

# Structures conditionnelles

## Exercice #1:

Pour chacune des instructions suivantes, validez chaque proposition en mettant dans la case correspondante la lettre **V** si elle est correcte ou **F** dans le cas contraire.

a/ soit le pseudo-code suivant :

```

si  $x = y$  alors
     $z \leftarrow \text{vrai}$ 
sinon
     $z \leftarrow \text{faux}$ 
finsi
  
```

Nous pouvons écrire le même code sans l'utilisation de la structure conditionnelle, par :

$z \leftarrow x \neq y$	
$z \leftarrow x = y$	
$z \leftarrow x = y \text{ ou } x \neq y$	

b/ soit le pseudo-code suivant :

```

si  $x = y$  alors
     $z \leftarrow z + 1$ 
sinon si  $x > y$ 
     $z \leftarrow z + 1$ 
sinon
     $z \leftarrow z - 1$ 
finsi
  
```

Nous pouvons écrire le même code en simplifiant l'utilisation de la structure conditionnelle, par :

<pre> <i>si</i> <math>x = y \text{ ou } x &gt; y</math> <i>alors</i>     <math>z \leftarrow z + 1</math> <i>sinon</i>     <math>z \leftarrow z - 1</math> <i>fin si</i>       </pre>	
<pre> <i>si</i> <math>x = y \text{ et } x &gt; y</math> <i>alors</i>     <math>z \leftarrow z + 1</math> <i>sinon</i>     <math>z \leftarrow z - 1</math> <i>fin si</i>       </pre>	

```

z ← z - 1
si x = y ou x > y alors
    z ← z + 2
fin si

```

**Exercice #2: racine carré**

Écrire un algorithme d'un programme qui lit un entier  $n$ , puis calcule et affiche:

$$\begin{cases} \sqrt{n}, si n \geq 0 \\ \text{'erreur'}, sinon \end{cases}$$

**Exercice #3: divisible par 5**

Écrire un algorithme d'un programme qui lit un entier  $n$ , affiche le premier entier inférieur ou égale à  $n$  qui est divisible par 5.

Exemple:

- Pour  $n=73$ , le programme affiche 70.
- Pour  $n=12$ , le programme affiche 10.
- Pour  $n=35$ , le programme affiche 35.

**Exercice #4: santé**

Écrire l'algorithme et la traduction Python d'un programme qui lit la masse et la taille d'une personne puis calcule son BMI (Body Mass Index):

$$BMI = \frac{\text{masse}}{\text{taille}^2}$$

et selon le BMI calculé, le programme affiche l'interprétation:

BMI	Interprétation
Moins de 16,5	dénutrition
16,5 à 18,5	maigreur
18,5 à 25	Poids idéal
25 à 30	surpoids
30 à 35	Obésité modérée
35 à 40	Obésité sévère
Plus de 40	Obésité morbide

Source [https://fr.wikipedia.org/wiki/Indice\\_de\\_masse\\_corporelle](https://fr.wikipedia.org/wiki/Indice_de_masse_corporelle)

**Exercice #5: chiffre, voyelle,consonne ou symbole?**

Écrivez un algorithme et son implémentation en Python d'un programme qui lit un caractère, puis détermine et affiche s'il est un chiffre ou une voyelle ou une consonne ou un symbole.

**Exercice #6: résolution d'un polynôme de second degré**

Une équation du deuxième degré (ou équation du second degré) est de la forme  $ax^2+bx+c=0$  .

écrire un algorithme d'un programme qui lit les trois coefficients  $a$  ,  $b$  et  $c$  puis affiche les solutions cette équation. On rappelle qu'on peut avoir (dans  $\mathbb{R}$  ) : deux solutions, ou une solution ou pas de solution.

**Exercice #7: Nature triangle**

Écrivez un algorithme lit trois coordonnées des trois points A, B et C d'un triangle dans un plan cartésien et affiche sa nature (équilatérale, isocèle, rectangle, quelconque)

**Exercice #8: entier palindrome**

Un entier de quatre chiffre est dit palindrome, si et seulement si le chiffre des unités est égale au chiffre des milliers et le chiffres des dizaines est égale au chiffre des centaines.

**Exemples:**

- 5005 est palindrome
- 1221 est palindrome
- 3224 n'est pas palindrome
- 3283 n'est pas palindrome
- 1234 n'est pas palindrome

Écrire un algorithme d'un programme qui lit un entier de quatre chiffres, puis détermine et affiche s'il est palindrome ou pas.

