

Programmierung 1

Übungsblatt Woche 5 - 19. - 25. November 2025

Testat!

1. BMI Berechnung (blatt5_1.c)

Implementieren Sie mit Hilfe von einer oder mehreren Funktion/en Programm zur Berechnung des BMIs, ähnlich dem in der Vorlesung vorgestellten.

Erweitern Sie ihr Programm, d.h. die main-Funktion, um eine Schleife, in der der Benutzer immer wieder gefragt wird, ob er einen weiteren BMI berechnen möchte. Recherchieren Sie online nach der Klassifizierung des BMIs und geben Sie zusätzlich zum errechneten Wert auch eine Klassifikation aus, indem Sie eine Funktion schreiben, der der BMI-Wert übergeben wird und die dann einen entsprechenden Text ausgibt.

2. Runden (blatt5_2.c)

Schreiben Sie eine Funktion `int rundeSumme (int a, int b, int c)`, die die Summe der auf Zehner gerundeten Werte von a, b und c zurückgibt, d.h. erst wird jede Zahl einzeln auf-/abgerundet, danach wird die Summe gebildet. Schreiben Sie hierzu eine Hilfsfunktion `int runde10 (int n)`, die den formalen Parameter auf den nächsten Zehner aufrundet (ab Endziffer 5) oder abrundet (bis Endziffer 4), sodass Code nicht unnötig wiederholt wird.

3. Teiler (blatt5_3.c)

Implementieren Sie eine Funktion `int ggT (int a, int b)` zur Bestimmung des größten gemeinsamen Teilers (ggT) mit Hilfe des euklidischen Algorithmus. Dieser funktioniert wie folgt: Gegeben seien zwei natürliche Zahlen a und b. Man zieht nun solange die kleinere von der größeren Zahl ab, bis beide Zahlen gleich sind. Das Resultat $ggT(a, b)$ ist der größte gemeinsame Teiler von a und b. **Implementieren Sie genau diese Version des euklidischen Algorithmus! Andere Versionen (auch effizientere) geben Punktabzug.**

4. Vielfaches (blatt5_4.c)

Gegeben seien wieder zwei natürliche Zahlen a und b. Gesucht ist diesmal das kleinste gemeinsame Vielfache (kgV), also die kleinste Zahl, die sowohl a als auch b als Teiler hat. Im Gegensatz zum ggT gibt es für den kgV leider keine so einfache Berechnungsvorschrift wie den euklidischen Algorithmus. Da wir seit Aufgabe 3) jedoch schon den größten gemeinsamen Teiler $ggT(a, b)$ ausrechnen können, können wir aber zur Berechnung des kgV die folgende Beziehung nutzen: $a \cdot b = ggT(a, b) \cdot kgV(a, b)$.

Zu implementieren ist nun also die Funktion `int kgV (int a, int b)`, welche zur Bestimmung des kgV zweier Zahlen a und b deren ggT verwendet.

5. Mauer (blatt5_5.c)

Implementieren Sie eine Funktion `bool mauern (int klein, int gross, int ziel)`. Wir wollen eine Mauer der Länge `ziel` mauern. Dazu haben wir Anzahl `klein` Steine der Länge 1m und `gross` Steine der Länge 5m. Die Funktion soll `true` zurückliefern, wenn das Ziel **exakt** erreicht werden kann, andernfalls `false`.

Beispiele:

- `mauern(3, 1, 8) → true`
- `mauern(3, 1, 9) → false` (wir kommen insgesamt nur auf 8m)
- `mauern(2, 3, 13) → false` (wir können 13m nicht exakt erreichen)
- `mauern(3, 2, 10) → true`

6. Hochladen und Vorstellen

Stellen Sie Ihre Lösung in Ihrer Übungsgruppe vor. Laden Sie bis spätestens Dienstag, den 25. November 2025, 23:59 Uhr, die Dateien `blatt5_1.c`, `blatt5_2.c`, `blatt5_3.c`, `blatt5_4.c` und `blatt5_5.c` im eLearning hoch. Überprüfen Sie, dass die erreichten Punkte auch als Bewertung für Ihre Abgabe eingetragen werden. Melden Sie sich ansonsten zeitnah, damit dies nachgeholt werden kann.