

TRONCATURE À DEUX DÉCIMALES

On demande d'écrire un algorithme qui, à partir d'un nombre positif x , donne sa troncature y à deux décimales. Par exemple :

Si $x =$	alors $y =$
π	3,14
1,235	1,23
1,9999	1,99
$\sqrt{2}$	1,41

On va donc demander à l'utilisateur d'entrer x et à la fin, on va afficher y . L'algorithme (à compléter) peut alors s'écrire

Algorithme 0.1: TRONCATURE À 10^{-2} PRÈS()

variables réelles x, y

Entrer (x)

.....

$y \leftarrow$

.....

Afficher (y)

On choisira l'outil de programmation que l'on veut, mais pas la Ti82! Par exemple, au choix ci-dessous, *Algobox*, *Python*, *CaRMetal*, etc.

Algobox

On peut compléter l'algorithme suivant (éventuellement en ajoutant des lignes vides) :

```

1  VARIABLES
2    x EST_DU_TYPE NOMBRE
3    y EST_DU_TYPE NOMBRE
4  DEBUT_ALGORITHME
5    LIRE x
6    y PREND_LA_VALEUR .....
7    AFFICHER y
8  FIN_ALGORITHME
    
```

Python

On peut compléter la définition de fonction $tronc2(x)$ suivante (attention à respecter la position des débuts de lignes) :

```

def tronc2(x) :

    y=

    return y

print(tronc2(1.9999))
    
```

JavaScript

Avec *CaRMetal*, on peut ouvrir la console *JavaScript* et y entrer la fonction *tronc2* suivante (à compléter) :

```
function tronc2(x){
var y=

return y;
}

x=Input("Entrer le nombre à tronquer");
y=tronc2(x);
Println("La troncature de "+x+" à 2 décimales est "+y);
```

Syntaxe éventuellement utile pour ce TP

Les outils présentés ci-dessous peuvent arrondir ou tronquer à l'entier le plus proche (ou inférieur). Voici leur syntaxe :

Langage	Arrondi	Troncature
AlgoBox	<code>round(x)</code>	<code>floor(x)</code>
JavaScript	<code>Math.round(x)</code>	<code>Math.floor(x)</code>
Python	<code>round(x,0)</code>	<code>int(x)</code>

Et la syntaxe de π et de $\sqrt{2}$:

Langage	π	$\sqrt{2}$
AlgoBox	<code>Math.PI</code>	<code>sqrt(2)</code>
JavaScript	<code>Math.PI</code>	<code>Math.sqrt(2)</code>

Avec *Python*, on peut faire ceci :

```
from math import pi, sqrt

print(pi)
print(sqrt(2))
```