



Chapitre 1 – Opérateurs et boucles for  
Operatoren und for-Schlaufen

Exercice Algo 1.4

Dans cet exercice nous voulons essayer l'utilisation de quelques opérateurs en C, utiliser une boucle for pour répéter des instructions et nous familiariser avec l'outil Code::Blocks.

In dieser Aufgabe wollen wir die Verwendung von einigen Operatoren in C ausprobieren, for-Schlaufen zur Repetition von Anweisungen verwenden und erste Schritte mit dem Tool Code-Blocks unternehmen.

- a) Déclarez globalement un **tableau** de chaînes de caractères **constant**, initialisez le en utilisant des noms de fruits en minuscules et affichez, en utilisant une boucle for, le contenu du tableau entier sur la console  
Deklarieren Sie eine **konstante Tabelle** von Zeichenketten global, initialisieren Sie diese mit kleingeschriebenen Namen von Früchten und geben Sie den Inhalt der ganzen Tabelle mittels einer for-Schleife auf der Konsole aus
- b) Adaptez la fonctionnalité de l'exercice précédant (sans adapter la déclaration) tel que le premier caractère du fruits est en plus indiqué en **majuscule** ci-après  
Passen Sie die Funktionalität der vorherigen Aufgabe (ohne die Deklaration anzupassen) so an, dass der erste Buchstabe der Frucht danach noch in einem **Grossbuchstaben** angegeben wird .
- c) Utilisez ici différents opérateurs **bit**.  
Verwenden Sie hier verschiedene **bit**-Operatoren.
- d) Utilisez différents opérateurs **logiques** et de comparaison. Pour ceci vous pouvez bien utiliser l'opérateur ternaire ?  
Verwenden Sie verschiedene **logische** und Vergleichs-Operatoren. Hierzu können sie gut den ternären Operator ? einsetzen:

```
expression ? ExpressionIfTrue : expressionIfFalse
```

### Example for Terminal output:

a) Show constant table with for-loop

```
Fruit 1) is orange
Fruit 2) is apple
Fruit 3) is banana
Fruit 4) is mango
Fruit 5) is litchi
```

b) Show constant table with for-loop - start with upper-case

```
Fruit 0) is orange - starts with O
Fruit 1) is apple - starts with A
Fruit 2) is banana - starts with B
Fruit 3) is mango - starts with M
Fruit 4) is litchi - starts with L
```

With values of cVarA=12 and cVarB=10

c) bit-operators

```
Result of bit operation OR (cVarA|cVarB) is: 14
Result of bit operation AND (cVarA&cVarB) is: 8
Result of bit operation XOR (cVarB^cVarA) is: 6
Result of bit operation INV (~cVarA) is: -13
```

d) logical and comparison operators

```
Result of logical operation (cVarA OR cVarB) is: true
Result of logical operation (cVarA AND cVarB) is: true
Result of logical operation (NOT(cVarA==cVarB)) is: true
Result of comparison (cVarA==cVarB) is: false
Result of comparison (cVarA!=cVarB) is: true
Result of comparison (cVarA>cVarB) is: true
Result of comparison (cVarA<cVarB) is: false
```

Process returned 0 (0x0) execution time : 0.099 s