

6. Übung

1. Vertiefung zu Strukturen und Operatoren

Geben Sie die Ausgabe für das unten dargestellte C-Programm an!

```
struct elem { int artnr;
              char bezeichnung[20];
              int lagerbestand; };

int main()
{
    struct elem lager[100] = {{21,"Winkeleisen",131},
                              {22,"Eisenbolzen",20},
                              {23,"Schraube",1355},
                              {32,"Mutter",1400},
                              {31,"Splint",410},
                              {0,"",0}};

    int i=0;
    int sum=0;
    int n=0;

    while (lager[i].artnr!=0)
    {
        sum += lager[++i].lagerbestand; /* M */
    }
    printf("Summe=%d\n", sum);

    i=0;
    while(lager[i++].artnr!=0) n++;
    printf("Anzahl=%d\n", n);

    return 0;
}
```

Wie würde sich die Ausgabe ändern, wenn die mit dem Kommentar M markierte Zeile lauten würde: `sum += lager[i++].lagerbestand;`

2. Datenstrukturen und Zugriffsoperationen

Zwei unterschiedliche lineare Datenstrukturen sind der Stapel und die Warteschlange.

Stapel (engl. Stack): Einzelne Elemente können mit der Zugriffsoperation PUSH nacheinander zugefügt werden. Die Elemente können mit der Operation POP in entgegengesetzter Reihenfolge ausgelesen werden. Mit dem Auslesen eines Elements wird das Element gleichzeitig aus dem Stapel entfernt. Bei POP wird immer das letzte zugefügte Element ausgelesen, sofern es existiert. Ist der Stack leer, so wird das in geeigneter Weise signalisiert und kein Element ausgelesen.

Warteschlange (engl. Queue): Einzelne Elemente werden mit der Operation ENQUEUE in die Warteschlange eingefügt. Die Elemente können mit der Operation DEQUEUE aus der Warteschlange entnommen werden. Die Entnahmereihenfolge entspricht der Reihenfolge des Zufügens.

Aufgaben:

- a) Erarbeiten Sie sich die Datentypen, Variablen und Zugriffsfunktionen für einen Stapelspeicher! Der Stapel soll durch ein Array realisiert werden, dessen Größe vor der Benutzung zur Laufzeit des Programms festgelegt werden soll.
- b) Realisieren Sie eine Warteschlange, ebenfalls basierend auf einem Array, dessen Größe vor der Benutzung zur Laufzeit des Programms festgelegt werden soll!

Zusatzaufgabe:

Realisieren Sie Stapel und Warteschlange mittels einer linearen Liste!