Lektion 3: Befehle mit Bedingungen	Übung 3: If…Then…Else…End und Teilbarkeit
In der dritten Übung der Lektion 3 geht es um eine weitere Form des Befehls **If** im TI Basic und um das Verstehen numerischer Algorithmen.	**Lernziele:** • Anwendung des Befehls **If…Then…Else…End.** • Testen einer Zahl auf Ganzzahligkeit. • Konzept **Algorithmus** verstehen.

Hinweis: Diese Übung soll den Begriff des **Algorithmus** verdeutlichen. Dahinter verbirgt sich ein Prinzip, das wichtig ist für erfolgreiches Programmieren. Allerdings kann dieses Prinzip hier nur in einer ersten Annäherung betrachtet werden. Entscheidend ist, dass man einen Plan haben muss, bevor man ein Programm schreibt, und für diesen Plan wiederum ist es wichtig, dass man die Programmierbefehle und ihre Anwendung kennt. Erst dann kann man das Programm überhaupt schreiben.
Die drei Formen des Befehls **If** werden in dieser Übung abgeschlossen mit der Untersuchung der letzten Form **If…Then…Else…End**. Wie immer ist die Syntax wichtig: jedes der vier Schlüsselwörter ist ein eigenständiger Befehl in einer eigenen Zeile und nur dem **If** folgt noch etwas, nämlich die Bedingung.

Über If…Then…Else…End

Bislang ging es um den Befehl **If…Then…End.** Allerdings gibt es Gelegenheiten, bei denen man zwei verschiedene Aktionen ausführen möchte, die von der Bedingung abhängen. Der Aufbau des neuen **If…Then…Else…End -** Befehls ist ähnlich dem bisherigen:

```
If  <Bedingung>
Then
   <Befehle wenn Bedingung wahr>
Else
  <Befehle wenn Bedingung falsch>
End
```

Hinweis:
Then, **Else** *und* **End** *stehen in einer eigenen Zeile.*
Es werden entweder *die Befehle der ersten Befehlsgruppe (wahr)* oder *die der zweiten (falsch) ausgeführt.*

Programmieren mit If…Then…Else…End

Es soll ein Programm geschrieben werden, mit dem man ermittelt, ob eine eingegebene natürliche Zahl eine „perfekte" Zahl ist, d.h. ob ihre Wurzel wieder eine natürliche Zahl ist. Der Programmcode ist rechts abgebildet.

```
PROGRAM:QUADRAT
:ClrHome
:Input "EINE ZAHL EINGEBEN",N
:If √(N)=ganzTeil(N)
:Then
:Disp "ES IST EINE PERFEKTE
ZAHL"
:Else
:Disp "ES IST KEINE PERFEKTE
ZAHL"
:End
```

Hinweis:
Für die Wurzel [√] wird das Symbol auf der Tastatur verwendet..
If, **Then**, **Else** *und* **End** *befinden sich alle im Menü* [PRGM] STRG.
Die Funktion **ganzTeil( )** *befindet sich im Menü* [MATH] NUM. *Die Funktion hat den ganzzahligen Teil einer Zahl als Ergebnis:*
ganzTeil(6.56) → 6 ganzTeil(9.999) → 9 ganzTeil(-2.3) → -2
Die Funktion ***ganzTeil(*** *sollte auf dem HOME-Bildschirm ausprobiert werden. Wie bei der Wurzelfunktion darf die schließende Klammer nicht vergessen werden.*

Hinweis: Man kann das Wort „**Es**“ im Befehl **Disp** ersetzen durch die eingegebene Zahl, muss dann aber statt **Disp Output(** verwenden.

Algorithmen

Das Programm ist ein gutes Beispiel für das, was man einen **Algorithmus** nennt. Ein **Algorithmus** ist eine Formel oder eine Folge von Anweisungen, mit denen man ein Problem lösen kann.
Ein Kuchenrezept ist auch ein solcher Algorithmus. Folgt man ihm, so erhält man einen Kuchen. Alle mathematischen Formeln wie z.B. die Dreiecksfläche $A = ½a \cdot h$ sind Algorithmen, denn sie zeigen den Weg auf, mit dem man aus gegebenen Werten einen neuen Wert bestimmt. Algorithmen sind wichtige Methoden der Problemlösung, gerade auch im Bereich der Programmierung. Deshalb ist es sehr lohnenswert, sich mit ihnen zu beschäftigen, da sie die Erfahrung im Programmieren bereichern.

Hier ist ein einfaches Kuchenrezept für einen Fertigteig:

1. Den Teig wie auf der Packung vermerkt zubereiten.
2. Auf zwei Formen verteilen und wie auf der Packung vermerkt backen.
3. 10 Minuten im Ofen abkühlen lassen.
4. Auf hitzebeständige Unterlagen stellen.
5. Vollständig auskühlen lassen.
6. Puddingpulver (Fertigpudding ohne Kochen) und Milch mit dem Mixer aufschlagen.
7. Über die zwei Kuchenböden verteilen.
8. Die Böden übereinander schichten.
9. Mit Schlagsahne überziehen.
10. Genießen!

Wie in einem Programm folgt der Bäcker den einzelnen Schritten von Anfang bis Ende. Zum Schluss hat er einen köstlichen Kuchen erzeugt. Folgt ein Computer den Programmschritten (Algorithmus), so erreicht er auch – hoffentlich – das gewünschte Ziel. Tatsächlich ist ein ganzer Wissenschaftszweig damit beschäftigt zu überprüfen, ob ein bestimmter Algorithmus das gewünschte Ergebnis erzielt. Dieser Vorgang ist ähnlich dem des Beweisens von mathematischen Sätzen anzusehen.

Hinweis: Ein Beispiel für einen Computer-Algorithmus
Für das übliche Runden einer Zahl kann man die folgende Formel verwenden:

A = ganzTeil(A+0,5)

Warum funktioniert diese Formel? Ist der Nachkommateil einer positiven Dezimalzahl kleiner als 0,5, so bleibt die Zahl immer noch kleiner als die nächst höhere natürliche Zahl. Andernfalls wird die nächst höhere Zahl überschritten. Da **ganzTeil(** nur den ganzzahligen Teil einer Dezimalzahl liefert, wird damit das korrekte Ergebnis des Rundens angezeigt.
Beispiele:
A=3,4 A+0,5 = 3,9 ganzTeil(3,9) = 3 abgerundet)
A=3,7 A+0,5 = 4,2 ganzTeil(4,2) = 4 aufgerundet)

Allerdings hat der TI 84 im Menü [MATH] NUM die eingebaute Rundungsfunktion **runde(**.