

restart;

▼ Aufgabe 2: Zeichenketten

`i := 3 + 4;`
`a := "i hat den Wert ";` "i hat den Wert " (1.1)

`b := ".";` "." (1.2)

`cat(a, i, b);` "i hat den Wert 7." (1.3)

▼ Aufgabe 3: Dezimalbruchentwicklung

```
dezbruch := proc(a, b)
  # Verwendete lokale Variablen
  local ggt, i, j, output, dividant, divisor, restepos, werte, periodenlaenge, periodenbeginn;

  # Abbrechen, wenn Nenner 0
  if b = 0 then return undefined; end if;

  # Richtiges Vorzeichen bestimmen
  if a < 0 xor b < 0 then output := "-"; else output := ""; end if;

  # Zähler und Nenner positiv machen.
  # Außerdem kürzen, um Rechenzeit und Speicher zu sparen (für die Aufgabe nicht notwendig).
  ggt := igcd(a, b);
  dividant :=  $\frac{\text{abs}(a)}{\text{ggt}}$ ;
  divisor :=  $\frac{\text{abs}(b)}{\text{ggt}}$ ;

  # Den Teil vor dem Komma bestimmen, Rest als neuen Dividenten speichern
  output := cat(output, iquo(dividant, divisor, 'divident'), ".");

  # Array um Positionen der Reste zu speichern
  restepos := Array(0..divisor - 1);

  # Array um Werte für die Ausgabe zwischenzuspeichern
  werte := Array(1..divisor);

  # Schleife zur "schriftlichen Divison"
  for i while dividant ≠ 0 do

    # Letzten Rest mit 10 multiplizieren, neue Nachkommastelle bestimmen und Rest als neuen
    # Dividenten speichern
    werte[i] := iquo(10·divident, divisor, 'divident');
```

```

if restepos[divident]  $\neq$  0 then # Rest bereits vorhanden, Periode gefunden

  # Periodenlaenge und -beginn bestimmen
  periodenlaenge := i - restepos[divident];
  periodenbeginn := restepos[divident];
  if werte[periodenbeginn]  $\neq$  werte[i] then periodenbeginn := periodenbeginn + 1; end if;

  # Nachkommanteil im periodischen Fall zusammensetzen
  for j to periodenbeginn - 1 do
    output := cat(output, werte[j]);
  end do;
  output := cat(output, "p");
  for j from periodenbeginn to periodenbeginn + periodenlaenge - 1 do
    output := cat(output, werte[j]);
  end do;

  return output;
else
  # Position des Restes merken
  restepos[divident] := i;
end if;
end do;

i := max(i, 2); # Bugfix für ganze Zahlen, damit eine Nachkomma-0 ausgegeben wird

# Nachkommanteil im nicht-periodischen Fall zusammensetzen
for j to i - 1 do output := cat(output, werte[j]); end do;

return output;
end proc;

```

```
dezbruch(1, 1);
```

"1.0" (2.1)

```
dezbruch(-3, 4);
```

"-0.75" (2.2)

```
dezbruch(1, 3);
```

"0.p3" (2.3)

```
dezbruch(1, 6);
```

"0.1p6" (2.4)

```
dezbruch(-1, 700);
```

"-0.00p142857" (2.5)