

# Fakultät

*(Fakultät)*

**Stichwörter:** Mehr-Adress-Befehl-Assembler

---

## Fakultät

Erstelle ein Assemblerprogramm, dass als Eingabe über eine Speicherzelle die Fakultät dieser Zahl berechnet. Dabei soll das Programm die Rekursionsvorschrift:

$$\text{fak}(n) = n \cdot \text{fak}(n - 1)$$

benutzen.

Das Ergebnis der Berechnung soll am Ende in *R5* liegen.

```

-- Fakultaet

-- Erstelle ein Assemblerprogramm, dass als Eingabe ueber eine Speicherzelle die Fa-
-- kultaet dieser Zahl berechnet. Dabei soll das Programm die Rekursionsvorschrift:
-- fak(n) = n * fak(n - 1)
-- benutzen.
-- Das Ergebnis der Berechnung soll am Ende in R5 liegen.

-- public static int fak(int n) {
--     if (n <= 1) {
--         return 1;
--     } else {
--         return n * fak(n - 1);
--     }
-- }

-- erg                R5

fakultaet:
SEG

                MOVE W I H'0000FFFF', SP
                JUMP einstieg

einstieg:
                MOVE W I -1, -!SP
                MOVE W n, -!SP
                -- fak(n);
                CALL fak
                ADD W I 4, SP
                MOVE W !SP+, R5
                JUMP abschluss

-- public static int fak(int n)
-- n -> R0
fak:
                PUSHR
                MOVE W 64+!SP, R0
                CMP W R0, I 1
                -- if (n <= 1) return 1;
                JLE klGleichEins

```

```

        MOVE W I -1, -!SP
        -- n - 1
        SUB W I 1, R0, -!SP
        CALL fak
        ADD W I 4, SP
        -- n * fak(n - 1)
        MULT W !SP+, R0
        JUMP rueckgabe

-- return 1;
klGleichEins:  MOVE W I 1, R0

rueckgabe:     MOVE W R0, 68+!SP
               POPR
               RET

abschluss:     HALT

-- int n = 7;
n:             DD W 7

-- Tests

-- n:          0 -- 1
-- n:          1 -- 1
-- n:          2 -- 2
-- n:          3 -- 6
-- n:          4 -- 24
-- n:          5 -- 120
-- n:          6 -- 720
-- n:          7 -- 5040
-- n:          8 -- 40320
-- n:          9 -- 362880
END

```

```

public class Fakultaet {

    public static int fak(int n) {
        if (n <= 1) {
            return 1;
        } else {
            return n * fak(n - 1);
        }
    }

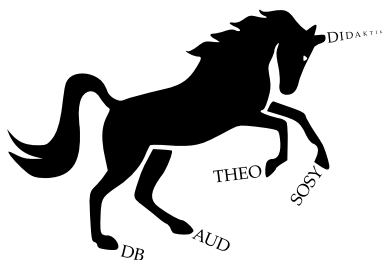
    public static void main(String[] args) {
        int n = 7;
        System.out.println(fak(n));

        System.out.println(fak(0)); // 1
        System.out.println(fak(1)); // 1
        System.out.println(fak(2)); // 2
        System.out.println(fak(3)); // 6
        System.out.println(fak(4)); // 24
    }
}

```

```
System.out.println(fak(5)); // 120
System.out.println(fak(6)); // 720
System.out.println(fak(7)); // 5040
System.out.println(fak(8)); // 40320
System.out.println(fak(9)); // 362880
System.out.println(fak(10)); // 3628800
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/tech\\_info/assembler/mehr\\_adress/Fakultaet.java](https://github.com/src/main/java/org/bschlangaul/aufgaben/tech_info/assembler/mehr_adress/Fakultaet.java)



## Die Bschlangaul-Sammlung

### Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der  $\text{\LaTeX}$ -Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: [https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/50\\_TECH/20\\_Mehr-Adress/Aufgabe\\_10-Fakultaet.tex](https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/50_TECH/20_Mehr-Adress/Aufgabe_10-Fakultaet.tex)