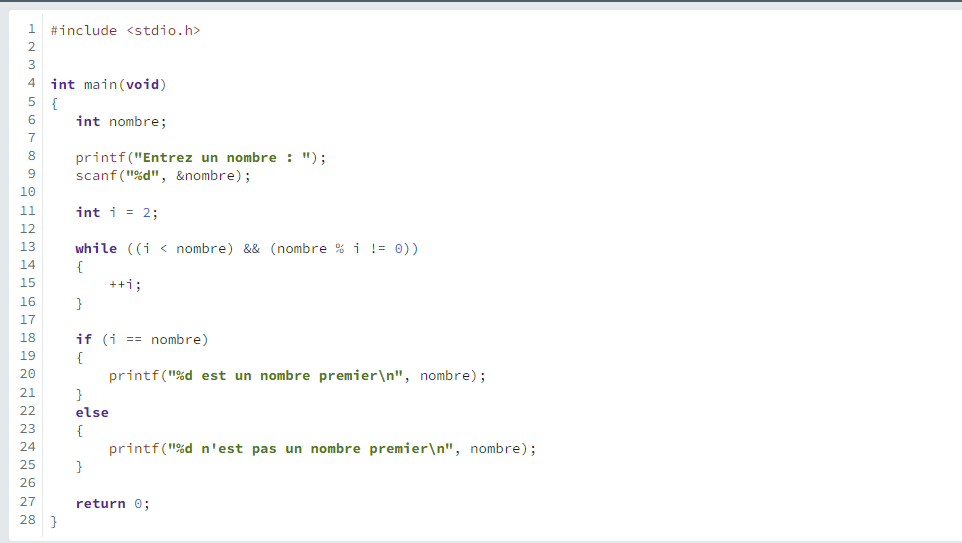
**Travaux pratiques 3 programmation en langage C –Les structures répétitives**

## Exercice 1

Réaliser un programme qui détermine si un nombre entré par l’utilisateur est premier. Pour rappel, un nombre est dit premier s’il n’est divisible que par un et par lui-même. Notez que si un nombre *x* est divisible par *y* alors le résultat de l’opération x % y est nul.

*Corrigé.*

**

## Exercice 2

Réaliser un programme qui calcule la somme de tous les nombres compris entre un et *n* (*n* étant déterminé par vos soins). Autrement dit, pour un nombre *n* donné, vous allez devoir calculer 1+2+3+...+(n−2)+(n−1)+n.

Demander à l’utilisateur de saisir un nombre entier.

Afficher si ce nombre est pair ou impair (le reste de la division entière de ce nombre par deux égal à 0 ou non).

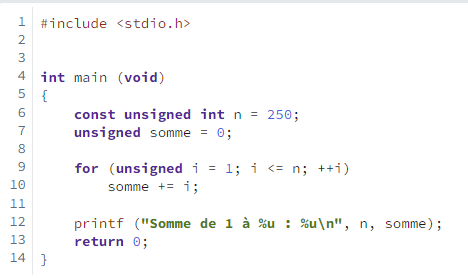
Pour obtenir le reste d'une division, on utilise l'opérateur "Modulo". En langage C l'opérateur Modulo est représenté par le symbole % :

*int a = 14;*

*int b = 5;*

*int reste = a % b; // suite à cette opération, reste = 4 (14/5 = 2 et il reste 4)*

*Corrigé.*

**

## Exercice 3

##### Calcul du PGCD de deux nombres

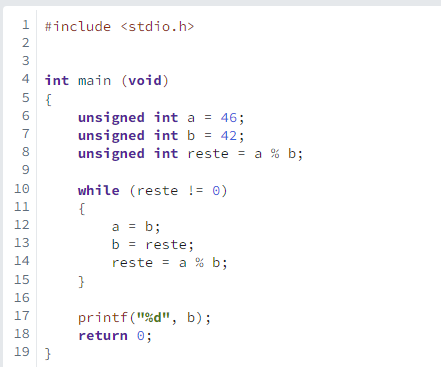
Le PGCD de deux nombres est le plus grand nombre qui peut diviser ces derniers. Par exemple, le PGCD de quinze et douze est trois et celui de vingt-quatre et dix-huit est six.

Pour le calculer, nous devons disposer de deux nombres *a* et *b* avec *a* supérieur à *b*. Ensuite, nous effectuons la division entière de *a* par *b*.

* si le reste est nul, alors nous avons terminé ;
* si le reste est non nul, nous revenons au début en remplaçant �*a* par �*b* et �*b* par le reste.

Avec cette explication, vous avez tout ce qu’il vous faut : à vos claviers !

