

```

1 // Vorlage zur Vorlesung C
2
3 // Erstellt von Sven Wambsganß/ Karlsruhe
4 // © 2010 Sven Wambsganß - www.sven-wambsganss.de
5 // * Keine Haftung für Fehler *
6
7 // Diese Vorlage bezieht sich auf das Buch
8 // "ANSI C 2.0 - Grundlagen Programmierung, Herdt Verlag, 2. Auflage"
9
10 // Bibliotheken einbinden "Seite 144 - 169"
11 #include <stdio.h>
12
13 // Konstante Definieren (ältere Form) "Seite 121 - 122"
14 #define PI 3.14
15
16 // Variablen / Datentyp "Seite 26 - 29"
17 int i, Variable, anzahl=5; // Typ int
18 char zahlenfolge[3]={1,3,4}; // von zahlenfolge[0] bis zahlenfolge[2]
19 int matrix[3][4]; // Matrix mit 3 Zeilen und 4 Spalten
20 const float PI = 3.14; // Konstante Definieren "Seite 16"
21 const char Zeichen = 'J'; // Konstante Definieren "Seite 16"
22 Typ, Name //Name darf nicht mit "Sonderzeichen" oder "0" beginnen, "_" möglich!!
23 char //char // 8 Bit = 1 Byte // 0...255 oder -128...+127
24 short //int* // 16Bit = 2 Byte // 0...65535 oder -32768...+32767
25 long //int* // 32Bit = 4 Byte // 0...4'294'967'295 oder
26 *Maschinenabhängig // -2'147'483'648...+2'147'483'647
27 float // 7 Stellen genau // 32Bit = 4 Byte //1.18*10e-38...3.4*10e38
28 double //15 Stellen genau // 64Bit = 8 Byte //2.23*10e-308...1.79*10e308
29 long double //18 Stellen genau // 80Bit = 10 Byte //3.37*10e-4932...1.18*10e4932
30
31 int Ganze_Zahlen // %i, %d
32 float, double Gleitkommazahlen // %e, %E, %f, %g
33 char Buchstaben... // %c
34
35 %d oder %i //int, Dezimalzahl, Ganze_Zahlen
36 %u //unsigned int, Ganze_Zahlen ohne Vorzeichen
37 %c //einzelnes Zeichen /Buchstabe
38 %e oder %E //double im Format [-]d.ddd e+dd bzw. [-]d.ddd E+dd
39 %f //double im Format [-]ddd.ddd
40 %g //double
41 %s //Zeichenkette ausgeben
42 %o //int als Oktalzahl ausgeben
43 %x oder %X //int als Hexadezimalzahl ausgeben z.B. ff oder FF
44 %#x oder %#X//int als Hexadezimalzahl ausgeben z.B. 0xff oder 0XFF
45 %% //Prozentzeichen
46
47 %7.2f //enthält 7 Stellen, inkl .und 2 Nachkommastellen xxxx.xx
48 %7f //rechtsbündig ausgeben: x;
49 %07f //linksbündig ausgeben: 000000x; und mit "0" auffüllen
50 %-7f //linksbündig ausgeben: x ;
51 %+7f //rechtsbündig ausgeben: +x; mit Vorzeichen Plus
52 %+07.2f //rechtsbündig ausgeben: +00x.xx; 7 Stellen inkl + und .
53 %+-7.2f //linksbündig ausgeben: +x.xx; 7 Stellen inkl + und .
