

Aufgabe 7.1

Schreiben Sie ein MIPS32-Programm, das folgende C-Funktion realisiert:

```
int ncstr(char *str, char c)
```

Die Routine soll in einem C-String an der Adresse `str` die Anzahl der vorkommenden Character `c` zählen und zurückgeben.

Schreiben Sie dann ein Programm, das

- den Benutzer auffordert, einen String einzugeben,
- den String einliest,
- den Benutzer auffordert, einen Character einzugeben,
- den Character einliest,
- `ncstr` mit den entsprechenden Argumenten aufruft,
- die Zahl der Character im String ausgibt und
- den Benutzer fragt, ob das Programm noch einmal ausgeführt werden soll.

Ein Dialog könnte etwa so aussehen:

```
Type a string>Mississippi
Type a char>i
4 occurrence(s)
Type a string>Main
Type a char>m
0 occurrence(s)
```

Für die Ausgabe der Prompts und des Integers, sowie das Einlesen des Strings und des Characters, können Sie die System Calls 4, 1, 8 und 12 benutzen. Für die Abfrage, ob das Programm beendet werden soll, empfiehlt sich der `ConfirmDialog`, System Call 50.

Aufgabe 7.2

Ein MIPS Unterprogramm besteht aus folgenden vier Instruktionen, die als Instruktionswörter in Hexadezimalschreibweise gegeben sind:

```
2008ffff
00a84004
00881024
03e00008
```

- Geben Sie die vier Instruktionen als Programm in der Syntax für den MARS Assembler an (Disassemblierung)! Nehmen Sie dafür das „Instruction Set Manual“ zu Hilfe! Die Instruktionskodierungen können im letzten Teil des Handbuchs unter „Instruction Bit Encodings“ nachgeschlagen werden.
- Verifizieren Sie, dass der Mars Assembler aus Ihrem Programm die vier gegebenen Instruktionswörter generiert!
- Was berechnet das Unterprogramm?