**Exercice #9: entier cubique**

Un entier naturel de trois chiffres est dit cubique s'il est égal à la somme des cubes de ses chiffres.

**Exemples :**

153 est cubique car  $153=1^3+5^3+3^3$

370 est cubique car  $370=3^3+7^3+0^3$

801 n'est pas cubique par ce que  $801 \neq 8^3+0^3+1^3$

Écrire un algorithme d'un programme permettant de saisir un entier n puis d'afficher s'il est cubique ou non.

**Exercice #10: Nombre de jours**

a/ Écrire un algorithme d'un programme qui lit le numéro du mois, puis détermine et affiche le nombre de jours du mois saisi.

**Exemples:**

$mois = 1$  , le programme affiche 31

$mois = 2$  , le programme affiche 28,29

$mois = 6$  , le programme affiche 30

b/ Suggérer une amélioration pour le mois de Février?

**Exercice #11: Nombre de jours amélioré**

Une année bissextile (ou un an bissextile) est une année de 366 jours au lieu de 365 pour une année commune. Le jour ajouté est le 29 parce que ce mois compte habituellement vingt-huit jours dans le calendrier grégorien.

**Les années bissextiles sont des multiples de quatre cent, ou bien des multiples de quatre, mais à l'exception de ceux qui ne sont pas des multiples de 100.**

Écrire un algorithme d'un programme qui lit le numéro du mois, puis détermine et affiche le nombre de jours du mois saisi, en tenant compte l'année est bissextile ou pas pour le mois de Février.

**Exercice #12: Racine numérique**

Le **racine numérique**(*digital root*) d'un entier naturel ou **résidu** est le nombre obtenu en additionnant tous les chiffres du nombre initial, puis en additionnant les chiffres du résultat, et ainsi de suite jusqu'à l'obtention d'un nombre à un seul chiffre.

Par exemple, dans le cas du nombre 65536, le résultat est 7, car  $6 + 5 + 5 + 3 + 6 = 25$ , puis  $2 + 5 = 7$ .

Écrire un algorithme qui lit un entier, puis calcule et affiche sa racine numérique.

**Exercice #13: Nombre oblong**

En arithmétique géométrique, un nombre oblong, ou nombre pronique ou nombre hétéromécique, **est le produit de deux entiers naturels consécutifs.**

Les vingt premiers nombres oblongs

sont : 0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342 et 380

**Exemples:**

12 est oblong, car  $12 = 3 \times 4$

342 est oblong, car  $342 = 18 \times 19$

345 n'est pas oblong, parce **qu'il n'existe pas deux entiers consécutifs** dont leur produit est égale  
 345 . Écrire un algorithme qui lit un entier, puis détermine et affiche s'il est oblong ou pas.

### Exercice #14: sur la droite!

Dans un plan cartésien on vous donne les coordonnées des trois points  $a$  ,  $b$  et  $c$  .

On veut savoir si la point  $c$  appartient à la droite  $(a,b)$  **oui** ou **non**.

#### Travail demandé

Écrire un algorithme d'un programme qui lit:

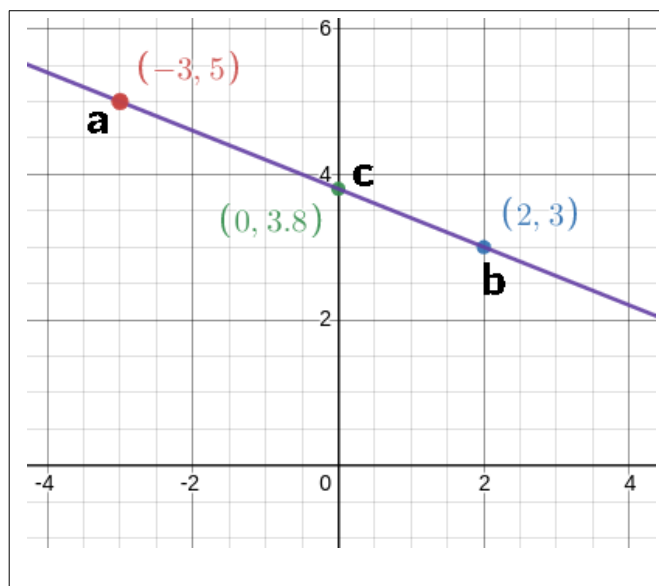
- les cordonnées  $x_a, y_a$  du point  $a$  ,
- les cordonnées  $x_b, y_b$  du point  $b$  ,
- les cordonnées  $x_c, y_c$  du point  $c$  ,

