

Aufgabe 1a

Entwickelt ein Pascal Programm, das folgendes leistet:
Unmittelbar nach dem Einlesen wird der zugehörige ASCII-Code
ausgegeben.

Freiwillig: Es sind fortlaufend Zeichen einzulesen. (Schleife)

Aufgabe 1b

Entwickelt nun ein Programm, das die umgekehrte Funktion hat:
Zu einem eingelesenen ASCII-Code soll das zugehörige Zeichen
ausgegeben werden. Hinweise:

- Macht Euch Gedanken über den verwendeten Datentypen!
- Eine Fehlerbehandlung für Eingaben außerhalb des Wertebereichs ist hier (noch) nicht notwendig.

Aufgabe 2a

Entwickelt ein Programm, das zu drei eingegebenen Ganzzahl-
Werten i_n sowohl das arithmetische als auch das geometrische
Mittel auf vier Nachkommastellen gerundet ausgibt.

Hinweise:

- Arithmetisches Mittel

$$AM = \frac{1}{N} \sum_{n=1}^N i_n$$

- Geometrisches Mittel

$$GM = \sqrt[N]{i_1 \cdot \dots \cdot i_N} = \sqrt[N]{\prod_{n=1}^N i_n}$$

Der Programmlauf könnte folgendermaßen aussehen:

```
Messwert 1? 2
Messwert 2? 4
Messwert 3? 7
```

```
Das arithmetische Mittel ist 4.3333
Das geometrische Mittel ist 3.8259
```

Hinweise :

Eine Funktion X^Y gibt es in Pascal nicht. Die Funktion *sqrt*
berechnet nur die Quadratwurzel.
Als nützlich könnten sich die Funktionen *ln(x)* (natürlicher
Logarithmus) und *exp(x)* (Exponentialfunktion) erweisen.

Aufgabe 2b

Es sollen nun Real- statt Ganzzahlwerten eingegeben werden.
Rundet die eingegebenen Werte auf zwei Nachkommastellen
und rechnet einmal mit den "echten", ein weiteres Mal mit den
gerundeten Werten weiter.

Der Programmlauf könnte folgendermaßen aussehen:

```
Messwert 1? 2.355
(2.355->2.36)
Messwert 2? 4.029
(4.029->4.03)
Messwert 3? 7
(7->7.00)
```

arithmetisches Mittel:

4.4613 mit exakten, 4.4633 mit gerundeten Werten

geometrisches Mittel:

4.0498 mit exakten, 4.0530 mit gerundeten Werten

Hinweis: Mit „Wert:n:m“ läßt sich die Ausgabe formatieren. Wobei
n= Anzahl aller Stellen und m=Anzahl der Nachkommastellen ist.
Zum Runden verwendet Trunc() und Frac().

Aufgabe 3

Ziel dieser Aufgabe ist es, zu einem eingegebenen Geldbetrag (höchstens DM 99.99) die Stückelung auszugeben, die die kleinste Anzahl von Münzen und Scheinen beansprucht.

Zum Beispiel so:

```
C:\WORK>geld
Betrag ? 55.43
1x 50.00 DM
1x 5.00 DM
4x 0.10 DM
1x 0.02 DM
1x 0.01 DM
```

Aufgabe 4

Programmiert einen "Mini-Taschenrechner", der die vier Grundrechenarten beherrscht. Nach Eingabe von zwei Ganzzahloperanden und des Operators soll das Endergebnis ausgerechnet werden.

Ist es möglich, dass Euer Programm sich bei bestimmten Werten "verrechnet"? Probiert einmal byte als Variablentyp und gebt im Programmablauf folgende Werte ein:

```
C:\WORK>rechn
Op.1 ? 290
Op.2 ? 10
Operator <+, -, *, /> ? -
```

```
290 - 10 = .....
```

Aufgabe 5

Der letzten Aufgabe soll folgende Situation zu Grunde liegen: Rund um ein rechteckiges Grundstück soll eine Hecke gepflanzt werden. Es wird lediglich der Parkplatz ausgespart, der halb so breit und 1/4 so lang ist wie das Grundstück (s. Skizze).

Zunächst muss ein Graben von 30 cm Breite und 40 cm Tiefe ausgehoben werden, danach werden die Setzlinge in einem Abstand von a cm gepflanzt. p Gärtner sind am Pflanzen beteiligt, von denen jeder 5 Meter Graben und Hecke pro Stunde fertigstellt.

Nach Eingabe von Breite x und Länge y des Grundstücks sowie den o.g. Variablen p und a soll folgendes berechnet werden:

- Das Gesamtvolumen des ausgehobenen Grabens
- Die benötigte Zeit
- Die Anzahl der benötigten Setzlinge

Hinweise:

- Benutzt, wann immer möglich, Konstanten!
- Seid bei den Berechnungen sehr genau – insbesondere an den Ecken

