Thema: Erste Schritte in C, Entwicklungsumgebung, Debugger

Letzter Abgabetermin: Am Anfang des zweiten Praktikumstermins

Aufgabe:[1] In dieser Aufgabe wird ein einfacher Reverse Polish Notation (RPN) Taschenrechner für das TI-Board entwickelt. Ein RPN-Rechner arbeitet Stack-basiert. Operatoren wie z.B. +, -, *, / nehmen die beiden obersten Einträge vom Stack, wenden die Operation auf sie an und legen das Ergebnis wieder auf dem Stack ab. Im Internet finden sie viele Erläuterungen zur genauen Arbeitsweise, z. B.:

http://de.wikipedia.org/wiki/Umgekehrte_Polnische_Notation

Eingabe: Zur Eingabe wird das Touch-Display des TI-Boards verwendet. In Emil finden Sie zu dieser Aufgabenstellung das Archiv keypad.zip (http://www.elearning.haw-hamburg.de/mod/resource/view.php?id=600927). Sie enthält den Code einer kleinen Tastatur. Fügen Sie diese Dateien zum Projekt hinzu. Mit dem Aufruf von `Make_Touch_Pad()` wird auf dem Display die Tastatur gezeichnet und der Touch aktiviert. Das Betätigen einer Taste kann über die Funktion `Get_Touch_Pad_Input()` abgefragt werden. Der Rückgabewert dieser Funktion enthält den Tastaturkode.

Ausgabe: Auf dem LCD-Display des TI-Boards werden die Einträge des Stacks dargestellt. In Emil finden Sie unter dem Eintrag `Die Bibliothek fuer das TI Board` (http://www.elearning.haw-hamburg.de/mod/resource/view.php?id=600810) die Beschreibung der notwendigen Bibliotheksfunktionen. Wählen Sie eine sinnvolle Darstellung, wenn nur ein Teil des Stacks auf dem Display darstellbar ist.

Funktionsumfang: Es sollen folgender Funktionsumfang implementiert werden:

- **+** nimmt zwei Werte vom Stack, addiert sie und legt das Ergebnis wieder auf dem Stack ab.
- **-** nimmt zwei Werte vom Stack, subtrahiert den ersten (oberen) Wert vom zweiten und legt das Ergebnis wieder auf dem Stack ab.
- ***** nimmt zwei Werte vom Stack, multipliziert sie und legt das Ergebnis wieder auf dem Stack ab.
- **/** nimmt zwei Werte vom Stack, teilt den zweiten Wert durch den ersten (ganzzahlige Division) und legt das Ergebnis wieder auf dem Stack ab.
- **p** druckt den obersten Wert des Stacks aus.
- **f** druckt den gesamten Stack aus.
- **c** löscht alle Einträge des Stacks
- **d** dupliziert den obersten Eintrag
- **r** vertauscht die Reihenfolge der beiden obersten Einträge des Stacks.
- Es soll nur die Eingabe von positiven ganzen Zahlen implementiert werden – trotzdem müssen Sie negative Zahlen darstellen und verarbeiten können.
  Sollen zwei Zahlen hintereinander auf den Stack geschrieben werden, werden diese durch die Eingabe der Enter Teste getrennt.
- Die Umwandlung der eingegebenen Zahlen in binäre Werte soll manuell Zeichen für Zeichen erfolgen. Es dürfen hierfür keine Funktionen aus der C-Bibliothek

[1] Diese Aufgabe basiert auf Unterlagen zum GSP Praktikum von Prof. Dr. Heiner Heitmann, HAW Hamburg.

verwendet werden (z. B. atoi, strltod, scanf).

Beispiel: Die Eingabesequenz: *25 8 67+r/p* soll den Wert *3* ausgeben. Bitte beachten Sie, dass hinter der 67 kein Leerzeichen steht - die Enter Taste wurde nicht gedrückt.

Unterlagen, die vor dem Praktikum verschickt werden müssen:

- Erstellen Sie ein schriftliches Konzept. **Dieses Konzept muss spätestens am Abend vor dem ersten Versuchstag via e-Mail an Silke Behn und mich geschickt werden**.
  Dieses Konzept muss folgende Themen beinhalten:
  - Analyse der Aufgabenstellung
  - Dokumentation des Programmaufbaus, z.B. als Flussdiagramm
  - Geben Sie mindestens 10 Eingabefolgen mit den zugehörigen Reaktionen an, mit denen Sie Ihre Anwendung testen wollen. Wie soll z.B. auf fehlerhafte Eingaben der Art *9+* reagiert werden?

Abnahme der Aufgabe im Praktikum:

- Das Konzept muss vorliegen.
- Kommentierter Quellcode, der dem C Coding Style (s. Emil) entspricht, muss vorliegen.
- Kommentierte Ausdrucke von Testfällen, die auch das Verhalten bei fehlerhafter Eingabe dokumentieren, müssen vorliegen.
- Sie müssen Ihr Programm erklären können.
- Sie müssen den Umgang mit dem Debugger an Ihrem Programm demonstrieren – also können Sie den Debugger auch schon zur Fehlersuche einsetzen.