

# TP 02 – Conditionnelles et boucles

## Exercice 1 Premiers tests

### 1.1 Tests simples

- ✎ Écrivez un programme qui teste si un entier saisi par l'utilisateur est pair et compris entre 100 et 1000.

### 1.2 Tests et calculs

- ✎ Créez un programme qui demande 4 nombres à l'utilisateur puis qui affiche les informations suivantes :
  - ➡ La somme des 4 nombres
  - ➡ Le minimum parmi les 4 nombres
  - ➡ Le maximum parmi les 4 nombres
  - ➡ La moyenne des 4 nombres

### 1.3 Année bissextile

- ✎ Créez un programme capable de vérifier si une année est bissextile ou non
  - ➡ Rappel : une année est bissextile si elle est multiple de 4 sauf si elle est multiple de 100 et non de 400. Ainsi 2000, 2020, 2008 sont bissextiles, mais 1900, 2100, 2021 ne le sont pas.

### 1.4 Nombre de jours par mois

- ✎ Créez un programme qui, pour une année donnée, est capable de donner le nombre de jours pour un mois particulier.
  - ➡ Vous devez valider les entrées de l'utilisateur, c'est-à-dire vous assurer qu'elles soient cohérentes.

### 1.5 Calcul de dates

- ✎ Créez un programme qui affiche la date du lendemain d'un jour saisi par l'utilisateur.
  - ➡ Votre programme devra dans un premier temps s'assurer que la date saisie est valide.
  - ➡ Testez correctement votre programme avec au minimum les dates suivantes :
    - ➡ 31/12/2021
    - ➡ 28/02/2021
    - ➡ 28/02/2020
    - ➡ 31/07/1907

### 1.6 Arrondi de nombres réels

- ✎ Sans utiliser de fonctions mathématiques, écrivez un programme capable d'arrondir un nombre réel au centième.
  - ➡ Ex : 19.995 et 19.999 doivent devenir 20.00 mais 19.991 et 19.9949 doivent devenir 19.99.

## Exercice 2 Premières boucles

### 2.1 Une simple boucle

- Écrivez un programme qui affiche toutes les puissances de 2 comprises entre 0 et une valeur positive saisie par l'utilisateur.

### 2.2 Un triangle

- Créez un programme affichant le « triangle » suivant :

```

-----0-----
-----000-----
-----00000-----
-----0000000-----
-----000000000-----
-----00000000000-----
-----0000000000000-----
-----000000000000000-----
-----00000000000000000-----
-----0000000000000000000-----
-----000000000000000000000-----

```

Triangle dans un espace de 51 caractères de long et 10 de haut.

- Faites en sorte que la taille de l'espace soit paramétrable par l'utilisateur
- Assurez-vous que pour une taille ne permettant pas d'afficher les coins inférieurs du triangle, la zone soit correctement remplie de 'o' :

```

-----0-----
-----000-----
-----00000-----
-----0000000-----
-----000000000-----
-----00000000000-----
-----0000000000000-----
-----000000000000000-----
-----00000000000000000-----
-----0000000000000000000-----
-----000000000000000000000-----

```

Triangle dans un espace de 11 caractères de long et 11 de haut.

### 2.3 Un peu de maths

- Créez un programme permettant de calculer le quotient et le reste de la division d'un entier *naturel*  $M$  par un entier *relatif*  $N$ . Votre programme ne doit pas utiliser les opérateurs  $*$ ,  $/$  et  $\%$ .

### 2.4 La suite de Fibonacci

La suite de Fibonacci est définie de la façon suivante :

$$\begin{cases} \mathcal{F}_0 = 0 \\ \mathcal{F}_1 = 1 \\ \forall n \geq 2, \mathcal{F}_n = \mathcal{F}_{n-1} + \mathcal{F}_{n-2} \end{cases}$$


- Créez un programme qui calcule  $\mathcal{F}_n$  pour un  $n$  saisi par l'utilisateur.

### 2.5 Inverser un nombre

- Créez un programme permettant d'inverser les chiffres d'un nombre entier naturel. Par exemple, 12345 doit devenir 54321.
- Votre programme ne doit contenir qu'une seule boucle et aucune conditionnelle.


## 2.6 Nombres premiers

Un nombre entier naturel est *premier* s'il n'est divisible que par lui-même et 1.



-  Créez un programme qui affiche tous les nombres premiers inférieurs à une limite déterminée par l'utilisateur.

## 2.7 Nombre parfait

Un nombre entier naturel est *parfait* s'il est égal à la somme de tous ses diviseurs, excepté lui-même. Par exemple, 6 est parfait car  $6 = 3 + 2 + 1$ .

-  Créez un programme qui affiche tous les nombres parfaits inférieurs à une limite déterminée par l'utilisateur.

## 2.8 Gestion d'un menu d'application

-  Créez un programme qui affiche un menu permettant de choisir l'un des programmes précédents (2.1 à 2.7).
  -  Une fois le résultat affiché, le menu doit être réaffiché jusqu'à ce que l'utilisateur choisisse l'élément du menu permettant de quitter l'application.