

Les structures de base

Séquence, sélection, itération

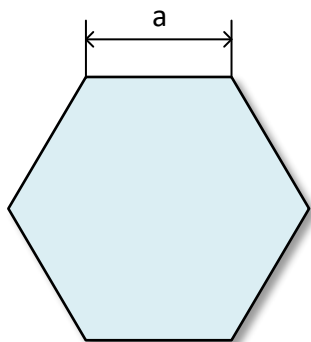
1 But

Le but de ces exercices est d'entraîner l'utilisation des structures de base, c'est-à-dire la séquence d'instructions, la sélection et l'itération.

Pour chaque exercice, un effort particulier devra être fait afin de produire un code parfaitement lisible : les identifiants seront bien nommés, le code sera judicieusement commenté et correctement indenté. Il ne sera pas dupliqué inutilement.

2 Séquence d'instructions

2.1 La surface d'un hexagone régulier



Écrivez un programme qui demande à l'utilisateur de saisir la longueur de l'arête d'un hexagone régulier, puis calcule et affiche la surface de ce polygone.

La surface S se calcule de la façon suivante :

$$S = \frac{3 \cdot \sqrt{3}}{2} \cdot a^2$$

Pour des raisons de simplification, les valeurs saisies par l'utilisateur ne seront pas vérifiées.

2.2 Unités et dizaines

Ecrivez un programme qui demande à l'utilisateur de saisir un nombre à deux chiffres et qui affiche ensuite le message suivant : Le nombre n contient x unités et y dizaines. où n est le nombre saisi par l'utilisateur (par exemple 37), x le nombre d'unités (par exemple 7) et y le nombre de dizaines (par exemple 3).

Pour des raisons de simplification, les valeurs saisies par l'utilisateur ne seront pas vérifiées.

3 Sélections

3.1 Porte-bonheur ?

Écrivez un programme qui demande à l'utilisateur de saisir un nombre entier. Si le nombre saisi est 13, le programme affiche la phrase « Vous avez saisi le nombre 13. Par superstition ? », sinon, il affiche la phrase « Vous avez saisi le nombre n . Merci ! », où n est remplacé par le nombre saisi.

Voici quelques exemples de fonctionnement :

```
Saisir un nombre entier : 28
Vous avez saisi le nombre 28. Merci !
```

```
Saisir un nombre entier : 13
Vous avez saisi le nombre 13. Par superstition ?
```

Avec quelles valeurs allez-vous tester votre programme pour garantir qu'il fonctionne correctement ?

3.2 Proches ou lointains

Écrivez un programme qui demande à l'utilisateur de saisir deux nombres entiers. Si la différence entre eux est inférieure à 10, le programme affiche *Proches*. Sinon, il affiche *Lointains*.

Voici quelques exemples de fonctionnement :

```
Saisir le 1er nombre : 7
Saisir le 2e nombre : 2
Proches
```

```
Saisir le 1er nombre : 2
Saisir le 2e nombre : 14
Lointains
```

Avec quelles valeurs allez-vous tester votre programme pour garantir qu'il fonctionne correctement ?

3.3 Le plus grand

Écrivez un programme qui demande à l'utilisateur de saisir trois nombres entiers et qui indique le plus grand.

Exemples :

```
Indiquez un premier nombre : 7
Indiquez un deuxieme nombre : 18
Indiquez un troisieme nombre : -5
Le plus grand nombre est 18.
```

```
Indiquez un premier nombre : 1234
Indiquez un deuxieme nombre : 555
Indiquez un troisieme nombre : 1234
Le plus grand nombre est 1234.
```

3.4 Les opérations de base

Écrivez un programme qui demande à l'utilisateur de saisir deux nombres entiers (les opérandes 1 et 2), puis qui affiche le résultat des 4 opérations arithmétiques de base : addition, soustraction, multiplication, division (décimale).

Voici un exemple de fonctionnement :

```
Saisir l'opérande 1 : 32
Saisir l'opérande 2 : 7

32 + 7 = 39
32 - 7 = 25
32 * 7 = 224
32 / 7 = 4.571
```

Remarques :

Le résultat de la division décimale doit s'afficher avec trois chiffres après la virgule.
 Dans le cas d'une division par zéro, le programme affiche Erreur comme résultat.

Voici un autre exemple de fonctionnement :

```
Saisir l'opérande 1 : 32
Saisir l'opérande 2 : 0

32 + 0 = 32
32 - 0 = 32
32 * 0 = 0
32 / 0 = Erreur
```

3.5 Prix d'entrée au musée

Écrivez un programme qui calcule le prix d'entrée au musée selon l'âge du visiteur.

