Aufgabe 7.1 (Unterprogramme, System Calls)
Es soll ein MIPS32-Programm geschrieben werden, das die Ein- und Ausgabe mit System Calls realisiert und eine Funktion `ncstr` (Unterprogramm) aufruft.

a) Erstellen Sie ein Unterprogramm, 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 Zeichen `c` zählen und zurückgeben. Beachten Sie die Konventionen zur Übergabe von Argumenten und Rückgabewerten, sowie zur Benutzung von Registern! Auf Daten darf nur über die der Funktion übergebenen Argumente zugegriffen werden.

b) Erstellen Sie das Programm zur Ein- und Ausgabe der Daten, das Ihre Funktion `ncstr` aus Aufgabenteil 1) mit den ensprechenden Argumenten aufruft, also nacheinander

i) auffordert, eine Zeichenkette einzugeben,
ii) die Zeichenkette einliest,
iii) auffordert, ein einzelnes Zeichen einzugeben,
iv) das Zeichen einliest,
v) die Funktion `ncstr` mit den entsprechenden Argumenten aufruft,
vi) die Anzahl der Zeichen in der Zeichenkette ausgibt und
vii) fragt, ob das Programm noch einmal ausgeführt werden soll.

Ein Dialog könnte etwa so aussehen:

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

Für die Ausgabe der Prompts und der Anzahl der Zeichen in der Zeichenkette, sowie das Einlesen der Zeichenkette und des einzelnen Zeichens können Sie die System Calls 4, 1, 8 und 12 benutzen. Für die Abfrage, ob das Programm beendet werden soll, kann auch der `ConfirmDialog` benutzt werden, System Call 50.

Aufgabe 7.2

Die Instruktionswörter der beiden nebenstehenden MIPS-Unterprogramme sind in Hexadezimalschreibweise gegeben und mit Marken (Labels) versehen.

```
func1:  2008ffff
p:      00a84004
q:      00881024
r:      03e00008
```

```
func2:  20080001
p:      2002ffff
q:      00884824
r:      20420001
s:      00084040
t:      1520fffc
u:      03e00008
```

a) Geben Sie die beiden Folgen von Instruktionswörtern symbolisch in der Syntax für den MARS Assembler an (Disassemblierung)! Die Kodierung der Instruktionen kann im letzten Teil des Handbuchs „Instruction Set Manual“ unter „Instruction Bit Encodings“ nachgeschlagen werden. Wenn nötig, benutzen Sie zur Kennzeichnung von Sprungzielen die angegebenen Marken (Labels)!

b) Überprüfen Sie, dass MARS aus Ihren Programmen die gegebenen Instruktionswörter generiert!

c) Was berechnen die Funktionen?