Puis affiche:

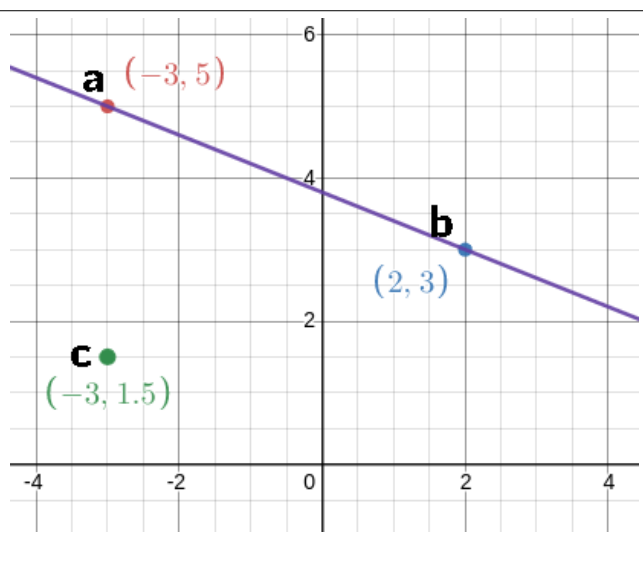
- "OUI", si  $c \in (a,b)$
- "NON", dans le cas contraire

#### Exemples

$a(-3,5)$ $b(2,3)$ $c(0,3.8)$	<pre>Point A?:-3 5 Point B?:2 3 Point C?:0 3.8 OUI</pre>	$a(-3,5)$ $b(2,3)$ $c(-3,1.5)$	<pre>Point A?:-3 5 Point B?:2 3 Point C?:-3 1.5 NON</pre>
-------------------------------------	--	--------------------------------------	---



Le programme affiche : "OUI"



Le programme affiche : "NON"

### Exercice #15: Conversion d'horloge

Le format 24 heures divise la journée en 24 heures à partir de 00 au 23, dont chacun dispose de 60 minutes de 00 à 59.

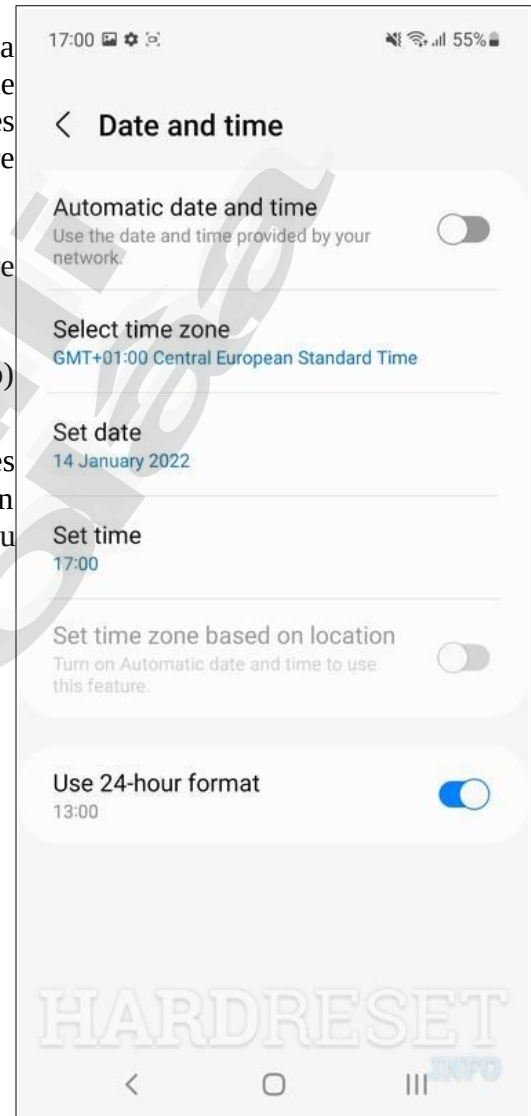
Le format 12 heures divise la journée en deux moitiés: la première moitié est le **AM** (Ante Meriduum), et la seconde moitié est **PM** (Post Meriduum). Dans chaque moitié, les heures sont numérotées dans l'ordre 12, 01, 02, 03,...,11. Chaque heure comporte 60 minutes numérotées de 00 à 59.

Étant donné l'heure au format 24 heures, affichez l'heure équivalente au format 12 heures.

