

# Mesures de température [tb08] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 19 mai 2018

## Table des matières

<b>1 Mesures de température / pgmesures</b>	<b>2</b>
1.1 Saisie et affichage . . . . .	2
1.2 Conversion Celsius de températures . . . . .	2
1.3 Moyenne, Variance, Écart-type . . . . .	4
1.4 Centrage des données . . . . .	6
1.5 Extremums . . . . .	6
1.6 Mise en place du tout . . . . .	7

## alg - Mesures de température (TD)



**Mots-Clés** Tableau unidimensionnel ■

**Requis** Structures de base, Structures conditionnelles, Algorithmes paramétrés, Structures répétitives, Schéma itératif ■

**Difficulté** ●●○ (2 h 30 à 3 h) ■



### Objectif

Cet exercice réalise une analyse statistique de mesures de température sur une année. Les données sont stockées dans un tableau pour les étudier.

# 1 Mesures de température / pgmesures

## 1.1 Saisie et affichage



Définissez la constante entière `TMAX=366` (nombre maximum de mesures, une par jour pour une année) et le type `Mesures` comme étant un tableau de `TMAX` réels.



Écrivez une fonction `saisirMesures(t)` qui saisit le nombre de mesures dans un entier `n` qui doit être compris entre 1 et `TMAX` puis effectue la saisie de `n` températures dans un `Mesures t`. La fonction renvoie le nombre de températures saisies, à savoir `n`. Affichez les invites comme dans l'extrait d'exécution suivant :

```
Nombre de mesures dans [1..[TMAX]]? 5
t[1]? -3.5
...
t[5]? 10.3
```

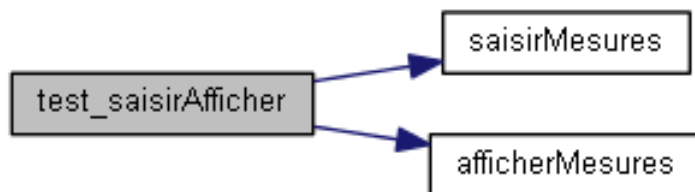


Écrivez une procédure `afficherMesures(t,n)` qui affiche les `n` températures d'un `Mesures t`. Affichez les valeurs à la queue-leu-leu séparées par un espace, le tout entre crochet, comme suit :

```
[-3.5 ... 10.3]
```



Écrivez une procédure de test `test_saisirAfficher` de la saisie et de l'affichage.



## 1.2 Conversion Celsius de températures

L'unité de température, par exemple pour la météo, change selon les pays. Dans certains pays (États-Unis et certains pays anglophones), on utilise l'échelle FAHRENHEIT, dans d'autres l'échelle CELSIUS. Dans ce problème, on suppose que les températures ont été exprimées en degré FAHRENHEIT.



**Propriété**

L'équivalent Celsius<sup>1</sup>  $C$  de  $F$  degrés Fahrenheit<sup>2</sup> est :

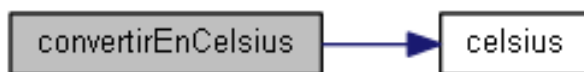
$$C = \frac{5}{9}(F - 32)$$



Écrivez une fonction `celsius(x)` qui calcule et renvoie l'équivalent Celsius de  $x$  (réel) degrés Fahrenheit.



Écrivez une procédure `convertirEnCelsius(t,n)` qui convertit  $n$  températures d'un Mesures  $t$  de degrés FAHRENHEIT en degrés CELSIUS.



Écrivez une fonction `arrondi1(x)` qui renvoie l'arrondi à une décimale d'un réel  $x$ .

$$\text{arrondi1}(x) = \begin{cases} \lfloor x * 10 + 0.5 \rfloor / 10 & \text{si } x \geq 0 \\ \lfloor x * 10 - 0.5 \rfloor / 10 & \text{sinon} \end{cases}$$



Écrivez une procédure `arrondirValeurs1(t,n)` qui convertit  $n$  températures d'un Mesures  $t$  à une décimale.



