

Aufgabe 1 (Fehlersuche).**(10 Punkte)**

Finden Sie möglichst viele Fehler und geben bitte an ob es sich dabei um einen logischen Fehler handelt oder der Compiler eine Warnung oder Fehlermeldung ausgibt.

Bitte kurze Beschreibung mit Zeilennummer.

```
#include istream
use namespace "std";

/**/
    In diesem Programm haben sich leider einige Fehler eingeschlichen.
    Beheben Sie diese, damit das Programm korrekt ausgeführt werden kann.
    /**/
const int MAX 10;
void main() ;
{
    vector<string> text;
    for (i=0, i < MAX, i++) {
        text.add(to_string(i));
    }
    for (int el ; text) {
        cout << el << endl;
    }
    return ;
}
```

Aufgabe 2 (Wissensfragen).**(15 Punkte)**

Erklären Sie bitte mit einer kurzen und präzisen Antwort:

- Wann darf der Returnwert einer Funktion eine Referenz sein und wann nicht?
- Wozu dient der Konstruktor und wann wird er aufgerufen?
- Wozu dient der Destruktor und wann wird er aufgerufen?
- Was ist eine abstrakte Klasse?
- Was ist eine virtuelle Methode?
- Was ist eine Member-Variable?

Aufgabe 3 (Entwurf einer Klasse).**(25 Punkte)**

Die Klasse `bruch` soll einen Bruch aus ganzen Zahlen darstellen. Zähler und Nenner sollen nicht direkt zugänglich sein. Sie werden im Konstruktor gesetzt und können dann nur noch über die Methoden `zaehler()` und `nenner()` gelesen werden. Der Konstruktor soll wahlweise mit zwei, einem oder gar keinem Argument aufrufbar sein.

Objekte der Klasse `bruch` sollen wir folgt anwendbar sein:

```

// Hier soll die Lösung der Aufgaben a - d mit der
// Implementierung der Klasse Vektor stehen
int main()
{
    bruch a(2, 3), b(2, 4);
    b.kuerzen();
    bruch c = a + b;
    double l = a.toDouble();

    cout << "a = " << a << endl;
    return EXIT_SUCCESS;
}

```

- a. Schreiben Sie die Klassendefinition der Klasse **bruch**. Berücksichtigen Sie hierbei auch die Methoden, die in den nachfolgenden Aufgaben gefordert werden.
- b. Schreiben Sie die Implementierung für den Konstruktor.
- c. Schreiben Sie die Implementierung für die beiden Methoden **zaehler()** und **nenner()** als inline Methoden innerhalb der Klassendefinition.
- d. Implementieren Sie die Methode **inverse()**, die einen Bruch (den Kehrwert) als Returnwert hat.
- e. Implementieren Sie die Methode **kuerzen()**.
- f. Überladen Sie den Operator **+**, sodass zwei Objekte der Klasse Bruch addiert werden können.
- g. Überladen Sie den Operator **<<** zur Ausgabe eines Bruchs auf ein **ostream** Objekt, wie z.B. **cout**.