

## Aufgabe 1:

Es soll eine Tabelle erstellt werden, die die Ergebnisse von Exponenten und ihren Potenzen enthält.

<pre>#include &lt;stdio.h&gt;  long long berechnen (int exponent, int potenz) {     int zaehler;     long long ergebnis = 1;      for (zaehler = 1; zaehler &lt;= potenz; zaehler++)     {         ergebnis *= exponent;     }      return ergebnis; }  int main() {     int zeile;     int spalte;     int exponententen;     int potenzen;      exponenten = 20;     potenzen = 3;      for (zeile = 0; zeile &lt;= exponenten; zeile++)     {         if (zeile == 0)             printf ("hoch     ");          else         {             printf ("%4d\t", zeile);             printf (" ");         }          for (spalte = 0; spalte &lt;= potenzen; spalte++)         {             if (zeile == 0)                 printf ("%12d\t", spalte);             else                 printf ("%12Lu\t", berechnen (zeile, spalte)); "berechnen mit den aktuellen Werten ausgegeben             printf (" ");         }         printf ("\n");          if (zeile == 0)         {             printf ("-----+-----");             for (spalte = 1; spalte &lt;= potenzen; spalte++)             {                 printf ("-----");             }             printf ("\n");         }     }      return 0; }</pre>	<p>selbst definierte Funktion zum Berechnen einer Potenz (long long für besonders große Zahlen) diese Variable wird für die gleich folgende Schleife benötigt hier wird das Ergebnis festgehalten</p> <p>Schleife für das Multiplizieren (bei <math>5^2</math> wird die Schleife zweimal durchlaufen) ermitteltes Ergebnis mit dem Exponenten multiplizieren</p> <p>Ergebnis an den Aufruf zurückgeben</p> <p>Hauptmethode</p> <p>Zeile für die Tabelle Spalte für die Tabelle Exponent-Variable Potenz-Variable</p> <p>Exponent wird definiert (da keine Abfrage) Potenz wird definiert (da keine Abfrage)</p> <p>Schleife für die Zeilen der Tabelle</p> <p>Wenn Zeile "0" ist, wird der Tabellenkopf ausgegeben hier beginnt die Beschriftung des Tabellenkopfes</p> <p>Wenn Zeile nicht "0" ist, wird die Zeilennummer ausgegeben</p> <p>Schleife für die Spalten der Tabelle</p> <p>Wenn Zeile "0" ist, werden die Spaltennummern ausgegeben</p> <p>Wenn Zeile nicht "0" ist, wird das Ergebnis der Funktion "berechnen mit den aktuellen Werten ausgegeben</p> <p>Zeilenschaltung, damit nicht alles in einer Zeile ausgegeben wird eine Linie unter dem Tabellenkopf - wird natürlich auch nur ausgegeben, wenn Zeile "0" ist</p> <p>Rückwert "0" für erfolgreiches Beenden des Programms</p>
--	--

## Code mit Kommentaren:

```
#include <stdio.h>

// Funktion "berechnen" zur Berechnung des // Ergebnisses aus Exponent und Potenz
long long berechnen (int exponent, int potenz)
{
    // Variablendefinitionen
    // für das Ergebnis wird vorsorglich "long long" deklariert
    int zaehler;
    long long ergebnis = 1;

    // Schleife: Nimm den Exponenten Potenz-mal mit sich selbst mal
    for (zaehler = 1; zaehler <= potenz; zaehler++)
    {
        ergebnis *= exponent;
    }

    // Ergebnis zurückgeben an Funktionsaufruf
    return ergebnis;
}

// main-Methode starten
int main()
{
    // Variablendefinitionen
    int zeile;
    int spalte;
    int exponenten;
    int potenzen;

    // Die Exponente und die Potenzen, deren Ergebnisse tabellarisch
    // ausgegeben werden sollen, werden hier festgelegt
    exponenten = 20;
    potenzen = 3;

    // Schleife für die Zeilen
    for (zeile = 0; zeile <= exponenten; zeile++)
    {
        // Wenn die Zeile "0" ist, einen Kopf bauen
        if (zeile == 0) {
            printf ("hoch   |");
        }

        // Zeile ist nicht "0", dann den aktuellen Exponenten angeben
        else
        {
            printf ("% 4d\t", zeile);
            printf ("|");
        }

        // Inhalt der Spalten
        for (spalte = 0; spalte <= potenzen; spalte++)
        {
            if (zeile == 0)
                printf ("%12d\t", spalte);
            else
                printf ("%12Lu\t", berechnen (zeile, spalte));
            printf ("|");
        }
        printf ("\n");

        // "Nachschliff" für die Tabelle (wird nur ausgeführt, wenn Zeile "0" ist)
        if (zeile == 0)
        {
            printf ("-----+-----+");
            for (spalte = 1; spalte <= potenzen; spalte++)
            {
                printf ("-----+");
            }
            printf ("\n");
        }
    }

    // Fehlercode 0 zurückgeben
    return 0;
}
```

## Ausgabe:

hoch	0	1	2	3
1	1	1	1	1
2	1	2	4	8
3	1	3	9	27
4	1	4	16	64
5	1	5	25	125
6	1	6	36	216
7	1	7	49	343
8	1	8	64	512
9	1	9	81	729
10	1	10	100	1000
11	1	11	121	1331
12	1	12	144	1728
13	1	13	169	2197
14	1	14	196	2744
15	1	15	225	3375
16	1	16	256	4096
17	1	17	289	4913
18	1	18	324	5832
19	1	19	361	6859
20	1	20	400	8000