

MOOC Init. Prog. C++

Exercices semaine 3

Exercice 7 : tables de multiplication (itération `for`, niveau 1)

Cet exercice correspond à l'exercice n°3 (pages 19 et 199) de l'ouvrage [C++ par la pratique \(3^e édition, PPUR\)](#).

Objectif

Écrivez un programme `tables.cc` affichant les tables de multiplication de 2 à 10.

Votre programme devra produire la sortie suivante à l'écran :

Tables de multiplication

Table de 2 :

1 * 2 = 2

...

10 * 2 = 20

...

Table de 5 :

1 * 5 = 5

2 * 5 = 10

...

...

Table de 10 :

1 * 10 = 10

...

Méthode :

Utilisez deux structures d'itération `for` imbriquées l'une dans l'autre.

Exercice 8 : rebonds de balles (itération `for`, niveau 2)

Cet exercice correspond à l'exercice n°6 (pages 21 et 202) de l'ouvrage [C++ par la pratique \(3^e édition, PPUR\)](#).

Objectif :

L'objectif de cet exercice est de résoudre le problème suivant :

Lorsqu'une balle tombe d'une hauteur initiale h , sa vitesse à l'arrivée au sol est

$v = \sqrt{2 \times h \times g}$. Immédiatement après le rebond, sa vitesse est $v1 = eps \times v$ (où eps est une constante et v la vitesse avant le rebond). Elle remonte alors à la hauteur $h = \frac{v1^2}{2g}$.

Le but est d'écrire un programme (`rebonds1.cc`) qui calcule la hauteur à laquelle la balle remonte après un nombre `nbr` de rebonds.

Méthode :

On veut résoudre ce problème, non pas du point de vue formel (équations) mais par **simulation** du système physique (la balle).

Utilisez une itération `for` et des variables `v`, `v1`, (les vitesses avant et après le rebond), et `h`, `h1` (les hauteurs au début de la chute et à la fin de la remontée).

Tâches :

Écrivez le programme `rebonds1.cc` qui affiche la hauteur après le nombre de rebonds spécifié.

Votre programme devra utiliser la **constante** `g`, de valeur 9,81 et demander à l'utilisateur d'entrer les valeurs de

- **H0** (hauteur initiale, contrainte : $H0 > 0$),
- **eps** (coefficient de rebond, contrainte $0 \leq eps < 1$)
- **nbr** (nombre de rebonds, contrainte : $0 \leq NBR$).

Essayez les valeurs $H0 = 25$, $eps = 0.9$, $NBR = 10$. La hauteur obtenue devrait être environ 3.04.

Remarque :

- Pour utiliser les fonctions mathématiques (comme `sqrt()`), ajoutez `#include <cmath>` au début de votre fichier source.

Exercice 9 : Rebonds de balles - le retour. (boucles do...while, niveau 2)

Cet exercice correspond à l'exercice n°7 (pages 22 et 203) de l'ouvrage [C++ par la pratique \(3^e édition, PPUR\)](#).

On se demande maintenant combien de rebonds fait cette balle avant que la hauteur à laquelle elle rebondit soit plus petite que (ou égale à) une hauteur donnée `h_fin`.

Écrivez le programme `rebonds2.cc` qui affiche le nombre de rebonds à l'écran.

Il devra utiliser une boucle `do...while`, et demander à l'utilisateur d'entrer les valeurs de :

- **H0** (hauteur initiale, contrainte : $H0 > 0$),
- **eps** (coefficient de rebond, contrainte $0 \leq \text{eps} < 1$)
- **h_fin** (hauteur finale désirée, contrainte : $0 < \text{h_fin} < H0$).

Essayez les valeurs `H0=10`, `eps=0.9`, `h_fin=2`. Vous devriez obtenir 8 rebonds.

Exercice 10: Nombres premiers (structures de contrôle, niveau 2)

Cet exercice correspond à l'exercice n°9 (pages 22 et 205) de l'ouvrage [C++ par la pratique \(3^e édition, PPUR\)](#).

Écrivez le programme `premier.cc` qui demande à l'utilisateur d'entrer un entier n strictement plus grand que 1, puis décide si ce nombre est premier ou non.

Algorithme :

1. Vérifier si le nombre n est pair (si oui, il n'est pas premier sauf si c'est 2).
2. Pour tous les nombres impairs inférieurs ou égaux à la racine carrée de n , vérifier s'ils divisent n . Si ce n'est pas le cas, alors n est premier.

Tâches :

- Si n n'est pas premier, votre programme devra afficher le message: "Le nombre n'est pas premier, car il est divisible par D ", où D est un diviseur de n autre que 1 et n .
- Sinon, il devra afficher le message: "Je crois fortement que ce nombre est premier".

Testez votre programme avec les nombres : 2, 16, 17, 91, 589, 1001, 1009, 1299827 et 2146654199. Indiquez ceux qui sont premiers.

Les résultats devraient ressembler à ceci :

```
2 est premier
16 n'est pas premier, car il est divisible par 2
17 est premier
91 n'est pas premier, car il est divisible par 7
589 n'est pas premier, car il est divisible par 19
1001 n'est pas premier, car il est divisible par 7
1009 est premier
1299827 est premier
2146654199 n'est pas premier, car il est divisible par 46327
```
