

Probleme 1

Lerndatei für das Programmieren von Listen und Doppellisten Ha 2011

Listen

$li := seq(i^2, i, 1, 10) \cdot \{1, 4, 9, 16, 25, 36, 49, 64, 81, 100\}$ i heißt Laufvariable.

Anstelle von i^2 kann jeder Term von i stehen.

Die Elemente von li erreicht man so: $li[6] \cdot 36$

$left(li, 3) \cdot \{1, 4, 9\}$ sind links die ersten drei Elemente.

$right(li, 3) \cdot \{64, 81, 100\}$ sind rechts die letzten drei Elemente

$mid(li, 4) \cdot \{16, 25, 36, 49, 64, 81, 100\}$ sind die Elemente ab Nr. 4.

$mid(li, 4, 5) \cdot \{16, 25, 36, 49, 64\}$ sind ab dem 4. Element 5 Elemente.

$leer := \{\}$ ist die leere Liste. Man braucht sie beim Programmieren.

$ung := seq(li[i], i, 1, 10, 2) \cdot \{1, 9, 25, 49, 81\}$ nimmt als der Liste alle Elemente an ungeraden Plätzen, entsprechend $ger := seq(li[i], i, 2, 10, 2) \cdot \{4, 16, 36, 64, 100\}$

$augment(ung, ger) \cdot \{1, 9, 25, 49, 81, 4, 16, 36, 64, 100\}$ fügt zwei Listen aneinander.

$ma := (ung, ger) \cdot \begin{bmatrix} 1 & 9 & 25 & 49 & 81 \\ 4 & 16 & 36 & 64 & 100 \end{bmatrix}$ Listen von Listen sind Matrizen

1.1

Vorbereitung einer Doppelliste

Solche Zeilen wie die beiden Folgenden soll man nicht einzeln eingeben müssen.

$seq(i^3, i, 2, 10) \cdot \{8, 27, 64, 125, 216, 343, 512, 729, 1000\}$

$seq(i^k, i, 2, 10) \cdot \{16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000\}$

$poti := seq(seq(i^k, i, 2, 8), k, 1, 10)$

2	3	4	5	6	7	8
4	9	16	25	36	49	64
8	27	64	125	216	343	512
16	81	256	625	1296	2401	4096
32	243	1024	3125	7776	16807	32768
64	729	4096	15625	46656	117649	262144
128	2187	16384	78125	279936	823543	2097152
256	6561	65536	390625	1679616	5764801	16777216
512	19683	262144	1953125	10077696	40353607	134217728
1024	59049	1048576	9765625	60466176	282475249	1073741824

$poti[6, 7] \cdot 262144$ So kann man auf die Elemente der Doppelliste zugreifen.

1.2

potte 0/2 **Potenztafeln als Doppelliste**

Define **potte(m,n)=**

Func

Local i, k

$seq(seq(i^k, i, 2, n), k, 1, m)$

EndFunc

$potte(4,6) \cdot \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 9 & 16 & 25 & 36 \\ 8 & 27 & 64 & 125 & 216 \\ 16 & 81 & 256 & 625 & 1296 \end{bmatrix}$

$potte(1,6) \cdot [2 \ 3 \ 4 \ 5 \ 6]$

$potte(12,5) \cdot \begin{bmatrix} 2 & 3 & 4 & 5 \\ 4 & 9 & 16 & 25 \\ 8 & 27 & 64 & 125 \\ 16 & 81 & 256 & 625 \\ 32 & 243 & 1024 & 3125 \\ 64 & 729 & 4096 & 15625 \\ 128 & 2187 & 16384 & 78125 \\ 256 & 6561 & 65536 & 390625 \\ 512 & 19683 & 262144 & 1953125 \\ 1024 & 59049 & 1048576 & 9765625 \\ 2048 & 177147 & 4194304 & 48828125 \\ 4096 & 531441 & 16777216 & 244140625 \end{bmatrix}$

1.3

Hinweise zur Seitenteilung und zur Programmierung

Zumindest am PC ist es sinnvoll, die Seite für eine Programmierung zu teilen, dann kann man nämlich gleich ausprobieren ob das Programm oder die Funktion richtig läuft.

Vorgehen PC:

Button Einfügen, während das nun sich öffnende Fenster wartet, Button Seitenlayout und das Passende wählen. Man wird für beide Hälften aufgefordert, eine Applikation zu wählen.

