

```
1  #include <iostream>
2
3  using std::cout;
4  using std::cin;
5  using std::endl;
6
7  void Ausgabe( const double *feld, int n)
8  {
9      for (int i=0; i<n; ++i)
10         cout << feld[i] << " ";
11     cout << endl;
12 }
13
14 double Durchschnitt( const double *feld, int n)
15 {
16     double sum= 0;
17     for (int i=0; i<n; ++i)
18         sum+= feld[i];
19
20     return sum/n;
21 }
22
23 int min_pos( const double *feld, int n)
24 {
25     if (n<=0) // leeres Feld
26         return -1;
27
28     double min= feld[0];
29     int pos= 0;
30     for (int i=1; i<n; ++i)
31     {
32         if (feld[i] < min)
33         {
34             min= feld[i];
35             pos= i;
36         }
37     }
38
39     return pos;
40 }
41
42 void sortiere( double *feld, int n)
43 {
44     for (int i=0; i<n; ++i)
45     {
46         // suche min. Element ab feld[i]
47         const int pos= i + min_pos( feld+i, n-i);
48         // statt feld+i kann man auch &(feld[i]) schreiben
49
50         if (pos==i)
51             continue;
52         // tausche feld[i] mit feld[pos]
53         double tmp= feld[i];
54         feld[i]= feld[pos];
55         feld[pos]= tmp;
56     }
57 }
58
59
60
```

```
61 void umdrehen( double *feld, int n)
62 {
63     for (int i=0; i<n/2; ++i)
64     {
65         const int imirror= n - 1 - i; // Position spiegelbildlich zu i
66
67         // tausche feld[i] mit feld[imirror]
68         double tmp= feld[i];
69         feld[i]= feld[imirror];
70         feld[imirror]= tmp;
71     }
72 }
73
74 double* glue( const double *feld1, int n1, const double *feld2, int n2)
75 {
76     double *gluefeld= new double[n1+n2]; // dynamischer Speicher
77
78     for (int i=0; i<n1; ++i)
79         gluefeld[i]= feld1[i];
80
81     for (int i=0; i<n2; ++i)
82         gluefeld[i+n1]= feld2[i];
83
84     return gluefeld;
85 }
86
87 int main()
88 {
89     double feld[]= { 47, 11, 0, 8, 15};
90     const int len= 5;
91
92     Ausgabe( feld, len); // Ausgabe des Feldes
93     Ausgabe( feld+2, len-2); // Ausgabe ab dem dritten Eintrag
94     cout << "Durchschnitt:\t" << Durchschnitt( feld, len) << endl;
95     int pos= min_pos( feld, len);
96     cout << "min. Element ist" << feld[pos] << "und befindet sich an"
97         "Position" << pos << endl;
98
99     cout << "sortiere das Feld...\n";
100     sortiere( feld, len);
101     Ausgabe( feld, len);
102
103     cout << "drehe das Feld um...\n";
104     umdrehen( feld, len);
105     Ausgabe( feld, len);
106
107     cout << "Teste glue:\n";
108     double *neuesFeld= glue( feld, len, feld, len);
109     Ausgabe( neuesFeld, 2*len);
110     delete[] neuesFeld; // wichtig: Speicher wieder freigeben!
111
112     return 0;
113 }
```