

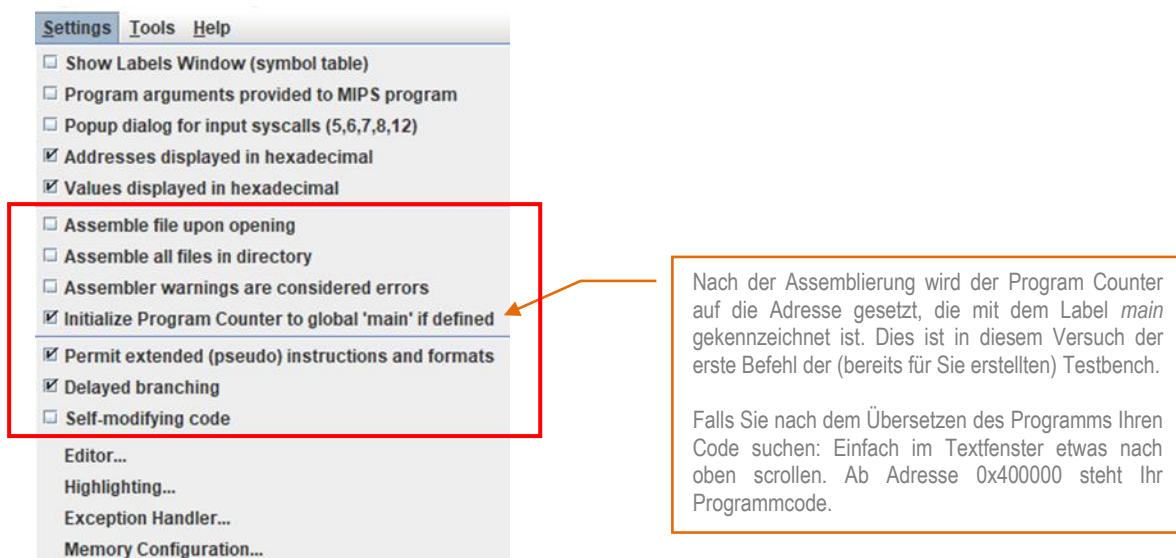
Versuch 3: MIPS-Instruktionssatz I

In diesem Versuch wird der Instruktionssatz eines MIPS-Prozessors thematisiert: Codebeispiele, die in Hochsprache formuliert sind, werden hierzu in Assembler übersetzt und getestet. In den ersten Aufgaben werden einfache Programmstrukturen wie if-Abfragen und Schleifen betrachtet. Zum Abschluss des Versuchs soll ein Algorithmus zur Berechnung der Quadratwurzel in MIPS-Assembler formuliert werden.

Hinweise:

Es sollen nur die aus der Vorlesung bekannten Befehle verwendet werden. Eine Tabelle mit den Befehlen wurde Ihnen im Rahmen der Vorlesung zur Verfügung gestellt.

Überprüfen Sie zu Beginn des Versuchs die Einstellungen des MARS-Simulators (Settings). Die korrekte Konfiguration zeigt das nachfolgende Bild:



Aufgabe 1: if-Abfrage

Gegeben ist die nachfolgende Funktion, die eine einfache if-Abfrage implementiert. Öffnen Sie die Datei [simpleIf.asm](#) im Simulator MARS. Erstellen Sie die Assemblerfunktion *simpleIf*, die die gleiche Funktion wie der untenstehende Code ausführt und testen Sie Ihr Programm. Der Aufruf der Tests (Testbench) ist bereits im vorgegebenen Code implementiert. Sie können sich also auf die Implementierung der Funktion *simpleIf* konzentrieren.

```
int simpleIf (int a, int b)
{
    int r;
    if (a<b) {
        r = a + b;
    } else {
        r = a - b;
    }
    return r;
}
```

Aufgabe 2: Fakultät

Gegeben ist die nachfolgende Funktion zur Berechnung der Fakultät einer Zahl. Öffnen Sie die Datei [fac.asm](#) im Simulator MARS. Erstellen Sie die Assemblerfunktion *fac*, testen Sie Ihr Programm und korrigieren Sie eventuelle Fehler.

```
int fac (int a)
{
    int r;
    int i;

    r=1;

    for (i=1; i<=a; i=i+1) {
        r = r * i;
    }

    return r;
}
```

Aufgabe 3: Berechnung der Quadratwurzel mit dem Heron-Verfahren

Gegeben ist die folgende Funktion, welche mit Hilfe des Heron-Verfahrens, die Quadratwurzel einer natürlichen Zahl berechnet. Öffnen Sie die Datei [squareRoot.asm](#) im Simulator MARS und erstellen Sie eine dem vorgegebenen Code entsprechende Assemblerfunktion.

Führen Sie die fehlerfreie Implementierung Ihrem Versuchsbetreuer vor.

```
int squareRoot (int a)
{
    int tmp;
    int x;

    tmp = a;

    do {
        x = tmp;
        tmp = a/x + x;
        tmp = tmp/2;
    }
    while (tmp < x);

    return x;
}
```