Copiez/collez la procédure `afficherMesures` en la procédure `afficherInMesures(t,n)` puis modifiez-la pour qu'elle affiche les températures, une valeur par ligne.

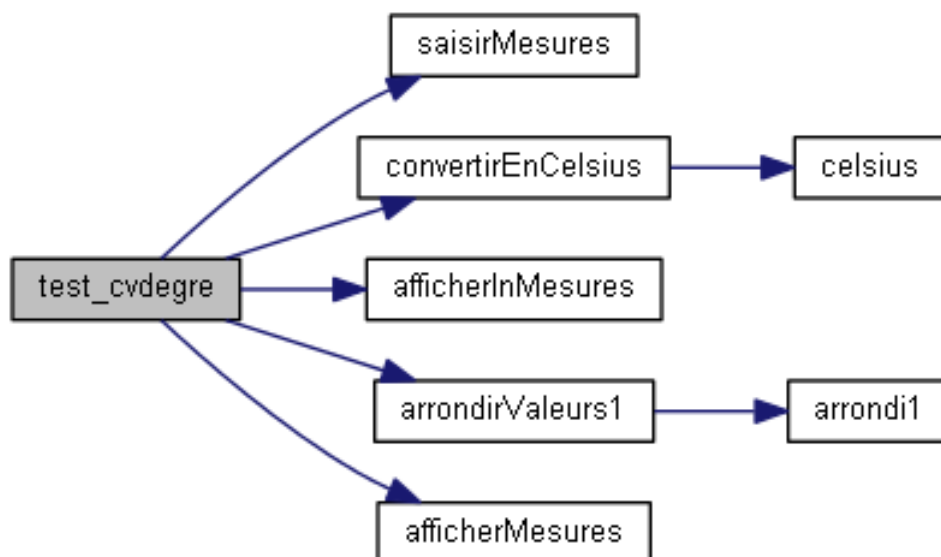


Écrivez une procédure `test_cvdegre` qui :

- Stocke des températures dans un Mesures.
- Convertit les mesures en degrés CELSIUS.
- Puis les affichent une valeur par ligne.
- Arrondit à une décimale ces mesures.
- Enfin les affichent à nouveau à la queue-leu-leu.

1. Appartient au système international des unités depuis 1948, son nom est une référence à l'astronome et physicien suédois ANDERS CELSIUS, inventeur en 1742 d'une des premières échelles centigrades de température

2. Proposée en 1724 par le physicien allemand DANIEL GABRIEL FAHRENHEIT



### 1.3 Moyenne, Variance, Écart-type



#### Définition

La **moyenne arithmétique** de  $n$  valeurs  $x_j$  est définie par :

$$moyenne = \frac{1}{n} \sum_{j=1}^n x_j$$



Écrivez une fonction `sommeMesures(t,n)` qui calcule et renvoie la somme des  $n$  températures d'un `Mesures t`.

$$somme = \sum t[k]$$



Déduisez une fonction `moyenneMesures(t,n)` qui calcule et renvoie la moyenne des  $n$  températures d'un `Mesures t`.



#### Définition

La **variance** de  $n$  valeurs  $x_j$  est définie par :

$$variance = \frac{1}{n} \sum_{j=1}^n (x_j - moyenne)^2$$



Écrivez une fonction `varianceMesures(t,n)` qui calcule et renvoie la variance des  $n$  températures d'un `Mesures t`.

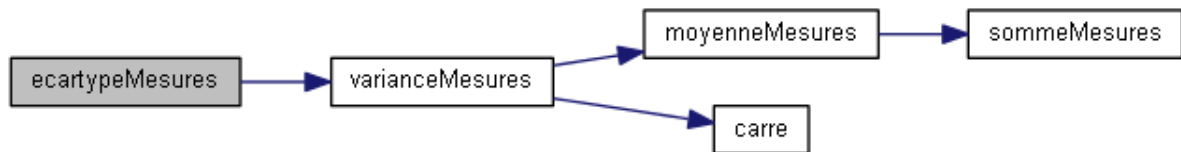


### Définition

L'**écart-type** est le nombre  $s = \sqrt{\text{variance}}$ .



