

Algorithmen und Programmierung

Summe von Zahlen mit unterschiedlichen Basen (Trainingsaufgabe)

Die folgende Datenstruktur dient zur Darstellung von positiven Zahlen mit Basis `base` als Zeichenketten:

```
1 typedef struct {  
2     char *number;  
3     unsigned int base;  
4 } basedNumber;
```

Es sollen nur Zahlen mit Basen $\in [2, 36]$ dargestellt werden. Bei Basen größer als 10 werden zur Darstellung jeder neuen Ziffer lateinische Großbuchstaben genutzt.

Beispiele:

- $100_2 = 4_{10} \Rightarrow \text{*number} = \text{"100"}, \text{base} = 2$
- $101_8 = 65_{10} \Rightarrow \text{*number} = \text{"101"}, \text{base} = 8$
- $111_3 = 13_{10} \Rightarrow \text{*number} = \text{"111"}, \text{base} = 3$
- $1A_{11} = 21_{10} \Rightarrow \text{*number} = \text{"1A"}, \text{base} = 11$
- $AB_{16} = 171_{10} \Rightarrow \text{*number} = \text{"AB"}, \text{base} = 16$
- $Z_{36} = 35_{10} \Rightarrow \text{*number} = \text{"Z"}, \text{base} = 36$
- $\text{HALLO}_{36} = 29048028_{10} \Rightarrow \text{*number} = \text{"HALLO"}, \text{base} = 36$

Aufgabe:

Implementieren Sie die Funktion

```
unsigned int sumOfBasedNumbers(basedNumber bn[], unsigned int size);
```

die die Summe von Zahlen mit unterschiedlichen Basen berechnet. Die gesuchte Summe von allen Zahlen soll in der Dezimaldarstellung von der Funktion zurückgegeben werden.

Beispiel:

```
sumOfBasedNumbers({{"FF", 16}, {"100", 10}}, 2) = 255 + 100 = 355.
```

Die Funktion soll in der Datei `basedNumbers.c` implementiert werden. Dazu sei eine Headerdatei `basedNumbers.h` verfügbar, welche die oben genannte Datenstruktur bereitstellt.

Hinweise zur Aufgabenstellung

Für die Lösung dieser Aufgabe benötigen Sie folgende Grundkenntnisse:

- Kontrollfluss in C (if)
- Schreiben von Funktionen
- Schleifen in C
- Felder in C
- Datenstrukturen (struct) in C
- Benutzung von gcc

Hinweise zur Abgabe

- Erstellen Sie eine ZIP- bzw. TGZ-Archivdatei, welche die geforderten Dateien enthält.
- Fügen Sie dem Archiv keine weiteren Dateien oder Ordner hinzu.
- Reichen Sie Ihre Lösung unter <https://osg.informatik.tu-chemnitz.de/submit> ein.
- Bis zum Abgabende (Deadline), sofern gegeben, können beliebig neue Lösungen eingereicht werden, die die jeweils älteren Versionen ersetzen.
- Ihr Programm muss auf der Testmaschine übersetzbar sein. Deren Details sind auf dem OpenSubmit-Dashboard verfügbar.
- Ihre Lösung wird automatisch validiert. Sie werden über den Abschluss der Validierung per eMail informiert.