Il faut que l'utilisateur saisisse l'âge du visiteur, puis le prix d'entrée est calculé et affiché.

Le prix est déterminé selon la méthode suivante :

1. **L'âge** du visiteur détermine une **catégorie de prix** selon le tableau suivant :

Âge	Catégorie d'âge	Catégorie de prix
Moins de 4 ans	Bébé	0
De 4 ans à 15 ans (non-compris)	Enfant	1
De 15 ans à 65 ans (non-compris)	Adulte	2
65 ans ou plus	Senior	1

2. À partir de la **catégorie**, le prix d'entrée est **calculé** avec la formule suivante :

$$\text{Prix d'entrée} = \text{Catégorie} \cdot 12 + 5$$

Exemples :

```
Saisir l'âge du visiteur : 8
Prix d'entrée : 17 CHF
```

EXERCICES		INF1 / SIN1	 CEJEF DIVISION TECHNIQUE ÉCOLE SUPÉRIEURE TECHNIQUE
M2 / Programmation / Les structures de base			
EST – M2 – Prog – ED 01	JCO	Mise à jour : 16.01.2019	

Saisir l'âge du visiteur : 45
Prix d'entrée : 29 CHF

Saisir l'âge du visiteur : 75
Prix d'entrée : 17 CHF

Saisir l'âge du visiteur : 3
Prix d'entrée : 5 CHF

3.6 Les super-héros

Écrivez un programme qui demande à l'utilisateur de saisir le nom d'un super-héros.

Si le nom saisi est *Superman*, écrire « De son vrai nom Kal-El, il a été créé en 1933. »
 Si le nom saisi est *Batman*, écrire « Il vit à Gotham et n'a aucun pouvoir surhumain. »
 Si le nom saisi est *Supergirl*, écrire « C'est la cousine de Superman ! »
 Si le nom saisi est *Superdupont*, écrire « Avec son béret et ses charentaises, il est ultra-patriote. »
 Si le nom saisi est *Leguman*, écrire « Il lutte contre les objets qui deviennent fous ! »
 Pour tout autre nom, écrire « Je ne connais pas ce super-héros ! »

Exemples :

Super-heros : **Superman**
De son vrai nom Kal-El, il a été créé en 1933.

Super-heros : **Leguman**
Il lutte contre les objets qui deviennent fous !

Super-heros : **Granitman**
Je ne connais pas ce super-heros !

Super-heros : **Supergirl**
C'est la cousine de Superman !

3.7 En toutes lettres

Écrivez un programme qui demande à l'utilisateur de saisir un **chiffre** et qui l'affiche en toutes lettres.

Si le nombre saisi est inférieur à zéro, il faut afficher *Trop petit*. S'il est égal ou supérieur à 10, il faut afficher *Trop grand*.

Exemples :

Saisir le chiffre à écrire en toutes lettres : 8
Huit

Saisir le chiffre à écrire en toutes lettres : -5
Trop petit

EXERCICES		INF1 / SIN1	 CEJEF DIVISION TECHNIQUE ÉCOLE SUPÉRIEURE TECHNIQUE
M2 / Programmation / Les structures de base			
EST – M2 – Prog – ED 01	JCO	Mise à jour : 16.01.2019	

Saisir le chiffre à écrire en toutes lettres : **15**
 Trop grand

EXERCICES		INF1 / SIN1	 CEJEF DIVISION TECHNIQUE ÉCOLE SUPÉRIEURE TECHNIQUE
M2 / Programmation / Les structures de base			
EST – M2 – Prog – ED 01	JCO	Mise à jour : 16.01.2019	

3.8 Le trésor des pirates¹

Vous êtes le capitaine d'un bateau de pirates qui doit déterminer l'état de la cargaison.

L'utilisateur doit saisir deux informations (entières) : la **quantité de pièces d'or** et le **nombre de pirates** à bord du bateau.

À partir de ces deux informations, le programme doit indiquer si la cargaison est **mauvaise**, **acceptable** ou **excellente** :

- Si le nombre de pièces d'or est inférieur au nombre de pirates, la cargaison est mauvaise ;
- Si le nombre de pièces d'or additionné au nombre de pirates dépasse 100, la cargaison est mauvaise, car elle risque de faire couler le bateau ;
- Si le nombre de pièces d'or est au moins égal au nombre de pirate, la cargaison est acceptable. Si elle se divise équitablement entre chaque pirate alors elle est excellente !