*Corrigé.*

**

## Exercice 4

##### Une overdose de lapins

Au treizième siècle, un mathématicien italien du nom de Leonardo Fibonacci posa un petit problème dans un de ses livres, qui mettait en scène des lapins. Ce petit problème mit en avant une suite de nombres particulière, nommée la [suite de Fibonnaci](http://fr.wikipedia.org/wiki/Suite_de_Fibonacci), du nom de son inventeur. Il fit les hypothèses suivantes :

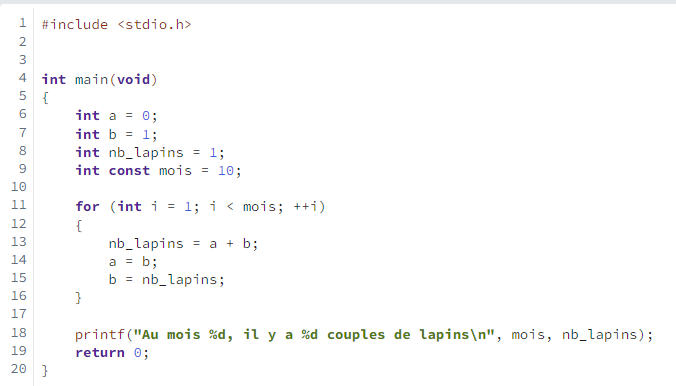
* le premier mois, nous plaçons un couple de deux lapins dans un enclos ;
* un couple de lapin ne peut procréer qu’à partir du troisième mois de sa venue dans l’enclos (autrement dit, il ne se passe rien pendant les deux premiers mois) ;
* chaque couple capable de procréer donne naissance à un nouveau couple tous les mois ;
* enfin, pour éviter tout problème avec la SPA, les lapins ne meurent jamais.

Le problème est le suivant : combien y a-t-il de couples de lapins dans l’enclos au n-ième mois ? Le but de cet exercice est de réaliser un petit programme qui fasse ce calcul automatiquement.

*Indice*

Un petit coup de pouce : suivant l’énoncé, un couple ne donne naissance à un autre couple qu’au début du troisième mois de son apparition. Combien de couple y a-t-il le premier mois ? Un seul. Combien y en a-t-il le deuxième mois ? Toujours un seul. Combien y en a-t-il le troisième mois (le premier couple étant là depuis deux mois) ? Deux. Avec ceci, vous devriez venir à bout du problème.

*Corrigé.*



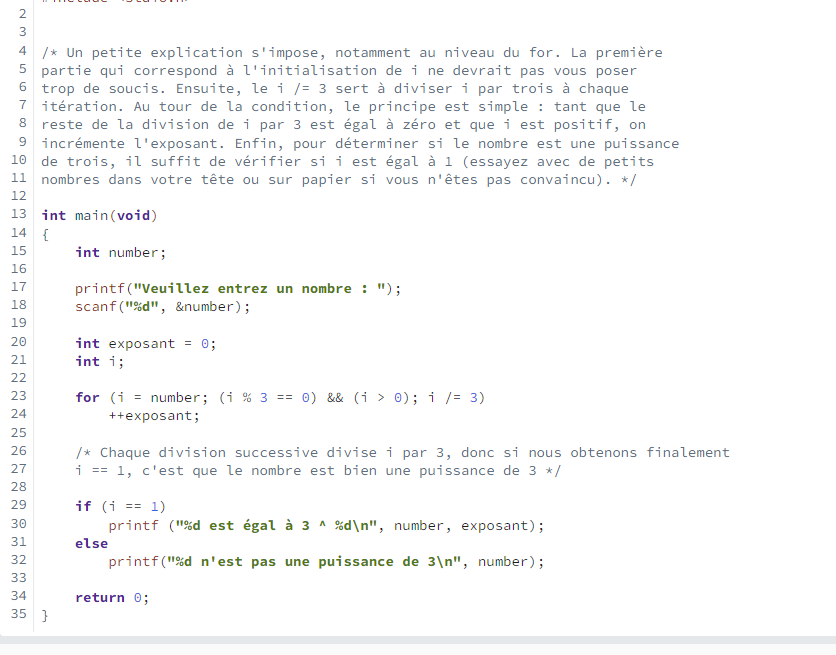
## Exercice 5

Vérifier si un nombre est une puissance de trois, et afficher le résultat. De plus, si c’est le cas, afficher l’exposant qui va avec.

Indice.

Pour savoir si un nombre est une puissance de trois, vous pouvez utiliser le modulo. Attention cependant : si le reste vaut 0, le nombre n’est pas forcément une puissance de trois (par exemple, le reste de la division de 15 par 3 est nul, mais 15 n’est pas une puissance de trois).

*Corrigé.*

**

## Exercice 6

Développer un programme qui demande à l’utilisateur de taper une phrase, puis qui affiche le nombre de « e » qu’il y a dans celle-ci. Une phrase se termine toujours par un point « . », un point d’exclamation « ! » ou un point d’interrogation « ? ». Pour effectuer cet exercice, il sera indispensable de lire la phrase caractère par caractère.

Entrer une phrase : Bonjour, comment allez-vous ?

Au moins une lettre 'e' a été repérée (précisément : 2) !

Corrigé.

