

Chapitre 1 : Exercices

Ex 1. Créer un programme permettant de calculer la consommation d'essence en fonction du nombre de litres ajoutés dans le réservoir et du nombre de kilomètres parcourus depuis le dernier plein.

Instructions de base

Algorithme

Variables à déclarer

Nom	Type	Valeur initiale	Description
<i>nbKms</i>	Réel		Nombre de kilomètres parcourus
<i>nbLitres</i>	Réel		Nombre de litres ajoutés dans le réservoir
<i>conso</i>	Réel		Consommation pour 100 kms

Début

Saisir le nombre de kilomètres ***nbKms*** parcourus
Saisir le nombre de litres d'essence ***nbLitres*** ajoutés dans le réservoir
conso \leftarrow ***nbLitres*** * 100 / ***nbKms***
Afficher ***conso***

Fin

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

Ex 2. Ecrire un programme qui calcule et affiche l'aire d'un triangle dont il faut entrer les longueurs des trois côtés. Utilisez la formule : $S^2 = P(P-A)(P-B)(P-C)$ où A, B, C sont les longueurs des trois côtés et P le demi-périmètre du triangle.

- Instructions de base
- Instructions conditionnelles

Algorithme

Variables à déclarer

Nom	Type	Valeur initiale	Description
<i>a, b, c</i>	Réel		Dimensions des 3 côtés du triangle
<i>p</i>	Réel		½ Périmètre
<i>tmp</i>	Réel		Variable temporaire
<i>aire</i>	Réel		Surface du triangle

Début

Saisir *a*, *b* et *c* au clavier

$p \leftarrow (a+b+c)/2$

$tmp \leftarrow p(p-a)(p-b)(p-c)$

Si *tmp* < 0 **alors**

Afficher "Erreur de saisie"

sinon

$aire \leftarrow \sqrt{tmp}$

Afficher *aire*

Finsi

Fin

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S
#include <math.h> // pour la racine carrée

using namespace std; // Espace de nommage standard

int main( )
{
    ...

    aire=sqrt(tmp) ; // aire ← √tmp

    ...

    cin.get();cin.ignore();
    return 0;
}
```

Ex 3. Ecrire un programme qui affiche la résistance équivalente à trois résistances R1, R2, R3. On demandera à l'opérateur s'il veut les brancher en série ou en parallèle.

- si les résistances sont branchées en série:

$$R_{ser} = R1 + R2 + R3$$

- si les résistances sont branchées en parallèle :

$$R_{par} = \frac{R1 \cdot R2 \cdot R3}{R1 \cdot R2 + R1 \cdot R3 + R2 \cdot R3}$$

- Instructions de base
- Instructions conditionnelles

Algorithme

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

Ex 4. Ecrire un programme C++ qui indique si une année saisie au clavier est bissextile ou non.

Une année est bissextile si :

- Elle est divisible par 4 et **pas** par 100
- Ou si elle est divisible par 400

Exemple :

2012 était bissextile car 2012 est divisible par 4 (reste de la division par 4 est nul) et pas divisible par 100

2000 était bissextile car elle était divisible par 400

1900 n'était pas bissextile (pas divisible par 400)

- Instructions de base
- Instructions conditionnelles

Algorithme**Programme C++**

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

En C++

&& : et logique

|| : ou logique

!= : différent

% : reste de la division entière

Ex 5. Ecrire un programme C++ qui calcule les deux chiffres de la clé d'un numéro de sécurité sociale

Le numéro de sécurité sociale correspond au numéro d'inscription au répertoire national d'identification des personnes physiques (RNIPP). Il est formé du numéro d'inscription (NIR) à 13 chiffres et d'une clé de contrôle à 2 chiffres.

- Instructions de base
- Instructions conditionnelles

La clé de contrôle peut être calculée de la manière suivante :

clé = 97 – (NIR modulo 97)

Exemple :

NIR = **1 54 02 50 025 005**

On calcule le reste de la division de NIR par 97 (le modulo) ; on obtient 16

La clé est donc 97-16 = 81

Algorithme**Programme C++**

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

Ex 6. Vérification d'une carte bancaire

- Instructions de base
- Instructions conditionnelles

Le code d'une carte bancaire contient 16 chiffres que nous noterons par les variables ***a, b, c, ,d ... p***

abcd efgh ijkl mnop

Banque Propriétaire de la carte Clé

Type de carte :
 3 : American Express
 4 : Visa
 5 : Eurocard/Mastercard

La clé ***p*** appelée clé de Luhn peut être trouvée à partir des chiffres ***a,b,c,d,...o*** de la manière suivante :

- On multiplie ***a*** par 2. Si le résultat est supérieur à 9, on soustrait 9. On remplace ***a*** par ce résultat. Faire de même avec ***c, e, g, i, k, m*** et ***o***
- Mettre ***a+b+...+m+n+o*** dans ***p***
- Remplacer ***p*** par son reste de division entière par 10
- Remplacer ***p*** par ***10-p*** si ***p*** est non nul

Ecrire l'algorithme puis le programme qui vérifie si la clé du code de carte bleue saisi au clavier (chiffres ***a*** à ***p***) est valide ou non.

Pour cela, ajouter une variable ***pCalcule*** que vous calculerez à partir des chiffres ***a*** à ***o***, puis vous comparerez ***pCalcule*** avec le chiffre ***p*** fourni.

Algorithme	Programme C++
	<pre>#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S using namespace std; // Espace de nommage standard int main() {</pre> <div data-bbox="1668 746 2166 1074"><p>En C++</p><p>&& : et logique</p><p> : ou logique</p><p>!= : différent</p><p>% : reste de la division entière</p></div>

Ex 7. Écrire un programme qui additionne tous les nombres impairs inférieurs à 100, et qui affiche le résultat obtenu.

- Instructions de base
- Boucle

Algorithme

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S  
  
using namespace std; // Espace de nommage standard  
  
int main( )  
{
```