Pour des raisons de simplification, on considère que l'utilisateur ne saisira pas de valeurs illégales (négatives).

On considère également que le nombre de pirates ne pourra pas être inférieur à 1.

Une fois la qualité de la cargaison affichée (en blanc sur fond bleu), la citation suivante est inscrite : *Tous les trésors ne sont pas d'argent et d'or...*

Exemples :

```
Saisir le nombre de pièces d'or : 75
Saisir le nombre de pirates à bord : 15

La cargaison est excellente.
Tous les trésors ne sont pas d'argent et d'or...
```

```
Saisir le nombre de pièces d'or : 90
Saisir le nombre de pirates à bord : 15

La cargaison est mauvaise.
Tous les trésors ne sont pas d'argent et d'or...
```

```
Saisir le nombre de pièces d'or : 50
Saisir le nombre de pirates à bord : 14

La cargaison est acceptable.
Tous les trésors ne sont pas d'argent et d'or...
```

```
Saisir le nombre de pièces d'or : 10
Saisir le nombre de pirates à bord : 15

La cargaison est mauvaise.
Tous les trésors ne sont pas d'argent et d'or...
```

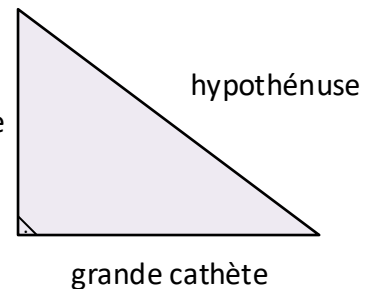
¹ Inspiré de l'exercice *Pirate Ship II* du site *Learneroo*.

3.9 Triangle quelconque ou rectangle

Le but est de déterminer, à partir de trois longueurs, si elles correspondent à un triangle quelconque ou rectangle.

Écrivez un programme qui demande à l'utilisateur de saisir la longueur des trois côtés d'un triangle.

Si au moins un de ces trois côtés est inférieur ou égal à zéro, le message Dimensions invalides s'affiche et le programme se termine.



Si les trois côtés sont valides, le programme doit déterminer lequel est l'hypoténuse, afin de pouvoir dire si les trois dimensions correspondent bel et bien à un triangle (la somme des longueurs des cathètes doit être supérieure ou égale à celle de l'hypoténuse). Si c'est le cas, il s'agit ensuite de déterminer si c'est un triangle rectangle ($hypothénuse = \sqrt{cathèteA^2 + cathèteB^2}$) ou quelconque.

Si les trois longueurs ne correspondent pas à un triangle, afficher le message Ce n'est pas un triangle. Si c'est un triangle rectangle, afficher le message C'est un triangle rectangle. Sinon, afficher le message C'est un triangle quelconque.

Exemples :

```
Saisir la longueur du premier côté : 6
Saisir la longueur du deuxième côté : 10
Saisir la longueur du troisième côté : 8
C'est un triangle rectangle.
```

```
Saisir la longueur du premier côté : 10
Saisir la longueur du deuxième côté : 5
Saisir la longueur du troisième côté : 7
C'est un triangle quelconque.
```

```
Saisir la longueur du premier côté : 3
Saisir la longueur du deuxième côté : 4
Saisir la longueur du troisième côté : 5
C'est un triangle rectangle.
```

```
Saisir la longueur du premier côté : 3
Saisir la longueur du deuxième côté : 7
Saisir la longueur du troisième côté : 15
Ce n'est pas un triangle.
```

```
Saisir la longueur du premier côté : 0
Saisir la longueur du deuxième côté : 6
Saisir la longueur du troisième côté : 4
Dimensions invalides.
```

```
Saisir la longueur du premier côté : 5
Saisir la longueur du deuxième côté : 7
Saisir la longueur du troisième côté : -3
Dimensions invalides.
```

4 Itérations

4.1 De pair à impair

Écrivez un programme qui demande à l'utilisateur de saisir un nombre entier **strictement positif**. La saisie est **répétée** tant que le nombre saisi n'est pas strictement positif.

Ensuite, **diviser ce nombre par deux** autant de fois que nécessaire jusqu'à l'obtention **d'un nombre impair**, en affichant le résultat obtenu après chaque division.

Une fois le nombre impair atteint, le programme se termine en affichant « Nombre impair atteint ! »

Exemples :

```
Saisir un nombre entier strictement positif : 88
44
22
11
Nombre impair atteint !
```

```
Saisir un nombre entier strictement positif : 53
Nombre impair atteint !
```

```
Saisir un nombre entier strictement positif : 0
Saisir un nombre entier strictement positif : 128
64
32
16
8
4
2
1
Nombre impair atteint !
```

4.2 Une centaine de nombres

Écrivez un programme qui affiche tous les nombres entiers compris entre 1 et 100, séparés par une tabulation.

