

```

1      .data
2  line1: .asciiz "Geben Sie die 1. Zahl ein: "
3  line2: .asciiz "Geben Sie die 2. Zahl ein: "
4  erg:   .asciiz "Das Ergebnis lautet: "
5  komma: .asciiz ", "
6
7      # Registerbelegungen
8      # $t0 = A
9      # $t1 = B
10     # $t2 = Summe (von A und B)
11     # $t3 = 2
12     # $t4 = Rest der Division
13     # $t5 = 10
14     # $s0 = Vorkommazahl
15     # $s1 = Nachkommazahl
16
17     .text
18  main:
19     #####
20     # Einlesen der 2 Werte
21     #####
22
23     li      $v0, 4          # 4: print_str:
24     la      $a0, line1      # Adresse der 1. auszugebende Zeile nach $a0
25     syscall                                # ausgeben
26
27     li      $v0, 5          # 5: read_int
28     syscall                                # integer einlesen
29     move    $t0, $v0        # eingelesenen Wert in Register $t0 verschieben (A)
30
31     li      $v0, 4          # 4: print_str:
32     la      $a0, line2      # Adresse der 2. auszugebende Zeile nach $a0
33     syscall                                # ausgeben
34
35     li      $v0, 5          # 5: read_int
36     syscall                                # integer einlesen
37     move    $t1, $v0        # eingelesenen Wert in Register $t1 verschieben (B)
38
39     #####
40     # Berechnung des Durchschnitts
41     #####
42
43     add     $t2, $t0, $t1    # $t2/Summe = $t0/A + $t1/B
44     li      $t3, 2          # Den Wert 2 in Register $t0 laden
45     div     $s0, $t2, $t3    # $s0/Vorkomma = $t2/Summe DIV $t3/2
46     rem     $t4, $t2, $t3    # $t4/Rest = $t2/Summe MOD $t3/2
47
48     li      $t5, 10         # Den Wert 10 in Register $t5 laden
49     mul     $t6, $t4, $t5    # Eine Null an den Rest haegen (= x 10)
50
51     div     $s1, $t6, $t3    # den mit 0 erweiterten Rest wieder durch 2 teilen
52
53     #####
54     # Ausgabe des Ergebnisses
55     #####
56
57     li      $v0, 4          # 4: print_str
58

```

59	<code>la</code>	<code>\$a0, erg</code>	<i># Adresse des Ergebnis-Texts nach la0</i>
60	<code>syscall</code>		<i># ausgeben</i>
61			
62	<code>li</code>	<code>\$v0, 1</code>	<i># 1: print_int</i>
63	<code>move</code>	<code>\$a0, \$s0</code>	<i># Vorkommastelle nach la0</i>
64	<code>syscall</code>		
65			
66	<code>li</code>	<code>\$v0, 4</code>	<i># 4: print_str</i>
67	<code>la</code>	<code>\$a0, komma</code>	<i># Adresse des Komma-Texts nach la0</i>
68	<code>syscall</code>		
69			
70	<code>li</code>	<code>\$v0, 1</code>	<i># 1: print_int</i>
71	<code>move</code>	<code>\$a0, \$s1</code>	<i># Nachkommastelle nach la0</i>
72	<code>syscall</code>		
73			
74	<code>li</code>	<code>\$v0, 10</code>	<i># Systemaufrufnr. 10 = EXIT</i>
75	<code>syscall</code>		