

---

## SPiC-Aufgabe #6: concat

(10 Punkte, keine Gruppen)

Schreiben Sie ein Programm `concat` (concatenate), das die per Kommandozeile übergebenen Parameter in einem dynamisch allokierten Buffer zu einer verketteten Zeichefolge zusammensetzt und anschließend auf dem Standardausgabekanal ausgibt. Die einzelnen Parameter werden dabei durch ein Leerzeichen voneinander abgegrenzt.

### Beispiele:

```
$ ./concat Dies ist ein Test
Dies ist ein Test
```

```
$ ./concat   Foo   Bar
Foo Bar
```

### Programmablauf:

Das `concat` Porgramm soll in folgenden Schritten ablaufen:

1. Iteration über die Parameter und Ermittlung der Gesamtlänge mit Hilfe von `str_len`
2. Dynamische Allokation von Speicher für die zusammengesetzte Zeichnfolge mit `malloc(3)`
3. Zusammensetzen der Zeichenkette mit den Hilfsfunktionen `str_cpy` und `str_cat`
4. Einmaliger Aufruf von `printf(3)` zur Ausgabe des Gesamtstrings
5. Freigabe des reservierten Speichers mit `free(3)`

### Hinweise:

- Implementieren Sie `str_len`, `str_cpy` und `str_cat` analog zu den Bibliotheksfunktionen `strlen(3)`, `strcpy(3)` und `strcat(3)`. Ein Aufruf der Original-Bibliotheksfunktionen ist nicht zulässig.
- Achten Sie auf aussagekräftige Fehlermeldungen, die alle auf dem Standardfehlerkanal ausgegeben werden sollen. (`fprintf(stderr, ...)(3)` / `perror(3)`)
- Übersetzen Sie das Programm mit `gcc -g -pedantic -Wall -Werror -std=c99 -o concat concat.c`
- Testen sie ihr Programm mit `valgrind ./concat foo bar`. Dies kann auch bei der Suche nach Fehlern helfen. ‘suppressed Errors’ können ignoriert werden.

### Abgabezeitpunkt

T01	02.07.2017	18:00:00
T02	02.07.2017	18:00:00
T03	03.07.2017	18:00:00
T04	03.07.2017	18:00:00
T05	27.06.2017	18:00:00
T06	27.06.2017	18:00:00
T07	27.06.2017	18:00:00
T08	28.06.2017	18:00:00
T09	28.06.2017	18:00:00