54
55 // Bitoperationen "Seite 23 - 24, 170"
56 x >> = 1 //x = x >> 1 //Bits um eine Stelle nach rechts schieben
57 x << = 1 //x = x << 1 //Bits um eine Stelle nach links schieben
58 x &= 1 //x = x & 1 //x wird mit Zahl 1 über UND verknüpft und in x gespeichert
59 x |= 1 //x = x | 1 //x wird mit Zahl 1 über ODER verknüpft und in x gespeichert
60 x ^= 1 //x = x ^ 1 //x wird mit Zahl 1 über Exklusiv-ODER verknüpft und in x
    gespeichert
61
62 // Arithmetische Operationen "Seite 22, 24, 170"
63 x += y //x = x + y
64 x -= y //x = x - y
65 x *= y //x = x * y
66 x /= y //x = x / y
67 x %= y //x = x % y //Modulo = Restwert! " 5 % 3 = (1) Rest 2"
68 x++ //x = x + 1
69 x-- //x = x - 1
70
71
72
73
74

```

```

75 // © Sven Wambsganß 2010 - www.sven-wambsganß.de - Keine Haftung für Fehler
76
77 //x² #include <math.h> "Seite 162"
78 double ergebnis,x,y=2;
79 ergebnis = pow (x,y); //double pow (double x, double y);
80
81 //Wurzel #include <math.h> "Seite 162"
82 double ergebnis,x;
83 ergebnis = sqrt(x); //double sqrt(double x);
84
85 // Sonderkomandos
86 break; // Abbruch "Seite 52, 60"
87 /* Kommentar über mehrere Zeilen*/ "Seite 9"
88 // Kommentar für eine Zeile
89
90 // Funktion Allgemein "Seite 30 - 38"
91 int rechnen(void) {...} //Funktion hat keinen Parameter
92 void rechnen(void){...} //Funktion gibt keinen Wert zurück und hat keinen Parameter
93 void rechnen(int) {...} //Funktion gibt keinen Wert
94
95 //-----
96 // Funktion steht vor dem Mainprogramm "Seite 30 - 38"
97 int rechnen(int a, int b) //Funktion
98 {
99     int c; //Deklaration der lokalen Variable
100     c=a+b; //Rechenfunktion
101     return c; //Rückgabewert
102 }
103 // Hauptprogramm Typ 1 "Seite 12 - 15"
104 void main(void)
105 {
106     int ergebnis, i=1, j=2; //Deklaration der lokalen Variablen
107     ergebnis = rechnen(i,j); //Aufruf einer Funktion rechnen
108     printf ("Loesung = %i ",ergebnis); //Programm
109 }
110
111 //-----
112 // Funktion steht nach dem Mainprogramm "Seite 30 - 38"
113 // mit Funktionsprototyp
114 int rechnen(int a, int b); //Funktionsprototyp
115
116 // Hauptprogramm Typ 2 "Seite 12 - 15"
117 int main(void)
118 {
119     int ergebnis, i=1, j=2; //Deklaration der Variablen
120     ergebnis = rechnen(i,j); //Aufruf einer Funktion
121     printf ("Loesung = %i",ergebnis); //Programm
122     return 0;
123 }
124
125 // Funktion "Seite 30 - 38"
126 int rechnen(int a, int b) //Funktion
127 {
128     return (a+b); //Rückgabewert
129 }
130
131 //-----
132 // Ausgabe "Seite 14 - 15"
133
134 printf("Text oder Sonderzeichen z.B. \x9a \bber %i eingeben \n", Variable+1);
135 // Ü als Hexzahl + Schreibmarke um 1Zeichen zurück + ber
136
137 //Beispiele mit Rechenoperation
138 {
139     int a=100;
140     printf("Zahl = %d \n",a++); //Gibt 100 aus und erhöht danach a auf 101
141     printf("Zahl = %d \n",++a); //Erhöht a auf 102 und gibt 102 aus
142     printf("Zahl = %d \n",a+1); //Gibt 103 aus a bleibt 102
143     printf("Zahl = %d \n",a=a+1); //Erhöht a auf 103 und gibt 103 aus
144     printf("Zahl = %d \n",a+=1); //Erhöht a auf 104 und gibt 104 aus
145     printf("Zahl = %d \n",a--); //Gibt 104 aus und reduziert danach a auf 103
146     printf("Zahl = %d \n",--a); //Reduziert a auf 102 und gibt 102 aus
147 }
148
149

```

```

150 // © Sven Wambsganß 2010 - www.sven-wambsganss.de - Keine Haftung für Fehler
151
152 // weitere auf "Seite 28"
153 \0 // Textende
154 \n // Zeilenumbruch
155 \b // Schreibmarke um 1Zeichen zurück
156 \' // Ausgabe von '
157 \? // Ausgabe von ?
