**2.1) Mips Assembler**

a.) Erklären Sie, welche und wie viele Register in der MIPS-Architektur zur Verfügung stehen und wozu sie verwendet werden.

MIPS Assembler hat 31 Register. Davon sind 18 für Variablen gedacht ($t0-$t9 = temporäre Variablen, $s0-$s7 = langlebige Variablen)  
$a0 bis $a3 sind Parameter Eingabe Register für funktionen während $v0 und $v1 Ausgabe Register für Funktionen sind ($v1 z.B. für Rest).  
$zero repräsentiert 0 und sollte nicht geschrieben werden.

b.) Erklären Sie den Unterschied zwischen I-, R-, und J-Befehlen. Warum wird zwischen diesen 3 Typen unterschieden?

R-I und J Befehle müssen sich voneinander unterscheiden da sie unterschiedlich funktionieren.   
R Befehle führen Funktionen auf 2 Registern aus (2 Input Register und 1 Output Register -> z.B. add $s1, $s2, $s3 Addiert Inhalt von $s2 zu $s3 uns speichert in $s1).  
I Befehle hingegen arbeiten mit Registern und Immediate Values gleichzeitig (Also Zahlen die nicht in Registern gespeichert sind (Maximal 16 Bit) (Beispiel: addi $s1, $s2, 100 -> Addiert Inhalt von $s2 mit 100 und speichert in $s1)  
J Befehle hingegen werden für Jump Anweisungen benutzt (z.B. j *dest*)

c.) Mit welchen Befehlen werden Multiplikation und Division von ganzen Zahlen bewerkstelligt, und wo sind die Ergebnisse zu finden?

Alle Multiplikationen und Divisionen mit ganzen Zahlen werden mit 4 Befehlen realisiert (div, divu, mult, multu wobei u für Unsigned steht) und ihre Ausgabe nicht in eines der 32 Register geschrieben wird sondern in die hi und lo Register die mit den Funktionen **mflo** und **mfhi** in normale Register gespeichert werden müssen nach der Operation (z.B. **mflo $s0**). Wobei **hi** den Rest der Operation beinhaltet und **lo** das Ergebnis.