Déduisez une fonction `ecartypeMesures(t,n)` qui calcule et renvoie l'écart-type des  $n$  températures d'un `Mesures t`.



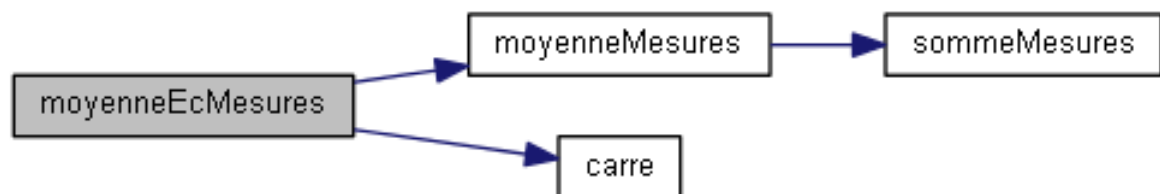
### Outil alg

L'opération  $\sqrt{x}$  s'écrit `RacineCarrée(x)`.



Finalement, écrivez une procédure `moyenneEcMesures(t,n,moy,ec)` qui calcule la moyenne dans `moy` (réel) et l'écart-type dans `ec` (réel) des  $n$  températures d'un `Mesures t`.

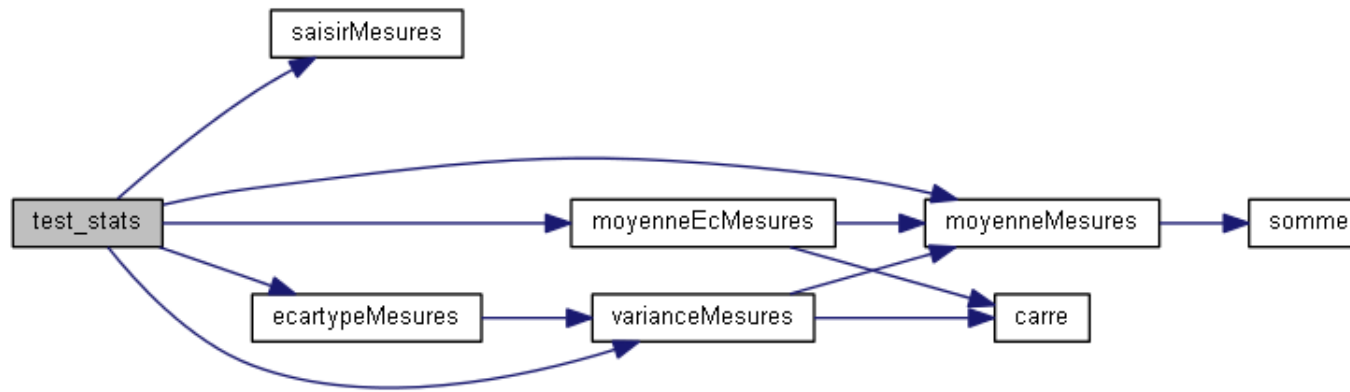
$$\begin{aligned} \text{moyenne} &= \frac{1}{n} \sum t[k] \\ \text{variance} &= \frac{1}{n} \sum (t[k])^2 - \text{moyenne}^2 \\ \text{ecartype} &= \sqrt{\text{variance}} \end{aligned}$$



Écrivez une procédure de test `test_stats` :

```

==> La moyenne est ...
==> La variance est ...
==> L'écart-type est ...
==> Moyenne bis est ...
==> Ecart-type bis est ...
  
```



## 1.4 Centrage des données

Il est parfois utile de **centrer** les séries statistiques. Cette opération consiste à remplacer chaque valeur de la série par sa valeur centrée égale à la différence de sa valeur initiale moins la moyenne de la série. Soit l'exemple suivant (en arrondissant au centième) :

- Série initiale : (15,78,34,28,2,90)
- Moyenne : 41.17
- Série centrée : (-26.14,36.83,-7.17,-13.17,-39.17,48.83)

### Remarque

La moyenne des valeurs d'une série centrée est égale à 0, si bien qu'elle contient forcément des valeurs négatives, sauf si la série d'origine n'était composée que de 0.



