qui demande les deux côtés adjacents à l'angle droit d'un
triangle rectangle et qui renvoie la longueur de l'hypoténuse.
\end{Exofn}

\newpage
\section{Les séquences}


Il est possible de "stocker" plusieurs grandeurs dans une même structure, ce
type de structure est appelé une \mc{séquence}.\\
```

```
\begin{Def}
De façon plus précise, nous définirons une séquence comme un ensemble fini et
ordonné d'éléments indicés de 0 à n-1 (si cette séquence comporte n éléments).
\end{Def}

Nous commencerons par étudier un premier "type" de séquence, les \mc{tuples},
puis nous enchainerons sur les \mc{tableaux}.

\subsection{Les tuples en Python}

\begin{verbatim}
mon_tuple = (5, 8, 6, 9)
\end{verbatim}


\begin{verbatim}
type(mon_tuple)
\end{verbatim}
Chaque élément du tuple est indicé.

\ding{43} Retrouver comment accéder au premier élément, au second..etc...

\danger dans les séquences les indices commencent toujours à 0 (le premier
élément de la séquence a pour indice 0)\\



\ding{43} Compléter le programme suivant:
\begin{verbatim}
mon_tuple = (Duthoit,Jean-Baptiste)
print(f"Mon nom est {mon_tuple[...]} et mon prénom {mon_tuple[...]}")
\end{verbatim}


\begin{Def}
Un \mc{tuple} est une séquence \mc{immuable} (c'est à dire qui ne peut pas
être modifiée),  pouvant contenir plusieurs autres objets.
\end{Def}

\begin{Exofn}
Créer une fonction nommée \texttt{distance}, qui prend en argument deux tuples
(Dans chaque tuple, se trouvent les coordonnées (x;y) d'un point) et qui
renvoie la distance entre ces deux points, dans un repère orthonormé.
\end{Exofn}


\begin{Exofn}
On considère un tuple de tuple: \\
(('Alim',17),('Pierre',34),('Sophie',19),('Angel',23))\\
Afficher 'bonjour Alim tu as 17 ans', et ainsi de suite pour les autres.
\end{Exofn}


\begin{Exofn}
Les commandes sum(tuple),len,min,max donnent respectivement la somme, le
nombre d'éléments, le min et le max d'un tuple.\\
Ecrire une fonction qui renvoie le tuple composé du min, max, moyenne.
\end{Exofn}

\subsection{Les tableaux en Python}
```

```
\subsubsection{Création d'un tableau vide}


Il existe deux moyens : \\
Avec la commande \cmd{list} :

\begin{verbatim}
mon_tableau = list()
\end{verbatim}

ou bien

\begin{verbatim}
mon_tableau = []
\end{verbatim}

\danger Python utilise le terme "list", mais il s'agit en réalité d'un tableau
(tableau dynamique en réalité)

\subsubsection{Création d'une tableau non vide}

\begin{verbatim}
mon_tableau_1 = [1, 2, 3, 4]
mon_tableau_2 = [36, "voiture", 2.8, True]
mon_tableau_3 = [36, "texte", 2.8, mon_tableau_1]
\end{verbatim}

\subsubsection{Accès au éléments}

\begin{verbatim}
mon_tableau_1[2] # pour obtenir 3
\end{verbatim}

\subsubsection{Modification de tableau}

\danger Ce que ne permet pas un tuple :-)

\begin{verbatim}
mon_tableau_1[2] = 120 # pour remplacer 3 par 120
\end{verbatim}

\subsubsection{Ajout d'un élément}

On utilisera ici la méthode \texttt{append} :
\begin{verbatim}
mon_tableau = [1,2,3,4]
mon_tableau.append(5)
\end{verbatim}


\danger La méthode "append" est très utilisée et permet d'ajouter un élément
\textbf{à la fin} du tableau.

\subsubsection{Suppression d'éléments}

On utilise ici la commande \cmd{del}:

\begin{verbatim}
mon_tableau = [1,2,3,4,5,6]
del mon_tableau[1] # suppression du second élément
\end{verbatim}
```

```
\subsubsection{Longueur d'un tableau}

On utilise ici la commande \cmd{len}:
\begin{verbatim}
mon_tableau = [1,2,3,4,5,6]
len(mon_tableau) # affiche 6
\end{verbatim}

\subsubsection{Parcourir un tableau}


\begin{verbatimtab}
mon_tableau = [1,2,3,4,5,6,7,8,9,10]
for element in mon_tableau:
    print(element)
\end{verbatimtab}

\subsection{Des tableaux et des chaînes}

\subsubsection{Des tableaux aux chaînes}
\ding{42} On utilise ici la commande \cmd{join} :
\begin{verbatim}
mon_tableau = ['Bonjour','à','tous']
a = " ".join(mon_tableau)
\end{verbatim}

\ding{43} Vérifiez que a est un str



\subsubsection{Des chaînes aux tableaux}

\ding{42} On utilise ici la commande \cmd{split} :
\begin{verbatim}
ma_chaine = " Bonjour tout le monde !"
a = ma_chaine.split(" ")
\end{verbatim}

\ding{43} Vérifiez que a est du type list
\ding{43} La méthode split() découpe donc la chaîne en fonction du paramètre
donné.


\begin{Def}
Un tableau (list en Python) est une séquence \textbf{mutable} pouvant contenir
plusieurs autres objets.
\end{Def}

\begin{Exofn}
Ecrire une fonction qui prend un tableau et un entier en argument et qui
renvoie le tableau où l'entier a été ajouté à la fin
\end{Exofn}


\danger Remarquez bien que la liste a été modifiée en place.



\begin{Exofn}
Écrire une fonction (sans paramètre) qui remplit un tableau avec les entiers
de 0 à 20
\end{Exofn}
```

```
\begin{Exofn}
Ecrire une fonction(sans paramètre) qui remplit un tableau avec 20 entiers
choisis au hasard entre 0 et 100
\end{Exofn}


\begin{Exofn}
Ecrire une fonction prenant en argument un entier n et  qui remplit un tableau
avec n entiers choisis au hasard entre 0 et 100
\end{Exofn}

\ding{43} Ne pas oublier : from random import *


\begin{Exofn}
Ecrire un programme prend en argument un tableau d'entiers et qui retourne un
autre tableau où chaque valeur a été doublée :
\begin{figure}[H]
\centering
\includegraphics[scale=1]{python_4_1.png}
\end{figure}
\end{Exofn}


\begin{Exofn}
Ecrire une fonction qui prend en argument un tableau d'entiers d'au moins 2
éléments  et qui retourne le tableau dans lequel on a échangé le premier et le
dernier élément.
\begin{figure}[H]
\centering
\includegraphics[scale=1]{python_4_2.png}
\end{figure}
\end{Exofn}


\begin{Exofn}
Écrire une fonction qui prend en argument un tableau d'entiers , et qui
retourne un nouveau tableau avec les carrées des entiers.
\end{Exofn}

\begin{Exofn}
Écrire une fonction qui prend en argument un tableau d'entiers avec au moins
deux éléments, et qui retourne le tableau avec les deux premiers termes
supprimés.
\end{Exofn}


\begin{Exofn}
Écrire une fonction qui prend en argument un tableau d'entiers et qui retourne
un tuple (n,p) où n est le nombre d'entiers négatifs et p le nombre d'entiers
positifs.
\end{Exofn}

\begin{Exofn}
Écrire une fonction qui prend en argument un tableau contenant des chaines de
caractères, et qui retourne une autre liste contenant les mots de plus de 5
caractères.
\end{Exofn}


\begin{Exomn}
```