Links nimmt man wieder den Button einfügen und weit unten Programmierer neu.

In dem nun erscheinenden Fenster gibt man der geplanten Funktion (oder dem Programm) einen Namen (ohne irgendwelche Klammern), wählt im nächsten Feld "Funktion" aus (Bib-zugriff soll hier nicht sein) und erhält die Definitionssyntax für eine Funktion.

Hinter dem Funktions- oder Programmnamen trägt man in die Klammern ein, welche Variablen man beim Aufruf übergeben will, hier $potte(m,n)$, also m und n , die Zeilen- und die Spaltenzahl.

Als erste Zeile (unter Func) schreibt man Local und dann die Variablen, die man innerhalb des Programms zu verwenden gedenkt. Die Ausgabe oder Rückgabe der Funktion ist entweder der zusetzt berechnete Wert oder der Wert der Variablen, die hinter dem Wort Return steht. Programme können auch eine text-Ausgabe enthalten, sie laufen aber nur im Calculator. Funktionen laufen auch in Notes.

1.4

Probleme 2

Nutzung von einmal überlegten Vorgehensweisen:

In eine TI-Nspire-Datei kann man sich mehrere "Probleme" eröffnen.

(Mit Einfügen Problem). Diese Probleme sind untereinander Variablen-geschützt.

Daher ist folgendes sinnvoll:

Mache eine neues Problem auf. Kopieren eine wichtige Seite, hier die mit der Definition von "potte", der Doppelliste. Gehe dazu in den Seitensortierer

(Strg Nord am Handheld, am PC links 2. Eintrag in der Werkzeugpalette)

Markiere die Seite (mit sh auch mehrere) aus Problem 1 mit Strg C und gehe im Seitensortierer in Problem 2 und füge mit Strg V die Seite dort ein.

Im Folgenden dann anstelle von i^k der Parabelterm $(x-i) \cdot (x-k)$ eingeben.

Dan sind die "Laufbereiche" angepasst (Start 0 beide Male).

Dann muss "Syntax überprüfen und speichern" erfolgen.

In einem Notefenster erscheint mit dem Aufruf jedem $potte(m,n)$ eine Tabelle mit allen Parabeltermen, die die Nullstellen ganzzahlig zw. m und n (incl.) haben.

2.1

potte 2/2 **potte(6,3)**

Define **potte(m,n)=**

Func

Local i, k

$seq(seq((x-i) \cdot (x-k), i, 0, n), k, 0, m)$

EndFunc

$potte(3,1) \cdot \begin{bmatrix} x^2 & x(x-1) \\ x(x-1) & (x-1)^2 \\ x(x-2) & (x-2)(x-1) \end{bmatrix}$

2.2

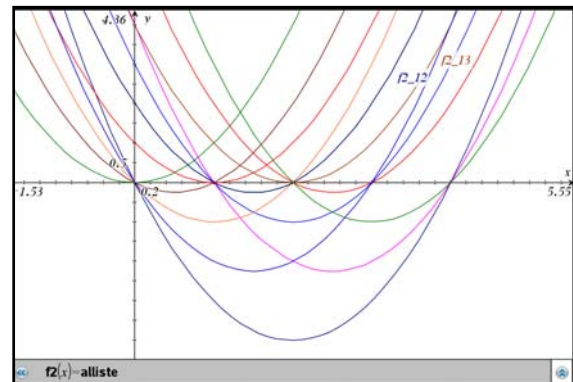
$alle := potte(4,4) \cdot \begin{bmatrix} x^2 & x(x-1) & x(x-2) & x(x-3) & x(x-4) \\ x(x-1) & (x-1)^2 & (x-2)(x-1) & (x-3)(x-1) & (x-4)(x-1) \\ x(x-2) & (x-2)(x-1) & (x-2)^2 & (x-3)(x-2) & (x-4)(x-2) \\ x(x-3) & (x-3)(x-1) & (x-3)(x-2) & (x-3)^2 & (x-4)(x-3) \\ x(x-4) & (x-4)(x-1) & (x-4)(x-2) & (x-4)(x-3) & (x-4)^2 \end{bmatrix}$

$alliste := mat \rightarrow list(alle)$

Im Graphfenster kommen leider nicht alle Parabeln.

Das leider ein Mangel, schief.

2.3



2.4