

Übungsaufgaben Programmierung

1. Schreiben Sie ein Programm *Kreis*, das einen Kreisradius entgegen nimmt und daraus Fläche und Umfang ausrechnet. Sorgen Sie dafür, dass die Ausgaben verständlich und die Eingaben vernünftig sind (geeignet überprüfen).
2. Das Programm *Registrierkasse* nimmt den zu zahlenden Betrag und den vom Kunden gegebenen Betrag. Ausgegeben werden Nettobetrag, Mehrwertsteuer, Bruttobetrag und Wechselgeld.
3. Das Programm *BMI* berechnet aus der Masse m in kg und der Größe h in m den Body-Mass-Index nach der Formel $bmi = \frac{m}{h^2}$ und gibt dazu Kommentare ab. Ein Wert von weniger als 20 deutet auf Untergewicht hin, ein Wert über 25 auf Übergewicht.
4. Der Ostersonntag ist der Sonntag nach dem ersten Frühlingsvollmond. Das Programm *Ostern* nimmt eine Jahreszahl und berechnet daraus den Ostersonntag-Termin. Gut, dass die genannte Regel von C. F. Gauß mathematisiert wurde. Die Berechnung ist trotzdem noch ein ziemlicher Zirkus (als Programmierübung aber genau das richtige).

Im folgenden sei j das Jahr, also z.B. $j = 2006$. Man setzt nun

$$\begin{array}{llll}
 m = 23 & n = 3 & \text{falls } j/100 = 17 & \text{(wobei '/' der Divisionsoperator für ganze Zahlen ist)} \\
 m = 23 & n = 4 & \text{falls } j/100 = 18 & \\
 m = 24 & n = 5 & \text{falls } j/100 = 19 & \text{und berechnet die Werte} \\
 m = 24 & n = 5 & \text{falls } j/100 = 20 & \\
 m = 24 & n = 6 & \text{falls } j/100 = 21 &
 \end{array}
 \quad
 \begin{array}{l}
 a = j\%19 \\
 b = j\%4 \\
 c = j\%7 \\
 d = (19a + m)\%30 \\
 e = (2b + 4c + 6d + n)\%7
 \end{array}$$

dann fällt Ostern auf den $(22 + d + e)$ -ten März oder den $(d + e - 9)$ -ten April. Dieses Ergebnis ist jedoch in zwei Fällen falsch: Falls die Formel den 26. April liefert, ist Ostern am 19. April und falls die Formel den 25. April liefert und $a \geq 11$, ist Ostern auch am 19. April.

5. Die quadratische Gleichung $ax^2 + bx + c = 0$ hat die Diskriminante $D = b^2 - 4ac$. Wenn $D < 0$, so hat die Gleichung keine Lösung, bei $D = 0$ hat sie die eine Lösung $x = -\frac{b}{2a}$ und bei $D > 0$ die beiden Lösungen $x_1 = \frac{-b + \sqrt{D}}{2a}$ und $x_2 = \frac{-b - \sqrt{D}}{2a}$. Das Programm *QuadGl* nimmt die drei Argumente a , b und c und gibt dann alles aus, was zu sagen ist.

Das Programm darf auch noch erweitert werden auf den Fall $a = 0$. Dann handelt es sich nicht mehr um eine quadratische sondern nur noch eine lineare Gleichung mit der einen Lösung $x = -\frac{c}{b}$.

6. Das Programm *Dreieck* nimmt drei Seitenlängen der Größe nach sortiert entgegen (die kürzeste Seite zuerst) und macht dann eine Aussage zu dem Dreieck (spitz-, recht- oder stumpfwinklig, gleichschenkelig, gleichseitig). Als Erweiterung können auch die Fläche, der Umfang und die Winkel berechnet werden.
7. Ein lineares Gleichungssystem mit zwei (oder auch mehreren) Gleichungen und ebenso vielen Unbekannten wird gelöst. Die Lösungen werden vom Programm *GLSys* ausgegeben, wenn sie existieren. Wenn nicht, kommt ein entsprechender Kommentar.
8. Das Programm *Formelsammlung* bietet dem Benutzer mehrere Berechnungsformeln (Kreis, Kugel, Kegel, Trapez, usw.) an und lässt ihn auswählen.
9. Bei *Mastermind* denkt sich der Computer eine vierstellige Zahl (aus lauter verschiedenen Ziffern). Der Spieler muss diese durch logisches Denken herausfinden. Er rät mehrmals, welche vierstellige Zahl es sein könnte und bekommt jedesmal vom Computer Tips dazu nach folgender Regel:

