

Kapitel 9 Beispiel 7

```

1 // Programm935.cpp: Hauptprojektdatei.
2 // Erstellung einer Wahrheitstabelle für das Problem
3 // Q1 = (A&&B&&C) || (A&&!B&&!C) || (!A&&B&&!C) || (!A&&!B&&C)
4 // Q2 = (A&&B&&C) || (A&&B&&!C) || (A&&!B&&C) || (!A&&B&&!C)
5 // Eingabe: Anzahl der Zustände (hier 3)
6 // Ausgabe: Wahrheitstabelle für Q1 und Q2
7 // Autor: Heiderich / Meyer
8 // -----
9 #include "stdafx.h"
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <conio.h>
13 #include <math.h>
14 #include "dualtabelle.h"
15 #include "q1.h"
16 #include "q2.h"
17 using namespace System;
18 void main()
19 {
20     int anzE;           // Anzahl der Eingänge
21     int anzA;           // Anzahl der Eingänge
22     short int a;        // Wert Eingang A
23     short int b;        // Wert Eingang B
24     short int c;        // Wert Eingang C
25     int anz_zeile;      // Anzahl der Zeilen in den Arrays
26     int anz_spalteE;    // Anzahl der Spalten im Array Eingänge
27     int anz_spalteA;    // Anzahl der Spalten im Array Ausgänge
28     int i;              // Hilfsvariable
29     short int ** dual;  // zweidimensionales Array der 3 Eingänge
30     short int ** q;     // zweidimensionales Array der 2 Ausgänge
31     // Begrüßung
32     printf("\n\n\tProgramm zur Erstellung einer Wahrheitstabelle
33           f%cr die Probleme\n",char(129));
34     printf("\n\t\tQ1 = (A&&B&&C) ||
35           (A&&!B&&!C) || (!A&&B&&C) und\n");
36     printf("\n\t\tQ2 = (A&&B&&C) ||
37           (A&&B&&!C) || (A&&!B&&C) || (!A&&B&&!C)\n");
38     printf("\t\t-----\n");
39     // Anzahl der Eingänge setzen
40     anzE = 3;
41     // Anzahl der Ausgänge setzen
42     anzA = 2;
43     // dynamisches Allocieren zweier zweidimensionaler Arrays
44     anz_zeile = (int)pow(2.,anzE) + 1;
45     anz_spalteE = anzE + 1;
46     anz_spalteA = anzA + 1;
47     dual = (short int **)malloc(anz_zeile * sizeof(short int *));
48     if (dual == NULL)
49     {
50         printf("Kein RAM mehr vorhanden f%cr dual ... !",char(129));
51         getch();
52         exit(0);
53     }
54     q = (short int **)malloc(anz_zeile * sizeof(short int *));
55     if (q == NULL)
56     {
57         printf("Kein RAM mehr vorhanden f%cr q ... !",char(129));
58         getch();
59         exit(0);
60     }
61     for(i = 0; i < anz_zeile; i++)
62     {
63         dual[i] = (short int *)malloc(anz_spalteE*sizeof(short int));
64         if (dual[i] == NULL)
65         {
66             printf("Kein RAM mehr f%cr Zeile %d von
67                   dual\n",char(129),i);
68             getch();
69             exit(0);
70         }
71         q[i] = (short int *)malloc(anz_spalteA * sizeof(short int));
72         if (q[i] == NULL)
73         {
74             printf("Kein RAM mehr f%cr Zeile %d von q\n",char(129),i);
75             getch();
76             exit(0);
77         }
78     }
79     // Aufruf der Belegungsfunktion für die Dualzahlentabelle
80     dualtabelle(anzE,dual);

```

```

77 // Erstellung der Wahrheitstabelle
78 for (i = 1; i < anz_zeile; i++)
79 {
80 // Belegung der Eingänge aus der Dualzahlentabelle
81 a = dual[i][1];
82 b = dual[i][2];
83 c = dual[i][3];
84 // Ermittlung der Ausgänge
85 q[i][1] = q1(a,b,c);
86 q[i][2] = q2(a,b,c);
87 }
88 // Ausgabe der Tabelle
89 printf("\n\t A B C | Q1 Q2\n");
90 printf( "\t-----+-----\n");
91 for (i = 1; i < anz_zeile; i++)
92 {
93     printf("\t %li %li %li | %li %li\n",
94           dual[i][1],dual[i][2],dual[i][3],q[i][1],q[i][2]);
95 }
96 getch();

```