Écrivez une procédure `centrerMesures(t,n)` qui réalise le centrage des  $n$  températures d'un `lstinline@Mesures t@`.

## 1.5 Extremums



Écrivez une procédure `minmaxMesures(t,n,tmin,tmax)` qui calcule les extrémums des  $n$  températures d'un `Mesures t` et restitue la valeur minimale dans `tmin` (réel) et la valeur maximale dans `tmax` (réel). Utilisez une seule boucle.



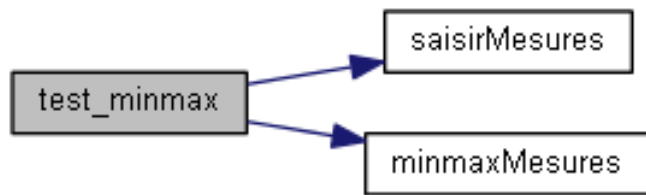
Écrivez une fonction `nTempSup(t,n,valeur)` qui calcule et renvoie le nombre de valeurs supérieures ou égales à une température `valeur` (réel) parmi les  $n$  températures d'un `Mesures t`.



Écrivez une procédure de test `test_minmax` :

```

==> Le minimum est ...
==> Le maximum est ...
  
```



## 1.6 Mise en place du tout

Ce problème définit une procédure qui permet à l'utilisateur d'effectuer une action sur ses données.



Écrivez une fonction `menu` qui affiche le menu suivant puis saisit et renvoie le choix de l'utilisateur (entier).

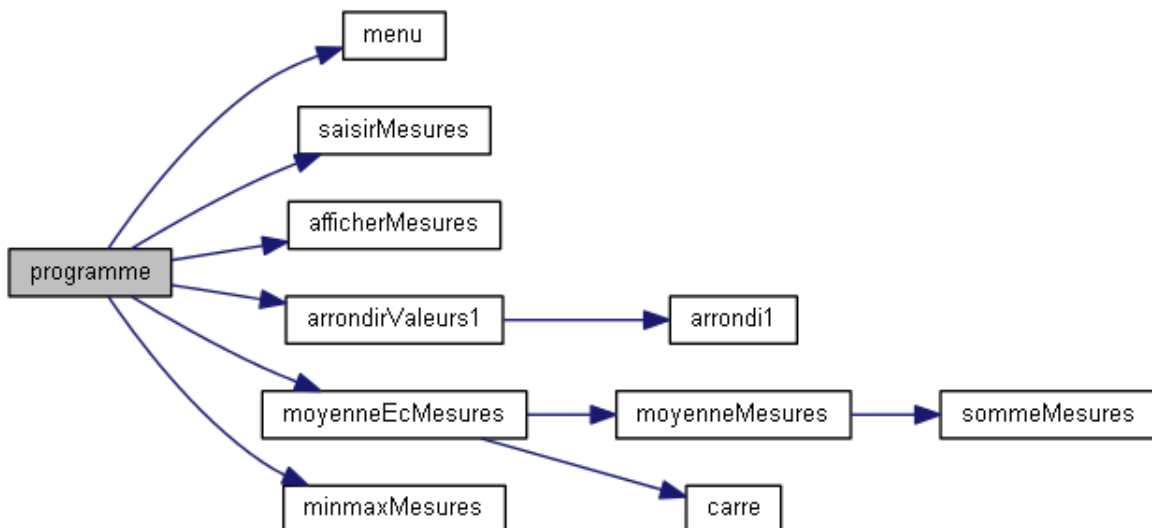
Voulez-vous:

```

1 entrer des températures
2 convertir des températures
3 afficher des statistiques
4 afficher les extremums
5 quitter
Votre choix?
  
```



Écrivez une procédure `programme` qui saisit le choix de l'utilisateur et effectue l'action souhaitée jusqu'à « plus soif » c.-à-d. jusqu'à ce que l'utilisateur quitte le programme.



### Aide simple

On ne peut convertir ou calculer que si les données ont été saisies (booléen `donneeslues`).