



Aufgabe 1. Werde zunächst mit Aufgabe 4 von Zettel 5 fertig. Implementiere dann noch die folgenden Stringfunktionen:

```
1  /* Allokiere genügend Speicher für eine Kopie von s
2   * und kopiere den Inhalt von s dorthin. Liefere als
3   * Rückgabe die Anfangsadresse dieser Kopie. */
4  char *str_dup(char *s);
5
6  /* Setze s auf den String, der aus genau n Leerzeichen
7   * besteht. */
8  char *str_spaceout(char *s, unsigned int n);
```

Aufgabe 2. Implementiere die folgenden Funktionen:

```
1  /* pieces ist ein Array (von Strings) der Länge count.
2   * Hänge diese Strings, getrennt durch sep, aneinander
3   * und liefere das Ergebnis. Beispiel:
4
5   *   char *p[3] = { "Ene", "Mene", "Miste" };
6   *   char *joined = str_join( p, 3, ", ");
7   *   printf("%s", joined);
8   *   free(joined);
9
10  *   gibt aus: Ene, Mene, Miste */
11 char *str_join(char **pieces, int count, char *sep);
12
13 /* Spalte str bei jedem Auftreten von sep auf und
14  * liefere das Array der entstehenden Strings.
15  * Wäre etwa str der String "Ene, Mene, Miste" und
16  * sep der String ",", so sollte das zurückgegebene
17  * Array die Strings
18
19  *   "Ene"      " Mene"      " Miste"
20
21  *   enthalten (man bemerke die Leerzeichen). In count
22  *   wird die Länge des Arrays gespeichert, in diesem
23  *   Fall drei. Der String str selbst sollte von der
24  *   Funktion natürlich nicht verändert werden. */
25 char **str_split(char *str, char *sep, int *count);
```



Aufgabe 3. Implementiere einige Funktionen um mit quadratischen Matrizen umzugehen:

- a) Eine Funktion, die Speicher für eine quadratische Matrix allokiert, eine um ihn freizugeben, eine um sie auszugeben und eine um sie zur Einheitsmatrix zu initialisieren (das ist die Matrix mit 1en auf der Hauptdiagonale und 0en sonst):

```
1 double **matrix_alloc(int n);  
2 void      matrix_free(double **A, int n);  
3 void      matrix_print(double **A, int n);  
4 double **matrix_id(double **A, int n);
```

- b) Eine Funktion um eine Matrix zu transponieren (d.h. an der Hauptdiagonale “zu spiegeln”)
- c) Eine Funktion, die zwei solche Matrizen miteinander multipliziert und eine neue Matrix zurück gibt. Für zwei $n \times n$ -Matrizen $A = (a_{ij})$ und $B = (b_{ij})$ ist $A \cdot B = C = (c_{ij})$ über folgende Formel definiert:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$