

## Addieren mit zwei überlangen Zahlen:

scanf("%s", s1); // Hier wird in **s1** eingelesen → Eingabe: 19019 → Ascii: 49|57|48

char s1[21];

49	57	48	49	57	\0	.....	\0
s1[0]							s[20]

scanf("%s", s2); // Hier wird in **s2** eingelesen → Eingabe: 149 → Ascii: 49|52|57

char s2[21];

49	52	57	\0	\0	\0	.....	\0
s2[0]							s[20]

```
if(strlen(s1) > strlen(s2))
{
    strcpy(ss1,s1);

    for(int i = 0; i < strlen(s2); i++)
    {
        ss2[i + (strlen(s1) - strlen(s2))] = s2[i];
    }
    for( i = 0; i < (strlen(s1) - strlen(s2)); i++)
    {
        ss2[i] = 48;
    }
    ss2[strlen(ss2)] = '\0';
}
```

char ss1[21]; (entspricht Zahl: 19019)

49	57	48	49	57	\0	\0
ss1[0]						ss1[6]

char ss2[21]; (entspricht Zahl: 00149 → 2 NULLEN wurden vorangestellt !)

48	48	49	52	57	\0	\0
ss1[0]						ss1[6]

Wenn s1 größer s2 ist, dann schreibe ich s1 direkt in ss1. s2 hingegen muss ich, um die Differenz von (s1 - s2) versetzt, in ss2 schreiben.(→ 1.for-Schleife).Dann fülle ich ss2 bis zur ersten Ziffer mit NULLEN auf. (→ 2.for-Schleife).Zum Schluss muss ich noch den String mit '\0' abschließen.

```
if(strlen(s1) > strlen(s2))
{
    strcpy(ss1,s1);

    for(int i = 0; i < strlen(s2); i++)
    {
        ss2[i + (strlen(s1) - strlen(s2))] = s2[i];
    }
    for( i = 0; i < (strlen(s1) - strlen(s2)); i++)
    {
        ss2[i] = 48;
    }
    ss2[strlen(ss2)] = '\0';
}
```

Ist s2 größer als s1, mache ich es genau umgekehrt!

**Errechnen der Addition:** Ergebnis wird in **ergebnis** geschrieben.

```
int uebertrag = 0;

for(int i = 0; i < strlen(ss1); i++)
{
    ergebnis[strlen(ss1)-i] = ( ( ss1[strlen(ss1)-1-i]-48 + ss2[strlen(ss2)-1-i]-48 + uebertrag) %10 )+48;
    uebertrag = (ss1[strlen(ss1)-1-i]-48 + ss2[strlen(ss2)-1-i]-48 + uebertrag) / 10;
}

// Wenn Übertrag > 0, dann wird er an die erste Stelle des Ergebnisses geschrieben.
// Ansonsten wird ein Leerzeichen eingefügt.
if(uebertrag > 0)
{
    ergebnis[0] = uebertrag + 48;
}
else
{
    ergebnis[0] = ' ';
}

char ergebnis[22]; (entspricht Zahl: 19168)
```

32	49	57	49	54	56	\0
[0]	...	...	...	...	...	[6]

Ich beginne bei den letzten beiden Stellen von `ss1` und `ss2`, den Inhalt zu addieren.  
Der Inhalt entspricht aber den Ascii-Werten, also alle Zahlen + 48, deshalb ziehe ich jeweils 48 ab, und errechne mir anschließend den Ergebniswert an dieser Stelle sowie den Übertrag.  
Der Stellenwert bei der letzten Stelle beträgt 8 →  $9 + 9 = 18$ , 18 modulo 10 ist 8.  
Der Übertrag wird durch den Ganzzahlwert der Division durch 10 bestimmt, ist also in diesem Fall 1.  
Das Ergebnis an der Stelle `i` schreibe ich in das Array an die Stelle `i + 1`, um am Ende die nullte Stelle (= `ergebnis[0]`) mit dem Übertrag aufzufüllen, falls dieser nicht 0 ist.  
Ist er 0, dann fülle ich zwecks Formatierung `ergebnis[0]` mit einem Leerzeichen (= AsciiWert 32).