

- **Im ANT-File nichts von ihm ändern, wegen der Testfälle! Nur Sachen hinzufügen.**
- BackendAsmRM: Backend → 2 Offsets → (allocStaticData) der nächste freie, z.B: Globalpointer verwenden. Im Data-Segment ein Label setzen → Offset dazu addieren (zur glob. Var)
- Generell brauchen wir uns um die Deallocierung nicht kümmern
- Es ist immer nur 1 Offset für den Stack zu merken.
- Compiler geht nur 1Mal über Schleife drüber, d.h. er übersetzt eine Schleife nur 1Mal.
- Array muss im Hauptspeicher abgelegt sein.
- **storeArrayDim** kann mehrmals aufgerufen werden → Länge des Array wird zur Compilezeit nicht bekannt sein, deshalb **lenReg**, welches die Länge eines Registers enthält.
- **loadWord**: reg = Register; addr = Offset
- **arrayOffset** für die Indexberechnungen; die absolute Adresse eines Arrayelements soll zurückgegeben werden (Adresse eines Arrayelements wird berechnet)  
**baseAddr** = zeigt auf den Header der Startadresse (ByteOffset 1\* ... \*4)
- Globalpointer ist in der Main zu initialisieren
- Label für das Ende der Prozedur
- **enterProc**: Stackpointer sichern; Register sichern;
- **returnFromProc**: an das Ende eines Labels springen
- **prepareProcCall(int)**: Die Anzahl, die benötigt werden/Argumente.
- **Wir können alle Argumente an Stack übergeben.**
- **passArg**: speichert die Argumente an die richtige Position
- **\_True-Dateien sollen wir nicht ändern.**
- **\_run-backend** = die Ausführung einer Testdatei
- Backend MIPS müssen wir programmieren
- Wir sollen weitere Testfälle schreiben, sodass **ALLE** Methoden aufgerufen werden (code coverage) → testen
- Wir sollen eine README erstellen:
  1. Welche Testfälle von ihm funktionieren und welche Methoden wurden dadurch abgedeckt
  2. Welche Testfälle wurden durch eigene Testfälle abgedeckt.

Viel Spaß

Sonja ;)