

Exercice 1

- Écrire la fonction **estPositif**
 - elle attend un paramètre **n** de type entier
 - elle retourne **Vrai** si l'entier **n** est positif ou nul, **Faux** si l'entier **n** est négatif
- Écrire un programme qui demande à l'utilisateur la saisie d'un entier **a** puis affiche :
 - "positif ou nul" si l'entier **a** est positif ou nul
 - "négatif" si l'entier **a** est négatif

On utilisera la fonction **estPositif**.

Exercice 2

- Écrire la fonction **calculerAbsolu**
 - elle attend un paramètre **n** de type entier
 - elle retourne la valeur absolue de l'entier **n**
- Écrire un programme qui demande à l'utilisateur la saisie d'un entier **a** puis affiche la valeur absolue de ce nombre. On utilisera la fonction **calculerAbsolu**.

Exercice 3

- Écrire la fonction **calculerMax2**
 - elle attend 2 paramètres entiers **a** et **b**
 - elle retourne le plus grand des 2
- Écrire la fonction **calculerMax3**
 - elle attend 3 paramètres entiers **a**, **b** et **c**
 - elle retourne le plus grand des 3
 - elle utilisera la fonction précédente
- Écrire un programme qui demande à l'utilisateur la saisie de 3 entiers **x**, **y** et **z**, puis affiche le plus grand des 3. On utilisera la fonction **calculerMax3**.

Exercice 4

- Écrire la fonction **natureSolution** qui retourne la nature de la solution selon la valeur du ph, sachant que :

ph	[0, 2]]2, 7[7]7, 12[[12, 14]
Nature	très acide	acide	neutre	alcaline	très alcaline
- Écrire un programme qui demande à l'utilisateur de saisir le ph d'une solution (entier naturel) et affiche la nature de la solution. On utilisera la fonction précédente. Le ph d'une solution étant un nombre entre 0 et 14 (bornes incluses), il faudra vérifier la saisie.

Exemple d'exécution

```
Entrer le ph: 15
Erreur, recommencer la saisie: -1
Erreur, recommencer la saisie: 11
La solution est alcaline
```

Exercice 5

- Écrire la fonction **poidsTheorique** permettant de connaître le poids théorique (nombre réel) en fonction du sexe (H ou F) et de la taille en cm (entier naturel)
 - $poidstheorique = taille - 100 - (taille - 150)/coef$, avec **coef** valant 4 pour les hommes et 2 pour les femmes
- Écrire un programme qui demande à l'utilisateur la saisie de la taille et du sexe et qui affiche le poids théorique.
- Compléter le programme pour demander à l'utilisateur la saisie d'une taille minimale **tmin** et d'une taille maximale **tmax**, et d'un pas de calcul, puis afficher la liste des poids pour les hommes et les femmes dans cette plage de taille $[tmin, tmax]$

Exemple d'exécution

```
Taille mini (en cm) ? 150
Taille maxi (en cm) ? 200
Pas (en cm) ? 10
Taille 150cm : Hommes 50kg      Femmes 50kg
Taille 160cm : Hommes 57.5kg   Femmes 55kg
...
Taille 200cm : Hommes 87.5kg   Femmes 75kg
```

Exercice 6

- Écrire la fonction **estBissextile** à un paramètre **a** de type entier qui retourne Vrai si l'année **a** est bissextile, et Faux dans le cas contraire.
- Écrire un programme qui demande à l'utilisateur d'une année, puis affiche "année bissextile" si l'année est bissextile sinon "année non bissextile". On utilisera la fonction précédente.

Rappel : une année est bissextile si elle est divisible par 400, ou divisible par 4 mais pas par 100.

- 2000, 2008 sont des années bissextiles
- 1900, 2100 ne sont pas des années bissextiles

Exercice 7

- Écrire la fonction **calculerPuissance** à 2 paramètres **x** et **y** de type entier qui calcule et retourne x^y .
- Écrire un programme qui demande à l'utilisateur la saisie de 2 nombres **a** et **b**, avec **a** non nul et **b** positif ou nul (il faudra vérifier les saisies et redemander les valeurs tant qu'elles ne sont pas correctes), puis qui calcule et affiche la valeur de a^b . On utilisera la fonction précédente.

Exercice 8

- Écrire la fonction **calculerCarre** à un paramètre **n** de type réel qui retourne le carré de **n**.
- Écrire la fonction **CalculerSommeCarreTableau** à un paramètre **t** de type tableau de 10 réels qui retourne la somme des carrés des éléments du tableau **t**. On utilisera la fonction précédente.
- Écrire un programme qui demande la saisie des éléments d'un tableau **v** de 10 réels, puis calcule et affiche la somme des carrés des éléments de ce tableau. On utilisera la fonction précédente.

Exercice 9

1. Écrire la fonction **calculerFibonacci** à un paramètre **n** de type entier qui retourne le nième terme de la suite de Fibonacci. On supposera que **n** est positif et non nul.

Les termes de la suite de Fibonacci sont définis de la manière suivante :

- $fib_1 = 1$
- $fib_2 = 1$
- $fib_n = fib_{n-1} + fib_{n-2}$, pour $n > 2$



NE PAS TENIR COMPTE : VALEURS ERRONÉES

Exemples :

```
calculerFibonacci(7) retourne 21
calculerFibonacci(19) retourne 6765
```

Exemples (corrigés):

```
calculerFibonacci(7) retourne 13
calculerFibonacci(19) retourne 4181
```

2. Écrire un programme qui demande la saisie d'un entier **x** positif non nul (vérifier la saisie), puis affiche les **x** premiers termes de la suite de Fibonacci. On utilisera la fonction précédente.

Exemples :

```
Entrez le nombre d'éléments : -1
Erreur, recommencez la saisie : 11
1 1 2 3 5 8 13 21 34 55 89
```

Exercice 10

1. Écrire la fonction **estPair** à un paramètre **n** de type entier qui retourne Vrai si **n** est pair, Faux dans le cas contraire.
2. Écrire la fonction **calculerSyracuse** à un paramètre **n** de type entier, et dont le retour, u_{n+1} , est défini par la relation de récurrence :

$$u_{n+1} = \begin{cases} \frac{u_n}{2}, & \text{si } u_n \text{ est pair.} \\ 3u_n + 1, & \text{sinon.} \end{cases} \quad (1)$$

On supposera que **n** est positif non nul. On utilisera la fonction précédente.

3. Écrire un programme qui demande la saisie d'un entier **x** positif non nul (contrôler la saisie) , puis affiche à partir de cet entier la suite de Syracuse. Le dernier terme de la suite de Syracuse est toujours 1. On appellera la fonction précédente.

Exemples :

```
Entrez un entier positif non nul : -1
Erreur, recommencez la saisie : 15
La suite de Syracuse est :
15 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
```

Exercice 11

1. Écrire la fonction **estPremier** à un paramètre **n** de type entier qui retourne Vrai si **n** est premier, Faux dans le cas contraire.
2. Écrire un programme qui demande la saisie répétée d'un entier **x** , puis affiche un message pour indiquer que cet entier est premier ou pas. La saisie de la valeur **0** permettra d'arrêter la répétition. On utilisera la fonction précédente.

Exemples :

```
Entrez un entier positif non nul : 12
12 n'est pas premier
Entrez un entier positif non nul : 11
11 est premier
Entrez un entier positif non nul : 0
au revoir...
```