

Beschreibung

Das Maximum bestimmt die größte Zahl aus einer Auswahl von vorliegenden Zahlen. Zur Ermittlung des Maximums gibt es in den gängigen Programmiersprachen eine Funktion. Wir wollen uns das aber in dieser Aufgabe selber schreiben.

Aufgabenstellung

Schreibe ein Programm, welches das Maximum von drei Integer-Variablen bestimmt und in der Kommandozeile ausgibt. Benutze bei den Vergleichen ausschließlich `if`- und `else`-Anweisungen sowie den `>`-Operator.

Testfälle

- Vorliegende Zahlen: 1, 2, 3 → Maximum davon: 3
- Vorliegende Zahlen: 42, 7, 13 → Maximum davon: 42
- Vorliegende Zahlen: -9, 4, 2 → Maximum davon: 4

Beschreibung

In dieser Aufgabe wollen wir die Summe von 24 aufeinanderfolgenden Zahlen berechnen. Da dies mit dem Taschenrechner ein sehr großer Tippaufwand wäre, stellt sich die Frage: Warum wollen wir das nicht einfach den Computer erledigen lassen?

Aufgabenstellung

Schreibe ein Programm, das mit einer for-Schleife die Summe der Zahlen von 3 bis 27 berechnet und das Ergebnis in der Konsole ausgibt.

Testfälle

- Das Ergebnis der Summe von 3 bis 27 ist 375.
- Das Ergebnis der Summe von 1 bis 100 ist 5050.

Beschreibung

Bei Fußballwettspielen werden die erzielten Wettpunkte nach folgenden Regeln ermittelt:

Exakter Tipp (3 Punkte):

- Sieg (z.B. Ergebnis: 3:2, Tipp: 3:2)
- Niederlage (z.B. Ergebnis: 0:1, Tipp: 0:1)
- Unentschieden (z.B. Ergebnis: 2:2, Tipp: 2:2)

Richtige Tendenz (1 Punkt):

- Sieg (z.B. Ergebnis: 3:2, Tipp: 2:1)
- Niederlage (z.B. Ergebnis: 0:1, Tipp: 1:2)
- Unentschieden (z.B. Ergebnis: 2:2, Tipp: 1:1)

Sonst 0 Punkte

Aufgabenstellung

Schreibe ein Programm, das für ein Fußballtippspiel die erzielten Punkte berechnet und das Ergebnis in der Konsole ausgibt. Dazu werden das getippte Ergebnis und das tatsächlich erzielte Ergebnis des Fußballspiels benötigt.

Testfälle

- | | |
|--|--|
| ▪ Ergebnis: 3:2 , Tipp: 3:2 → Punkte: 3 | ▪ Ergebnis: 2:2 , Tipp: 1:1 → Punkte: 1 |
| ▪ Ergebnis: 0:1 , Tipp: 0:1 → Punkte: 3 | ▪ Ergebnis: 3:2 , Tipp: 2:0 → Punkte: 0 |
| ▪ Ergebnis: 2:2 , Tipp: 2:2 → Punkte: 3 | ▪ Ergebnis: 0:1 , Tipp: 1:1 → Punkte: 0 |
| ▪ Ergebnis: 3:2 , Tipp: 2:1 → Punkte: 1 | ▪ Ergebnis: 2:2 , Tipp: 2:4 → Punkte: 0 |
| ▪ Ergebnis: 0:1 , Tipp: 1:2 → Punkte: 1 | |

Beschreibung

In dieser Aufgabe wollen wir alle vierstelligen PINs einer Bankkarte oder Passcodes erzeugen und in der Konsole ausgeben.

Aufgabenstellung

Schreibe ein Programm, das die oben beschriebene Aufgabe umsetzt. Setze diese mit einer for-Schleife um.

Testfall

- Ausgabe:

0000

0001

0002

0003

0004

...

9995

9996

9997

9998

9999

Alle PINs sind vierstellig!

Zahlen-Palindrom

Sonntag, 2. Dezember 2018 13:34

Themen

Mit dieser Aufgabe wollen wir Folgendes trainieren:

- Kontrollstrukturen
- Bedingte Anweisungen
- Wiederholungsanweisungen

Beschreibung

Ein Palindrom ist eine Zeichenkette, die von vorne und hinten gelesen dasselbe ergibt. Diese Eigenschaft wollen wir in dieser Aufgabe mit ganzen Zahlen testen.

Aufgabenstellung

Schreibe ein Programm, das testet, ob eine **ganze Zahl** ein Palindrom ist. Die zu testende Zahl darf dabei nicht in ein String-Objekt umgewandelt werden. Gebe das Ergebnis in der Konsole aus.

Testfälle

- 12321 ist ein Palindrom
- 12345 ist kein Palindrom
- 2123212 ist ein Palindrom
- 2123213 ist kein Palindrom

Algorithmische Tipps

Wenn du stockst und nicht weiter weißt, dann versuch mal Folgendes:

- Schreibe dir zunächst einen der Testfälle auf Papier und überlege, wie du an die entsprechenden Ziffern kommst. Kleiner Tipp: Per Division „schneidest“ du Stellen am Zahlenende weg. Mit dem Modulo-Operator „schneidest“ du Stellen am Zahlenanfang weg.
- Bestimme zunächst die Anzahl der Dezimalstellen. Dafür kannst du die Palindrom-Zahl so lange mit der Zahl 10 dividieren (entferne letzte Dezimalstelle), bis die Zahl 0 erreicht ist.
- Anschließend kannst du die Zahl Schritt für Schritt durchgehen, die vordere und hintere Dezimalstelle bestimmen und miteinander vergleichen. Je nachdem, welche Zahl du überprüfen willst, musst du dann zunächst vorne und hinten entsprechend Dezimalstellen abschneiden, bis nur noch die gewünschte Ziffer übrig bleibt. Zur Bestimmung der dafür passenden Zehnerpotenz kannst du die Funktion `pow(Zahl, Exponent)` verwenden. Diese Funktion berechnet die Potenz einer Zahl. Zum Testen solltest du am besten eine Nicht-Palindrom-Zahl verwenden. Damit kannst du überprüfen, welche Zahlen miteinander verglichen werden.