

## Übungsblatt 3 (Zusatzblatt)

Ziel dieses Übungsblattes ist es, dass Sie sich mit dem Prinzip der Rekursion vertraut machen.

### Aufgabe 4 – Quersumme bestimmen

Schreiben Sie eine Funktion **berechneQuersummeIT**, die die Quersumme einer Zahl iterativ bestimmt und am Ende als Rückgabewert zurückliefert.

Schreiben Sie anschließend eine Funktion **berechneQuersummeRK**, die die Quersumme rekursiv bestimmt.

Verwenden Sie folgende Main-Funktion zum Testen.

```
int main()
{

    int zahl;
    printf("Geben Sie eine Zahl ein: ");
    scanf("%d",&zahl);

    int quersumme = berechneQuersummeIT(zahl);
    printf("Quersumme von %d: %d\n\n", zahl, quersumme);

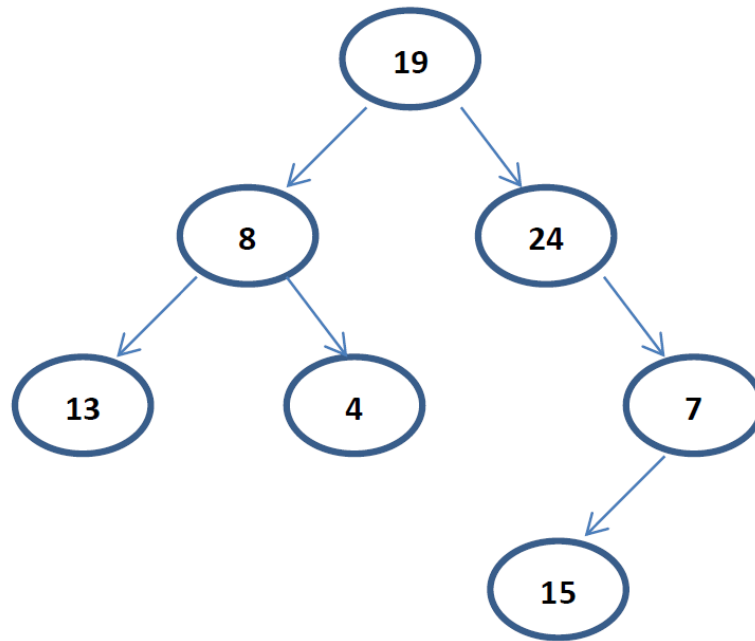
    quersumme = berechneQuersummeRK(zahl);
    printf("Quersumme von %d: %d\n\n", zahl, quersumme);

    return 0;
}
```

### Aufgabe 5 – Binärbaum durchsuchen

Ein Binärbaum ist eine Datenstruktur, die als spezielle Form eines Graphen an jedem Knoten höchstens zwei Kindknoten hat. Dadurch lässt sich eindeutig jeder Nachfolger in linkes und rechtes Kind einteilen.

Einen möglichen Binärbaum sehen Sie in folgender Abbildung:



Schreiben Sie eine rekursive Funktion **berechneSumme**. Die Funktion bekommt den Wurzelknoten des Baumes übergeben und bestimmt die Summe aller Zahlen in dem Binärbaum.

```
typedef struct BBaum BBaum;

typedef struct BBaum {
    BBaum* left;
    BBaum* right;
    int value;
} BBaum;

int main()
{
    BBaum b_15 = {NULL, NULL, 15};
    BBaum b_7 = {&b_15, NULL, 7};
    BBaum b_24 = {NULL, &b_7, 24};
    BBaum b_4 = {NULL, NULL, 4};
    BBaum b_13 = {NULL, NULL, 13};
    BBaum b_8 = {&b_13, &b_4, 8};
    BBaum root = {&b_8, &b_24, 19};

    int summe = berechneSumme(&root);

    printf("Die Summe beträgt: %d", summe);

    return 0;
}
```