Für jede Ziffer, die schon an der richtigen Stelle ist gibt es ein Plus. Für jede richtige Ziffer an der falschen Stelle ein Minus und für jede falsche Ziffer gar nichts. Diese Tips gibt der Rechner so lange, bis der Spieler die zu erratende Ziffernfolge gefunden hat.

Jede Zeile im Beispiel zeigt links, was der Mensch geraten hat und rechts den Tip des Computers. In der letzten Zeile erkennt man, dass die zu erratende Zahl 5328 war.

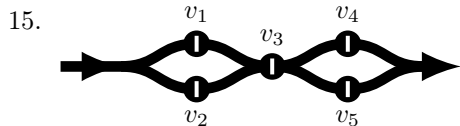
| | |
|------|-------|
| 1234 | -- |
| 2138 | +-- |
| 5382 | +++-- |
| 5328 | ++++ |

10. Bei *MastermindInvers* sind die Rollen von Mensch und Computer vertauscht. Der Mensch denkt sich eine Zahlenfolge und der Computer versucht, diese in möglichst wenigen Schritten herauszufinden. Hier muss der Mensch nach jedem Rateschritt Tips nach dem oben genannten Muster geben.
11. Die Mitarbeiter einer Firma erhalten eine Lohnerhöhung um 3%, mindestens jedoch 200 Euro. Nach Eingabe des alten Lohns soll die Erhöhung und der neue Lohn ausgegeben werden.

- Eine Bank führt die ersten 10 Buchungen auf einem Girokonto kostenlos durch, die nächsten 10 kosten 0,50 Euro und jede weitere 0,40 Euro.
- Drei eingegebene Zahlen sollen der Größe nach sortiert werden.

- Das Wetter in Gießen ist einfach vorherzusagen. Nach Eingabe von Luftdruck und Windrichtung soll die Vorhersage ausgegeben werden.

| Luftdruck | Wind aus | | |
|------------|--------------|-------|--------------|
| | O/N | S | W |
| < 1000 hPa | veränderlich | Regen | Regen |
| ≥ 1000 hPa | Sonne | Sonne | veränderlich |



- Kühlflüssigkeit kann über fünf Ventile von links nach rechts fließen, wenn diese geeignet geöffnet sind. Das Programm *Ventile* nimmt als Argumente fünf Buchstaben (o = offen, z = zu) und gibt aus, ob die Kühlflüssigkeit durch kommt.

- In einer Klassenarbeit wird ab 20% der erreichbaren Punkte die Note 5 gegeben, ab 40% die Note 4, ab 55% die Note 3, ab 70% die Note 2 und ab 85% die Note 1. Das Programm *Noten* nimmt als Argumente die Anzahl der erreichten Punkte und die Höchstpunktzahl und berechnet daraus die Note.
- Ein Dreieck ist durch die Koordinaten der Ecken A , B und C gegeben. Ein vierter Punkt D kann innerhalb, außerhalb oder auf dem Rand des Dreiecks liegen. Das Programm *DreieckCheck* nimmt als Argumente die Koordinaten der vier Punkte und macht eine diesbezügliche Aussage.
- Sherlock Holmes konnte aus den Eigenschaften von Pfeifenasche auf die Tabakmarke schließen. Das Programm *Tabak* versucht dies ebenfalls nach folgender Tabelle:

| Marke | Beschaffenheit | Farbe | Teilchen | Nikotingehalt |
|--------------|----------------|--------|----------|---------------|
| Mixture | blättrig | hell | nein | gering |
| BorkumRif | gepresst | dunkel | nein | gering |
| ThreeNuns | locker | dunkel | ja | hoch |
| PrinceArthur | blättrig | hell | nein | mittel |
| NavyCut | gepresst | hell | ja | mittel |
| Country | blättrig | hell | ja | gering |
| RollCake | gepresst | dunkel | ja | hoch |