158 \" // Ausgabe von "
159 \\ // Ausgabe von \
160 \xWert // Wandeln in Hexadezimal
161 \Wert // Wandeln in Oktal
162 // weitere auf "Seite 28"
163
164 // Einlesen von Zahlen "Seite 40 - 43"
165 int Zahl; //Variable
166 fflush(stdin); //Tastaturspeicher löschen
167 scanf("%d",&Zahl); //Einlesen der Zahl
168
169 // Einlesen von Buchstaben
170 char Variable; //Variable
171 fflush(stdin); //Tastaturspeicher löschen
172 scanf("%c",&Variable); //Einlesen des Buchstabens
173
174 // Einlesen von String
175 char Name[20]; //Variable
176 fflush(stdin); //Tastaturspeicher löschen
177 scanf("%s",&Name); //Einlesen des String
178
179 //-----
180 // Vergleiche "Seite 50 - 51"
181 if (i>0) //Prüfen ob i größer 0 ist
182 //if (zahlenfolge[i] == zahlenfolge[j]) //Prüfen auf Gleichheit der 2 Variablen
183 //if (zahl >= 0 && zahl <= 255) //Prüfen ob Zahl zwischen 0 und 255 liegt
184 {
185     printf("true"); //Anweisung bei wahrer Bedingung
186 }
187 else
188 {
189     printf("false"); //Anweisung bei falscher Bedingung
190 }
191
192 //-----
193 // Schleifen Kopfgesteuer "Seite 55 - 58"
194 // for- Schleife vorwärts
195 int i = 0, anzahl = 5; // 5 Durchläufe der Schleife
196 for (i=0; i<anzahl; i++)
197 {
198     printf("Zahl = %i \n",i); //Ausgabe: "Zahl = 0 ... Zahl = 4"
199 }
200
201 // for- Schleife rückwärts
202 int i; anzahl = 5; // 5 Durchläufe der Schleife
203 for (i=anzahl-1; i>=0; i--)
204 {
205     printf("Zahl = %i \n",i); //Ausgabe: "Zahl = 4 ... Zahl = 0"
206 }
207
208 // while- Schleife vorwärts
209 int i = 0; anzahl = 5; // 5 Durchläufe der Schleife
210 while (i<anzahl) // Führt die Schleife aus, bis die Bedingung erfüllt ist
211 {
212     printf("Zahl = %i \n",i); //Ausgabe: "Zahl = 0 ... Zahl = 4"
213     i++;
214 }
215
216 // while- Schleife rückwärts
217 int i,anzahl =5;
218 i=anzahl-1; // 5 Durchläufe der Schleife
219 while (i>=0) // Führt die Schleife aus, bis die Bedingung erfüllt ist
220 {
221     printf("Zahl = %i \n",i); //Ausgabe: "Zahl = 4 ... Zahl = 0"
222     i--;
223 }
224

```

```

225 // © Sven Wambsganß 2010 - www.sven-wambsganss.de - Keine Haftung für Fehler
226
227 // Schleifen Fußgesteuert "Seite 55 - 58"
228 // do- while- Schleife vorwärts
229 int i=0, anzahl =5; // 5 Durchläufe der Schleife
230 do // Führt die Schleife aus, bis Bedingung am Ende erfüllt ist
231 {
232     printf("Zahl = %i \n",i); //Ausgabe: "Zahl = 0 ... Zahl = 4"
233     i++;
234 }
235 while (i<anzahl);
236
237
238 // do- while- Schleife rückwärts
239 int i,anzahl =5;
240 i=anzahl-1; // 5 Durchläufe der Schleife
241 do // Führt die Schleife aus, bis Bedingung am Ende erfüllt ist
242 {
243     printf("Zahl = %i \n",i); //Ausgabe: "Zahl = 4 ... Zahl = 0"
244     i--;
245 }
246 while (i>=0);
247
248 //-----
249 // Zahlenfolge ausgeben nach Größe geordnet
250 for (i=1; i<anzahl; i++)
251 {
252     zaehlen_der_vertauschungen = 0; //Vertauschung auf 0 setzen
253     for (j=1; j<anzahl-i+1; j++)
254     {
255         if (zahlenfolge[j-1] > zahlenfolge[j]) //Sortieren nach Größe
256         {
257             zwischenspeicher=zahlenfolge[j-1]; //Tausch der Zahlen
258             zahlenfolge[j-1]=zahlenfolge[j];
259             zahlenfolge[j]=zwischenspeicher;
260
261             zaehlen_der_vertauschungen++;
262         }
263     }
264     // Wenn es 0 Vertauschungen gibt, dann wird die Schleife beenden
265     if (zaehlen_der_vertauschungen == 0)
266     {
267         break; //Abbruch der Schleife
268     }
269 }
270
271 //-----
272 // Sprungmarke "Seite 60 - 61"
273 Merker: //Sprungmarke
274 goto Merker; //Springe zu Sprungmarke
275
276 //-----
277 // case "Seite 52 - 54"
278 int zahl;
279 char wert;
280 switch (zahl oder wert)
281 {
282     case 1 : Anweisung; break; // Zahlen
283     case 'A': Anweisung; break; // Buchstaben
284     default : Anweisung; // undefinierter Wert
285 }
286
287
288 // Rekursion "Seite 62 - 69"
289 // enum "Seite 71 - 72"
290 // Arrays "Seite 72 - 80"
291 // Zeiger "Seite 88 - 119"
292 // define "Seite 120 - 127"
293 // Bibliotheken "Seite 144 - 169"
294 // ASCII "Seite 171"
295
296
297 /* Version 2.3 Stand 02.05.2010 */

```