TP1 : NOTIONS DE BASE DU LANGAGE C++

Exercice 1 :

Écrire un programme qui permet d'afficher une table de correspondance entre les températures en Celsius et en Fahrenheit en sachant que : F=C*(9/5)+32. Commencer à 0° et procéder jusqu'à 100° avec un pas de 20°.

Exercice 2 :

Ecrire un programme C++ permettant d'afficher:

1. A^B,
2. l'hypoténuse d'un triangle rectangle de côtés A et B,
3. la tangente de A en n'utilisant que les fonctions **sin** et **cos**,
4. la valeur arrondie (en moins) de A/B puis à trois positions derrière la virgule de A/B.

Exercice 3 :

Ecrire un programme en C++ qui demande à l'utilisateur de taper 20 entiers et qui affiche le plus petit de ces entiers.

Exercice 4 :

Un nombre entier est parfait s'il est égal à la somme de ses diviseurs (sauf lui même).

Ex : 6 = 1 + 2 + 3 est parfait.

- Ecrire un programme en C++ qui teste si un nombre saisi est parfait ou non.

Exercice 5 :

Ecrire un programme en C++ qui demande à l'utilisateur de taper un entier N et qui calcule u(N) défini par :

$$\begin{cases} u(0)=3 \\ u(n+1)=3.u(n)+4 \end{cases}$$

Exercice 6 :

Ecrire un programme en C++ qui demande à l'utilisateur de taper un entier N et qui calcule u(N) défini par :

$$\begin{cases} u(0)=1 \\ u(1)=1 \\ u(n+1)=u(n)+u(n-1) \end{cases}$$

Exercice 7 :

La suite de Syracuse repose sur un principe simple. Prenez un nombre entier positif quelconque :

- S'il est pair, divisez-le par 2;
- S'il est impair, multipliez-le par 3 et ajoutez 1.

Renouvelez cette opération plusieurs fois. Après suffisamment d'itérations, vous devriez finir par tomber sur la valeur 1.

- Ecrire un programme en C++ qui demande à l'utilisateur d'entrer un nombre entier n ($1 \leq n \leq 50$) et qui affiche à l'écran les valeurs successives de la suite de Syracuse relative à ce nombre (en s'arrêtant bien sûr à 1), ainsi que le nombre d'itérations qui ont été nécessaires pour parvenir à 1.

Par exemple, à partir du nombre 5, on trouve la suite de valeurs: 16 8 4 2 1 et le nombre d'itérations vaut 5.

Exercice 8 :

Ecrire un programme C++ qui calcule une valeur approchée de sin(x) à un rang k entré au clavier, et pour une valeur x exprimée en radians. On utilisera le développement suivant:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^k \times \frac{x^{2k+1}}{(2k+1)!}$$

Exercice 9 :

Lorsqu'une balle tombe d'une hauteur initiale h_0, sa vitesse d'arrivée au sol est $v_0 = \sqrt{2gh_0}$

Immédiatement après le rebond, sa vitesse est $v_1 = \varepsilon . v_0$ (où ε est le coefficient de rebond de la balle). Elle remonte alors à la hauteur $h = v_1^2 / 2g$.

Le but est d'écrire un programme en C++ qui calcule et affiche la hauteur à laquelle la balle remonte après un nombre donné de rebonds.

Le programme devra utiliser la constante g (de valeur 9.81) et demander à l'utilisateur d'entrer, en assurant le respect des contraintes associées, les valeurs de:

- ε, le coefficient de rebond, t.q. $0 \leq \varepsilon < 1$;
- h_0, la hauteur initiale, $h_0 > 0$;
- n, le nombre de rebours, t.q. $n \geq 0$.

En testant le programme avec les valeurs $\varepsilon = 0.9$, $h_0 = 25$ et $n = 10$, la hauteur obtenue devrait être environ 3.04.

Exercice 10 :

On cherche maintenant le nombre de rebonds que fait cette balle avant que la hauteur à laquelle elle rebondit ne soit plus petite ou égale à une hauteur donnée h_{fin}.

Ecrire un programme en C++ qui calcule au moyen d'une boucle « do…while » le nombre de rebonds.

Le programme doit demander à l'utilisateur d'entrer les valeurs de:

- ε, le coefficient de rebond, t.q. $0 \leq \varepsilon < 1$;
- h_0, la hauteur initiale, $h_0 > 0$;
- h_{fin}, la hauteur finale désirée, t.q. $0 < h_{fin} < h_0$.

En testant le programme avec les valeurs $\varepsilon=0.9$, $h_0=10$ et $h_{fin}=2$, on devrait obtenir 8 rebonds.

Exercice 11 (Affichage d'un triangle isocèle d'étoiles.)

Ecrire un programme C++ qui, étant donne un entier naturel impair base, affiche un triangle isocèle d'étoiles, ayant pour base, base étoiles. La valeur de la base sera saisie par l'utilisateur et on considérera qu'il saisit bien un nombre impair.

Trois exemples d'exécution sont les suivants :

Nombre d'étoiles a la base du triangle (impair) ? 5

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

Nombre d'étoiles a la base du triangle (impair) ?3

```
  *
 ***
```

Nombre d'étoiles a la base du triangle (impair) ?1

```
*
```