Écrire un algorithme d'un programme qui lit (d'un seul coup) l'heure au format 24 heures au format "**hh:mm**". Où **hh** représente l'heure de 00 à 23 et **mm** représente les minutes de 00 à 59. Puis convertir l'heure saisie en format 24 heures en son équivalent en format 12 heures au format "**hh:mm AM**" ou "**hh:mm PM**"

#### exemples:

Format 24 heures	Son équivalent en format 12 heures
09:41	09:41 AM
18:06	06:06 PM
12:14	12:14 PM
00:59	12:59 AM
00:00	12:00 AM
14:34	02:34 PM
01:01	01:01 AM
19:07	07:07 PM
11:59	11:59 AM
12:00	12:00 PM
21:37	09:37 PM



## Application: Déterminer le jour

En utilisant la formule de Christian ZELLER(1822-1899), on peut déterminer le jour de la semaine qui correspond à une date donnée en passant par trois étapes:

- calcul de  $z = q + \text{ent}\left(2.6 \times (m+1)\right) + aa + \text{ent}\left(\frac{aa}{4}\right) + \text{ent}\left(\frac{ss}{4}\right) - 2 \times ss$
- calculer le rang du jour: le reste de la division euclidienne de  $z$  par 7
- déterminer le jour selon la rang calculé

source: [https://en.wikipedia.org/wiki/Zeller%27s\\_congruence](https://en.wikipedia.org/wiki/Zeller%27s_congruence)

<https://www.themathdoctors.org/zellers-rule-what-day-of-the-week-is-it/>

### calcul de z

$$z = q + \text{ent}\left(2.6 \times (m+1)\right) + aa + \text{ent}\left(\frac{aa}{4}\right) + \text{ent}\left(\frac{ss}{4}\right) - 2 \times ss$$

avec:

- $q$  : le quantième du mois,
- $m$  : le mois
- $ss$  : l'année séculaire (les deux premiers chiffres de l'année),
- $aa$  : l'année spécifique (les deux derniers chiffres de l'année)

Pour une date donnée: *jour/mois/année* , on a:

**N.B.: l'année  $\geq 1583$**

$$q = \text{jour}$$

$$m = \begin{cases} \text{mois}, & \text{si } \text{mois} \in [3, 12] \\ 13, & \text{si } \text{mois} = 1 \\ 14, & \text{si } \text{mois} = 2 \end{cases}$$

pour calculer  $ss$  et  $aa$  , il faut, d'abord, calculer  $ssaa$  :

$$ssaa = \begin{cases} \text{année}, & \text{si } \text{mois} \in [3, 12] \\ \text{année} - 1, & \text{si } \text{mois} = 1 \text{ ou } \text{mois} = 2 \end{cases}$$

après avoir calculé  $ssaa$  , il est facile de calculer  $ss$  et  $aa$  .



## Calcul du le rang du jour

Une fois  $z$  est calculé, nous devons calculer le rang du jour qui est le reste de la division euclidienne de  $z$  par 7 :

**N.B.:** Certain langage de programmation donne un résultat étonné si le dividende de la division euclidienne est négatif, donc il faut savoir d'abord si  $z$  est négatif ou non et est-ce-que ça pose un problème dans le langage de programmation que vous utilisez.

## Détermination du jour

Puisqu'on divise par 7, le rang du jour est toujours dans l'intervalle  $[0,6]$ . Donc, nous pouvons déterminer le jour de la semaine selon le tableau de correspondance suivant donné par M. ZELLER:

Rang jour	1	2	3	4	5	6	0
jour	Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi

Avant de passer à la programmation, Compléter le tableau ci-dessous et de déterminer le jour de la semaine des deux derniers dates:

	01/05/1902	15/02/2000	10/10/2010	20/01/1999
<b>q</b>	1	15		
<b>m</b>	5	14		
<b>ss</b>	19	19		
<b>aa</b>	2	99		
<b>z</b>	-16	143		
<b>Rang jour</b>	5	3		
<b>jour</b>	Jeudi	Mardi		

## La programmation

Faire des calculs en utilisant une calculatrice c'est pénible. Nous allons automatiser le calcul via un programme.

Écrivez un algorithme d'un programme qui lit une date de la forme `jj/mm/aaaa`, puis détermine le jour de la semaine qui correspond à cette date. (il est garanti que la date saisie est toujours valide)

```
Donner une date (de la forme jj/mm/aaaa):
01/05/1902
Jeudi
```

```
Donner une date (de la forme jj/mm/aaaa):
15/02/2000
Mardi
```