Ex 8. Calculez la somme des n premiers termes de la série harmonique : $1 + 1/2 + 1/3 + \dots + 1/n$ (n est saisi au clavier).

- Instructions de base
- Boucle

Algorithme

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

Ex 9. Créer un programme permettant d'afficher une approximation du nombre d'or

- Instructions de base
- Boucle

Le nombre d'or est une valeur qui inspira et qui inspire toujours les architectes, les artistes, les poètes etc....
Je vous propose d'en trouver une valeur approchée. Ce nombre peut s'écrire sous la forme :

$$\Phi = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \dots}}}}$$

Eh oui, il s'agit bien d'un ensemble de fractions !!! Ne vous effrayez pas. Ecrivons les valeurs suivantes :

$$v_1 = 1 \quad v_2 = 1 + \frac{1}{1} \quad v_3 = 1 + \frac{1}{1 + \frac{1}{1}} \quad v_4 = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}}$$

Trouvez le lien entre v_2 et v_1 puis entre v_3 et v_2 puis entre v_4 et v_3 etc...Ecrivez le programme C permettant de calculer v_{100} qui constituera une bonne approximation du nombre d'or.

Ce nombre peut aussi s'écrire :

$$\Phi = \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \dots}}}}}$$

En utilisant la fonction sqrt (inclure <math.h>), trouvez une approximation du nombre d'or en utilisant cette formule.

Algorithme	Programme C++
	<pre>#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S using namespace std; // Espace de nommage standard int main() {</pre>

Ex 10. Créer un programme permettant d'afficher le maxi, le mini et la moyenne d'un ensemble de valeurs saisies. Après chaque valeur saisie, vous demanderez à l'opérateur s'il veut continuer.

- Instructions de base
- Instructions conditionnelles
- Boucle

Algorithme

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S  
  
using namespace std; // Espace de nommage standard  
  
int main( )  
{
```

Ex 11. On lâche une balle d'une hauteur h . Elle rebondit sur place et chaque rebond a pour hauteur les $7/8$ ^{ème} du rebond de la hauteur précédente. Elle s'arrête lorsque la hauteur devient inférieure à 0,5 mm. Ecrire un programme qui lit au clavier la hauteur initiale en mm et qui calcule la distance parcourue par la balle.

- Instructions de base
- Instructions conditionnelles
- Boucle

Algorithme

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

Ex 12. Afficher le nombre binaire associé au nombre entier saisi au clavier (nombre positif sur 2 octets)

Exemple: Exemple : on saisit 135 → Le programme affiche 1000 0111.

Astuce :

En partant du bit de gauche (MSB), faire un **masque**¹ pour identifier si ce bit est à 1 ou à 0. Le masque correspond en fait au poids. Commencer à comparer le nombre avec le poids le plus grand puis diviser le poids par 2 (ou faire un décalage du masque) pour continuer.

- Instructions de base
- Instructions conditionnelles
- Opérations bits à bits (&, |, ...)
- Boucle

Algorithme**Programme C++**

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

¹ Et bit à bit (&) , Ou bit à bit (|), ...

Ex 13. Calcul d'un nombre élevé à une certaine puissance.

Saisir au clavier un réel x puis un entier n et calculer x^n en remarquant que

- $x^n = x.x.x.x.x \dots$ (n fois) si $n > 0$
- $x^0 = 1$
- $x^n = \frac{1}{x.x.x.x.x \dots (|n| \text{ fois})}$ si $n < 0$
- Par exemple : $7^3 = 7.7.7 = 343$
 $7^{-3} = \frac{1}{7.7.7} = 0,0029$

- Instructions de base
- Instructions conditionnelles
- Boucle

Algorithme**Programme C++**

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S

using namespace std; // Espace de nommage standard

int main( )
{
```

Ex 14. Ecrire un programme qui demande à l'utilisateur de saisir un entier N et qui affiche N étoiles sur la première ligne, (N-2) sur la seconde, le tout centré

si N=5, on affiche

*

- Instructions de base
- Boucles imbriquées

Algorithme

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S
```

```
using namespace std; // Espace de nommage standard
```

```
int main()  
{
```


Ex 15. Ecrire un programme qui demande à l'utilisateur un entier n et affiche un cadre de dimension n * n entouré par des étoiles.

```
*****  
*           *  
*           *  
*           *  
*           *  
*           *  
*           *  
*****
```

Exemple avec n=8

- Instructions de base
- Boucles imbriquées

Algorithme

Programme C++

```
#include <iostream> // Fichier à inclure pour utiliser les flots d'E/S  
  
using namespace std; // Espace de nommage standard  
  
int main( )  
{
```