

<b>M411 Prüfung 1</b>	
<b>Datum</b>	<b>04.11.2021</b>
<b>Zeit</b>	<b>60 min</b>
<b>Name</b>	
<b>Max. Punktzahl</b>	<b>48P</b>
<b>Punktzahl</b>	
<b>Note</b>	

**Teil 1 Schriftliche Aufgaben (Total 28P)****Aufgabe 1 Fehlerkorrektur****6P**

Das folgende Programm soll das Volumen eines Quaders berechnen und ausgeben. Drei Fehler haben sich in das Programm eingeschlichen. Korrigieren Sie die Fehler im Code.

```
#include <stdio.h>

int main()
{
    CalcVolumenQuader(10, 10, 12);
    printf("Volumen: %f\n", volumen);

    return 0;
}

float CalcVolumenQuader(int l, int b, int h) {
    float volumen = l * b * h;
}
```

**Aufgabe 2 Parameterübergabe per Call By Value****8P**

- 1) Analysieren Sie folgendes Programm und ergänzen Sie die Konsolenausgabe. 3P

```
#include <stdio.h>

void Verdoppeln(int zahl);

int main()
{
    int zahl = 5;
    printf("Das Programm soll die Zahl %d verdoppeln:", zahl);

    Verdoppeln(zahl);

    printf("Am Ende des Programms: zahl = %d\n", zahl);
}

void Verdoppeln(int zahl) {
    zahl *= 2;
    printf("Innerhalb Funktion Verdoppeln() : zahl=%d\n", zahl);
    return;
}
```

Konsolenausgabe:  
Das Programm soll die Zahl \_\_\_\_ verdoppeln:  
Innerhalb der Funktion Verdoppeln() : zahl= \_\_\_\_  
Am Ende des Programms: zahl = \_\_\_\_

- 2) Wurde die Variable *zahl* nach dem Funktionsaufruf verdoppelt? Begründen Sie Ihre Antwort. 2P
- 3) Verbessern Sie das Programm, damit es die Variable *zahl* verdoppelt. Machen Sie die Änderungen direkt im Code. 3P

**Aufgabe 3 Structure Chart****14P**

Sie sind beauftragt ein Programm zu entwerfen, welches die Höhe einer Masse, welche senkrecht nach oben geworfen wird, berechnet. Die Formel dazu lautet:

$$h_m = \frac{v_o^2}{2g}$$

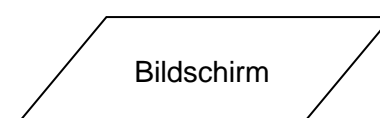
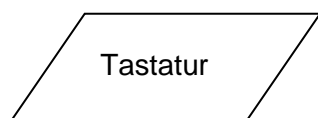
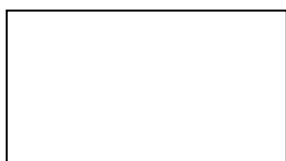
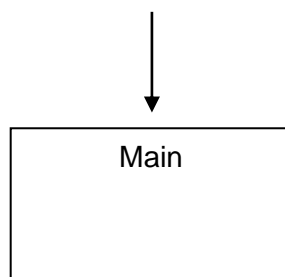
$h_m$  = maximale Steighöhe

$v_o$  = Anfangsgeschwindigkeit senkrechter Wurf nach oben

$g$  = Fallbeschleunigung 9,807 m/s<sup>2</sup>

Das Programm besteht aus drei Schritten: Eingabe der Anfangsgeschwindigkeit  $v_o$ , Berechnung der maximalen Steighöhe  $h_m$  und Ausgabe des Ergebnisses.

Ergänzen Sie das folgende Structure Chart mit Funktionsaufrufe, Datenflüsse und Zugriff auf Tastatur bzw. Bildschirm. Es soll keine Bibliotheksfunktion verwendet werden und geben Sie die Ein-/Ausgabe direkt auf die Tastatur bzw. Bildschirm:



**Teil 2 Programmieraufgaben (Total 20P)****Aufgabe 4 Bruchzahl****10P**

In einer Anwendung werden Brüche verwendet. Um die Eingabe zu vereinfachen, werden die Brüche in codierter Form als ganze positive Zahl eingegeben. Diese ist wie folgt aufgebaut:

zzzzznnn

zzzzz    Zähler mit maximal 5 Stellen

nnn      Nenner mit immer 3 Stellen

z.B.    3004 entspricht dann  $\frac{3}{4}$       Quotient = 0.75

432130 entspricht dann 432/130      Quotient = 3.3231      etc.

- Es ist eine Funktion zu schreiben, welche aus einer solchen codierten Bruchzahl den Quotient zurückgibt. Falls der Nenner Null ist, soll -1 retourniert werden.
- Erstellen Sie auch ein Hauptprogramm (*main*), welches diese Funktion testet. Das Hauptprogramm liest eine codierte Bruchzahl von der Tastatur ein und gibt den berechneten Quotient aus.

**Aufgabe 5 Binärdarstellung****10P**

Ein Algorithmus zur Umwandlung einer positiven Dezimalzahl in eine Dualzahl:

Solange zahl ungleich 0 führe folgende Schritte durch:

    zahl = zahl mod 2

    zahl = zahl / 2

Gib die errechneten Reste in umgekehrter Reihenfolge (von unten nach oben) aus.

Beispiel:

    12 : 2 = 6    Rest 0    ^

    6 : 2 = 3    Rest 0    |

    3 : 2 = 1    Rest 1    |

    1 : 2 = 0    Rest 1    |

Die Dualzahl zu 12 ist somit 1100.

- Schreiben Sie eine rekursive Funktion, welche eine positive dezimale Ganzzahl in eine binäre Zahl umwandelt und die binären Ziffern in der richtigen Reihenfolge ausgibt.
- Erstellen Sie auch ein Hauptprogramm (*main*), welches diese Funktion testet. Das Hauptprogramm liest eine positive Dezimalzahl von der Tastatur ein und gibt die Dualzahl aus.