

## U12A Horner Schema II

### Aufgabe

Es ist ein C-Funktion zu entwickeln, die ein Polynom  $p(x)$  beliebigen Grades ( $\deg p > 1$ ) mit reellen Koeffizienten an einer reellen Stelle  $x$  auswertet.

### Randbedingungen

1. Das Resultat soll als Funktionswert zurückgegeben werden.
2. Die Koeffizienten des Polynoms sind als double-Typ zu wählen und werden in einem Array als Parameter übergeben.
3. Keine globalen Variablen und Seiteneffekte

### Test

Berechnen Sie konkret unter Zuhilfenahme der soeben entwickelten Funktion den Wert des Polynoms

$$p(x) = \sum_{i=0}^{100} i \cdot x^i = 100x^{100} + 99x^{99} + 98x^{98} + \dots + 2x^2 + x + 0$$

an den Stellen  $x_1=1$  und  $x_2=2$ .

### Vorgehen

- Definition der Schnittstellen zur Funktion. Resultattyp, Parameter, Name der Funktion. Daraus Resultiert der Funktionsprototyp.
- Beispiel von Hand rechnen, unter Berücksichtigung der später zu verwendenden Datenstrukturen.
- Umsetzen des Rechenbeispiels in einen allgemeingültigen C-Code.
- Verfassen eines Programmgerüsts zum Test der Funktion.

## Horner Schema (Lösung)

Funktionsprototyp: `double horner(double p[], double x, int n);`

Aus der Aufgabenstellung ist der Name `horner` vorgegeben. Ebenso der Typ der Funktion, d.h. der Datentyp `double` den die Funktion als Resultat retourniert. Die Parameter sind nur teilweise durch die Aufgabenstellung vorgegeben. Die Polynomkoeffizienten werden über eine Zeigerreferenz `p[]` übergeben. `p[0]` verkörpert den Koeffizienten zu  $x^0$ , `p[1]` zu  $x^1$ , etc. Die Stelle `x`, an der die Auswertung erfolgen soll, wird als `double` übergeben. Die Dimension des Arrays, d.h. Anzahl Polynomkoeffizienten wird als Ganzzahl `n` übergeben. Dies ist notwendig, da in der Funktion mit `sizeof()` nicht die Anzahl der Elemente im Array bestimmt werden kann.

Wir wählen als Beispiel das Polynom  $p(x) = 2x^5 - x^4 + 3x^2 - x + 2$ , welches an der Stelle  $x=2$  ausgewertet wird. Es dient nachher auch als Grundlage für den Test.

```
/* Funktion horner()
Auswerten eines Polynoms mit reellen Koeffizienten, welche im Array p[] uebergeben werden an der
Stelle x.
Das Resultat wird als Funktionswert retourniert.
*/
double horner(double p[], double x, int n)
{ long double r;
  int i;

  r = p[n];
  for (i = n-1; i >= 0; i--)
    r = r * x + p[i];

  return (double) r;
}
```

/\* Aufgabe IAM: Polynomauswertung nach dem Verfahren von Horner.

Auswerten des Polynoms  $p(x) = 100x^{100} - 99x^{99} + \dots - x + 0$  an den Stellen  $x_1=1$  und  $x_2=2$ .

```
Autor: Gerhard Krucker
Datum: 28.4.1997
Sprache: MS Visual-C V4.1 (NT Console Application)
Filename: hornerSchema.c
*/
#include <stdio.h>
#include <math.h>
#include <conio.h>

// #define DEBUG

/* Funktion horner()
Auswerten eines Polynoms mit reellen Koeffizienten, welche im Array p[]
uebergeben werden an der Stelle x.
Das Resultat wird als Funktionswert retourniert.
*/
double horner(double p[], double x, int n)
{ long double r;
  int i;

  r = p[n];
  for (i = n-1; i >= 0; i--)
    r = r * x + p[i];

  return (double) r;
}
```

```
main()
{ int i;
double resultat;
double x;

#ifdef DEBUG
double a[]={3.0,-1.0,0.0,3.0,-1.0,2.0}; /* Testpolynom, ergibt p(2)=73 */
#else
double a[101];
for (i=0; i<=100; i++) a[i]=i * pow(-1.0, (double)( i % 2)); /* Polynomkoeffizienten aufbereiten */
#endif

printf("Auswerten eines Polynomes nach dem Verfahren von Horner.\n");

#ifdef DEBUG
x=1.0;
resultat=horner(a,x,sizeof(a)/sizeof(double)-1);
printf("p(x=%f)=%f\n",x,resultat);
#endif
x=2.0;
resultat=horner(a,x,sizeof(a)/sizeof(double)-1);
printf("p(x=%f)=%f\n",x,resultat);

while (!_kbhit());

return 0;
}
```