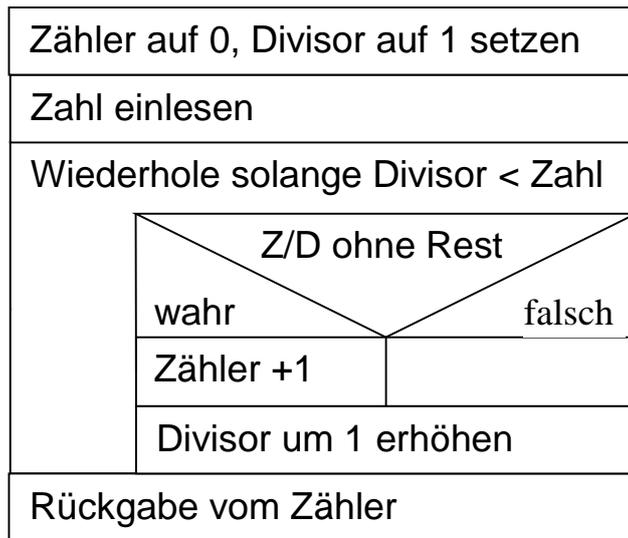


Struktogramm:JAVA-Methode:

```
int primzahltest(int t){
    int div = 2;           // damit Teiler 1 nicht gezählt wird
    int zaehler = 0;
    while (div<t){
        if (t%div == 0){
            zaehler++;
        }
        div++;
    }
    return zaehler;
}
```

Assemblerprogramm:

```
1: DLOAD 18 -- Testzahl laden
2: STORE 1  -- und in R1 speichern
3: DLOAD 1  -- Zähler laden
4: STORE 2  -- und in den Registern
5: STORE 4  -- R2 (Divisor) und R4 (Teilerzähler) speichern.
6: LOAD 1   -- Testzahl laden
7: DIV 2    -- und durch Divisor dividieren.
8: JEQ 19   -- wenn nur 0 dann ist Divisor größer als Testzahl
9: MULT 2   -- wieder mit Divisor multiplizieren
10: SUB 1    -- und Testzahl davon abziehen.
11: JNE 15   -- Wenn nicht 0 rauskommt → Rest → nix machen
12: LOAD 3   -- ansonsten Teilerzähler laden
13: ADD 4    -- und um eins erhöhen
14: STORE 3  -- und wieder speichern.
15: LOAD 2   -- Divisor
16: ADD 4    -- um eins erhöhen
17: STORE 2  -- und wieder speichern.
18: JUMP 6   -- Das Ganze wiederholen
19: LOAD 3   -- Teilerzählerergebnis laden
20: END     --
```