4.3 Les nombres pairs

Écrivez un programme qui affiche tous les nombres pairs compris entre 1 et 100, séparés par une tabulation.

4.4 Liste d'entiers

Écrivez un programme qui demande à l'utilisateur de saisir un nombre entier positif et qui affiche ensuite tous les nombres entiers compris entre 1 et le nombre saisi.

Si l'utilisateur saisi un nombre inférieur à 1, la saisie est répétée.

Les nombres affichés sont séparés par une tabulation.

4.5 Pair ou impair

Écrivez un programme qui demande à l'utilisateur de saisir un nombre compris entre 1 et 100 et qui indique si ce nombre est pair ou impair.

Si l'utilisateur saisit un nombre qui n'est pas compris dans l'intervalle requis, le programme répète la saisie (autant de fois que nécessaire).

4.6 Pair ou impair II

Écrivez un programme qui demande à l'utilisateur de saisir un nombre compris entre 1 et 100 et qui indique si ce nombre est pair ou impair.

Si l'utilisateur saisit un nombre qui n'est pas compris dans l'intervalle requis, le programme **affiche un message d'erreur** et répète la saisie (autant de fois que nécessaire).

4.7 Les n premiers impairs

Écrivez un programme qui demande à l'utilisateur de saisir un nombre positif (n) et qui affiche les n premiers nombres impairs ainsi que leur somme.

Si l'utilisateur saisit un nombre invalide, le programme affiche un message d'erreur et répète la saisie (autant de fois que nécessaire).

Voici un exemple du résultat à obtenir :

```
Entrez un nombre entier positif : -10
Entrez un nombre entier positif : 5
1      3      5      7      9
Somme : 25
```

4.8 Le royaume des Sansète

Dans le royaume des Sansète, le chiffre 7 porte malheur. On tente par tous les moyens de l'éviter, lui et ses multiples.

Écrivez un programme qui affiche tous les nombres compris entre 1 et 100, sauf 7 et ses multiples.

4.9 Le royaume des Sansète II

Désormais, le royaume des Sansète a décidé de bannir complètement le chiffre 7.

Écrivez un programme qui affiche tous les nombres compris entre 1 et 100, sauf 7 et ses multiples, ni les nombres qui contiennent le chiffre 7 (par exemple 27).

4.10 Le test du Fizz Buzz

Écrivez un programme qui affiche tous les nombres de 1 à 100. Toutefois, pour les multiples de 3, écrire « Fizz » à la place du nombre, pour les multiples de 5, écrire « Buzz » à la place du nombre et pour les multiples de 3 et de 5, écrire « FizzBuzz ».²

² Une vidéo de Tom Scott est disponible sur ce sujet, avec sous-titres en français : https://www.youtube.com/watch?v=QPZ0pIK_wsc

EXERCICES		INF1 / SIN1	 CEJEF DIVISION TECHNIQUE ÉCOLE SUPÉRIEURE TECHNIQUE
M2 / Programmation / Les structures de base			
EST – M2 – Prog – ED 01	JCO	Mise à jour : 16.01.2019	

4.11 PPMC

Écrivez un programme qui demande à l'utilisateur de saisir deux nombre entiers strictement positifs et qui retourne le PPMC de ces deux nombres.

Pour des raisons de simplification, on admet que l'utilisateur ne saisira pas de nombres invalides (pouvant provoquer une division par zéro).

Exemples :

```
Saisir le premier nombre : 3
Saisir le deuxième nombre : 5
Le PPMC de 3 et 5 est 15.
```

```
Saisir le premier nombre : 10
Saisir le deuxième nombre : 4
Le PPMC de 4 et 10 est 20.
```

4.12 Les chiffres d'un nombre

Écrivez un programme qui demande à l'utilisateur de saisir un nombre entier et qui affiche de nombre, chiffre par chiffre en **commençant** par les unités.

Remarque : pour ceux qui seraient tentés de passer par des chaînes de caractères, ce n'est pas nécessaire !

Voici un exemple d'exécution :

```
Saisir un nombre : 712253
3
5
2
2
1
7
```

Pour les plus avancés et motivés :

Modifiez le programme pour qu'il affiche le nombre, chiffre par chiffre, en **finissant** par les unités.

Voici un exemple d'exécution :

```
Saisir un nombre : 712253
7
1
2
2
5
3
```