



Chapitre 1 – Factoriel d'un nombre: $n!$
Fakultät einer Zahl: $n!$

Exercice Algo 1.2

Dans cet exercice nous voulons implémenter un algorithme qui nous calcule le nombre factoriel d'un nombre quelconque. On va utiliser des fonctions et des structures de contrôle du code.

In dieser Übung wollen wir einen Algorithmus implementieren, welche die Fakultät einer beliebigen Zahl berechnet. Hierzu werden wir Funktionen und Code-Kontrollstrukturen verwenden.

$$n! = \prod_{k=1}^n k, \quad \text{ou} \quad n! = \begin{cases} 1 & \text{if } n = 0, \\ (n-1)! \times n & \text{if } n > 0. \end{cases}$$

<https://en.wikipedia.org/wiki/Factorial>

```
Algo-Ex_1_2-Factorial\bin\Debug>Algo-Ex_1_2-Factorial.exe
The factorial of 0 using loop is: 1
The factorial of 1 using loop is: 1
The factorial of 2 using loop is: 2
The factorial of 3 using loop is: 6
The factorial of 4 using loop is: 24
The factorial of 5 using loop is: 120
The factorial of 6 using loop is: 720
The factorial of 7 using loop is: 5040
The factorial of 8 using loop is: 40320
The factorial of 9 using loop is: 362880
The factorial of 10 using loop is: 3628800
The factorial of 11 using loop is: 39916800
The factorial of 12 using loop is: 479001600
The factorial of 13 using loop is: 6227020800
The factorial of 14 using loop is: 87178291200
The factorial of 15 using loop is: 1307674368000
The factorial of 16 using loop is: 20922789888000
The factorial of 17 using loop is: 355687428096000
The factorial of 18 using loop is: 6402373705728000
The factorial of 19 using loop is: 121645100408832000
The factorial of 20 using loop is: 2432902008176640000
The factorial of 0 using recursion is: 1
...
```

Notes/Bemerkungen:

- On peut utiliser `unsigned long long` et `%I64u` avec `printf` pour des résultats plus grands
- Für grössere Resultate kann man den Datentyp `unsigned long long` zusammen mit `%I64` für `printf` verwenden.

- a) Implémentez une **fonction** qui calcule le factoriel d'un nombre passé comme paramètre à l'aide d'une **itération**, faites l'appel de celle-ci dans main() et affichez le résultat des facultés de 0 à 20 avec printf.
- Implementieren Sie eine **Funktion** welche die Fakultät einer als Parameter übergebenen Zahl mittels einer **Iteration** berechnet, rufen Sie diese in main() auf und zeigen Sie das Resultat für die Fakultäten von 0 bis 20 mittels printf an.

```
#include <stdio.h>

typedef unsigned long long Argument_t;

Argument_t getFactorialWithLoop(unsigned int n)
{
    Argument_t result=1;
    for (unsigned int i=1; i<=n;i++) {
        result = result * i;
    }
    return result;
}

int main()
{
    // a)
    for (unsigned int n=0;n<=20;n++) {
        printf("The factorial of %2d using loop is: %I64u\n",
            n, getFactorialWithLoop(n));
    }
    return 0;
}
```

Solution alternative

```
for (unsigned int n=1; n<=number; n++) {
    result *= n;
}
```

- b) Implémentez une **fonction** qui calcule le factoriel d'un nombre passé comme paramètre à l'aide d'une **réursion**, faites l'appel de celle-ci dans main() et affichez le résultat avec printf.
- Implementieren Sie eine **Funktion** welche die Fakultät einer als Parameter übergebenen Zahl mittels **Rekursion** berechnet, rufen Sie diese in main() auf und zeigen Sie das Resultat mittels printf an.

```
Argument_t getFactorialRecursive(unsigned int n)
{
    Argument_t result=1;
    if (n>1) {
        result = n*getFactorialRecursive(n-1);
    }
    return result;
}

int main()
{
    // b)
    for (unsigned int n=0;n<=20;n++) {
        printf("The factorial of %2d using recursion is: %I64u\n",
            n, getFactorialRecursive(n));
    }
    return 0;
}
```