

## BMI

Schreiben Sie ein C-Programm, das den Body-Mass-Index (BMI) berechnet sowie eine Bewertung des errechneten BMI ausgibt.

**Einleitung** Der BMI ist eine (positive) Maßzahl für die Bewertung des Körpergewichts ( $m$ ) eines Menschen in Relation zu seiner Körpergröße ( $l$ ) und errechnet sich als Gewicht in Kilogramm durch Quadrat der Körpergröße in Metern:

$$BMI = m/l^2$$

In unserem Beispiel soll die Berechnung des BMI nur durchgeführt werden für Körpergewichte ab 40kg bzw. Körpergrößen ab 160cm.

Tabelle 1: Adipositas-Kategorien

BMI [ $kg/m^2$ ]	Kategorien-Bezeichnung	Kategorien-Nummer
< 16,00	stark untergewichtig	10
16,00–16,99	mäßig untergewichtig	11
17,00–18,49	leicht untergewichtig	12
18,50–24,99	normalgewichtig	20
25,00–29,99	präadipös	30
30,00–34,99	adipös Grad I	40
35,00–39,99	adipös Grad II	41
>= 40,00	adipös Grad III	42

**Aufgabe** Implementieren Sie hierfür die folgenden Funktionen

1. `double bmi_calc(double height, double weight)`

berechnet den BMI für gegebene `height` (Körpergröße) und `weight` (Körpergewicht). Die Funktion geht von den oben genannten Einheiten Meter für Größe bzw. Kilogramm für Gewicht aus.

Für ungültige Argumente gibt die Funktion -1 zurück.

2. `int bmi_class(double bmi)`

berechnet die Adipositas-Kategorie für den BMI `bmi`.

Retourniert die entsprechende Adipositas-Kategorien-Nummer (siehe Tabelle).

Im Falle ungültiger Argumente (negativer BMI) wird -1 zurückgegeben.

3. `void bmi_message(int bmi_class)`

gibt die Bezeichnung der Kategorie `bmi_class` aus (siehe Tabelle)

Im Falle ungültiger Argumente wird die Nachricht *ungueltig* ausgegeben.

Die `main`-Funktion fragt nach Körpergröße und –gewicht und gibt dann die eingelesenen Werte, (auf zwei Kommastellen genau), den BMI sowie den Namen der BMI-Kategorie aus. (siehe Beispiel).

**Beispiele:** (Text in rot = Benutzereingabe)

```
Koerpergewicht [kg]: 121.5
Koerpergroesse [m]: 1.96
```

```
m=121.50kg l=1.96m -> BMI=31.63 (adipoes Grad I)
```

```
Koerpergewicht [kg]: 25  
Koerpergroesse [m]: 1.16  
m=25.00kg l=1.16m -> BMI=-1.00 (ungueltig)
```

### Hinweise:

- ersetzen Sie in der Ausgabe  $\beta$  durch *ss*,  $\delta$  durch *oe*,  $\ddot{a}$  durch *ae* usw.
- verwenden Sie zum Testen die angegebene `main`-Funktion (auch als File zum Download in moodle)

```
int main() {  
    double weight, height, bmi;  
    printf("Koerpergewicht [kg]: ");  
    scanf("%lf", &weight);  
    printf("Koerpergroesse [m]: ");  
    scanf("%lf", &height);  
    printf("m=%.2fkg l=%.2fm -> BMI=%.2f (",  
        weight, height, bmi=bmi_calc(height, weight));  
    bmi_message(bmi_class(bmi));  
    printf(")\n");  
    return 0;  
}
```

### Abgabe:

Laden Sie in moodle hoch und benennen Sie die Datei `main.c` in moodle um auf `BMI_id.c`, wobei `id` ihre Technikum-Kennung ist.

**Beispiel:** `id` lautet `mr31b090`. filename lautet: `BMI_mr31b090.c`

Tabelle 2: BMI Komplexität d. Lösung

Method name	Complexity	Category	SLoC
bmi_class	10	simple, without much risk	23
bmi_message	10	simple, without much risk	19
bmi_calc	5	simple, without much risk	8
main	1	simple, without much risk	11