- Das Programm *Quadrate* gibt in der ersten Spalte die Zahlen 20 bis 40 und in der zweiten Spalte die zugehörigen Quadratzahlen aus. Ändern Sie danach das Programm so ab, dass der Zahlenbereich als Argumente übergeben werden kann.
- Gegeben ist ein Anfangskapital K_0 und ein Zinssatz p . Es soll die Kapitalentwicklung in den nächsten 5 Jahren ausgegeben werden. Außerdem soll ermittelt werden, wann genau sich das Kapital verdoppelt hat.
- Ein Darlehen D wird mit dem Zinssatz p verzinst. (a) Jedes Jahr wird ein fester Betrag B zurück gezahlt. (b) Jedes Jahr werden ein fester Betrag B und die Zinsen Z zurück gezahlt. Das Programm *Tilgung* soll bis zum Ende der Laufzeit tabellarisch auflisten, wie hoch die Schulden noch sind und wie viel zurück gezahlt wurde.
- Das Programm *AlleTeiler* nimmt eine ganze positive Zahl und listet alle Teiler auf.
- Das Programm *GGT* nimmt zwei ganze Zahlen und berechnet deren größten gemeinsamen Teiler.
- Das Programm *PrimTest* nimmt eine ganze Zahl und stellt fest, ob diese prim ist.
- Das Programm *XTabelle* erzeugt eine Tabelle von Multiplikationsergebnissen.
- Das Programm *EinmalEins* stellt dem Benutzer Multiplikationsaufgaben. Die Anzahl der gestellten Aufgaben und der Zahlenbereich sind frei wählbar. Das Programm gibt nach jeder Antwort aus, ob das Ergebnis richtig oder falsch war. Bei falschen Ergebnissen wird das richtige ebenfalls genannt. Am Ende wird der Prozentsatz der richtig gelösten Aufgaben ausgegeben.
- Das Programm *EinmalEins2* tut das gleiche wie *EinmalEins*, merkt sich aber zusätzlich die falsch gelösten Aufgaben und stellt diese zwischendurch immer wieder, bis der Benutzer sie irgendwann richtig beantwortet hat. Dann wird die Ausgabe aus der Merkleiste wieder entfernt.

28. Das Programm *Rausgeben* nimmt einen Geldbetrag (mit höchstens zwei Nachkommastellen) und listet die kleinst mögliche Anzahl von Scheinen und Münzen auf, die diesen Betrag ergeben. Beispiel: 219,58 Euro sind $1 \times 200 + 1 \times 10 + 1 \times 5 + 2 \times 2 + 1 \times 1 + 1 \times 0,50 + 1 \times 0,05 + 1 \times 0,02 + 1 \times 0,01$.
29. Das Programm *RoemDez* nimmt eine römische Zahl und wandelt sie in eine Dezimalzahl um. Das Programm *DezRoem* tut das umgekehrte. Um dies programmieren zu können, sollte man die Regeln für römische Zahlen genau kennen. Diese Regeln haben sich im Lauf der Jahrhunderte geändert, aber man hält sich üblicherweise an folgendes:
- (a) Es gibt die Zeichen $I = 1$, $V = 5$, $X = 10$, $L = 50$, $C = 100$, $D = 500$ und $M = 1000$.
 - (b) Wird einem Zeichen ein kleineres nachgestellt, so wird dessen Wert addiert.
 - (c) Wird einem Zeichen ein kleineres vorangestellt, so wird dessen Wert subtrahiert. Es darf jedoch nur ein kleineres Zeichen vorgestellt werden.
 - (d) Die Zeichen I , X , und C dürfen höchstens dreimal hintereinander auftreten.
 - (e) Die Zeichen V , L und D werden nicht wiederholt und nicht vorgestellt.

Damit sind $IM = CMXCIX$ zwei gültige Darstellungen der Zahl 999. (Manche Gelehrte sagen, IM sei ungültig, aber darum kümmern wir uns selbstbewusst nicht.)