

```
1 // Exercise 1.18 and Exercise 1.19
2
3 #include <math.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 double Energy1(double alpha)
8 {
9     double pi = 4.0 * atan(1.0);
10    double N2 = pow(2.0 * alpha / pi, 1.5);
11    double energy = N2 *
12        (alpha * sqrt(pi / (2.0 * alpha)) -
13         (alpha * alpha * sqrt(pi) /
14          pow(2.0 * alpha, 1.5)));
15    return energy - 1.0;
16 }
17
18 double Energy2(double alpha)
19 {
20     double pi = 4.0 * atan(1.0);
21     double N2 = pow(2.0 * alpha / pi, 1.5);
22     double integ1 = 2.0 * sqrt(pi) /
23         (8.0 * pow(2.0 * alpha, 1.5));
24     double integ2 = 24.0 * sqrt(pi) /
25         (32.0 * 2.0 * pow(2.0 * alpha, 2.5));
26     double integ3 = 1.0 / (4.0 * alpha);
27     double term1 = 3.0 * alpha * 4.0 * pi * N2 * integ1;
28     double term2 = -4.0 / 2.0 * 4.0 * pi * alpha * alpha *
29         N2 * integ2;
30     double term3 = -4.0 * pi * N2 * integ3;
31     double energy = term1 + term2 + term3;
32     return energy;
33 }
34
35 double tol(double x)
36 {
37     return (fabs(x) * 1.0e-6 + 1.0e-6);
38 }
39
40 double minin(double* x, double* a, double* b, double (*fx)(double),
41             double (*tolx)(double))
42 {
43     double z, c, d = 0, e, m, p, q, r, tol, t, u, v, w, fu, fv, fw, fz;
44
45     c = (3.0 - sqrt(5.0)) / 2.0;
46     if (*a > *b) {
47         z = *a;
48         *a = *b;
49         *b = z;
50     }
51     w = *x = *a;
52     fw = (*fx)(*x);
```

```
53     z = *x = *b;
54     fz = (*fx)(*x);
55     if (fz > fw) {
56         z = w;
57         w = *x;
58         v = fz;
59         fz = fw;
60         fw = v;
61     }
62     v = w;
63     fv = fw;
64     e = 0.0;
65     while (1) {
66         m = (*a + *b) * 0.5;
67         tol = (*tolx)(*x);
68         t = tol * 2.0;
69         if (fabs(z - m) <= t - (*b - *a) * 0.5) break;
70         p = q = r = 0.0;
71         if (fabs(e) > tol) {
72             r = (z - w) * (fz - fv);
73             q = (z - v) * (fz - fw);
74             p = (z - v) * q - (z - w) * r;
75             q = (q - r) * 2.0;
76             if (q > 0.0)
77                 p = -p;
78             else
79                 q = -q;
80             r = e;
81             e = d;
82         }
83         if (fabs(p) < fabs(q * r * 0.5) && p > (*a - z) * q && p < (*b - z) * q) ↗
84             {
85                 d = p / q;
86                 u = z + d;
87                 if (u - (*a) < t || (*b) - u < t) d = ((z < m) ? tol : -tol);
88             }
89         else {
90             e = ((z < m) ? *b : *a) - z;
91             d = c * e;
92         }
93         u = *x = z + ((fabs(d) >= tol) ? d : ((d > 0.0) ? tol : -tol));
94         fu = (*fx)(*x);
95         if (fu <= fz) {
96             if (u < z)
97                 *b = z;
98             else
99                 *a = z;
100         }
101         v = w;
102         fv = fw;
103         w = z;
104         fw = fz;
105         z = u;
```

```
104         fz = fu;
105     }
106     else {
107         if (u < z)
108             *a = u;
109         else
110             *b = u;
111         if (fu <= fw) {
112             v = w;
113             fv = fw;
114             w = u;
115             fw = fu;
116         }
117         else
118             if (fu <= fv || v == w) {
119                 v = u;
120                 fv = fu;
121             }
122     }
123 }
124 *x = z;
125 return fz;
126 }
127
128 void PrintResults(
129     double energy, double alpha, double error, int exercise)
130 {
131     double pi = 4.0 * atan(1.0);
132     if (exercise == 1)
133         printf_s("Exercise 1.18 Results\n");
134     else
135         printf_s("Exercise 1.19 Results\n");
136     printf_s("My Best Energy      = %-13.10lf\n", energy);
137     printf_s("My Best Alpha        = %+13.10lf\n", alpha);
138     printf_s("My Percent Error      = %+13.10lf\n", error);
139     if (exercise == 1)
140         printf_s("QC Best Energy          = %-13.10lf\n", -1.0 / pi);
141     else
142         printf_s("QC Best Energy          = %-13.10lf\n", -4.0 / (3.0 * pi));
143     printf_s("Experimental Energy = %-13.10lf\n", -0.5);
144     if (exercise == 1)
145         printf_s("QC Percent Error      = %+13.10lf\n",
146             100.0 * fabs(-1.0 / pi + 0.5) / 0.5);
147     else
148         printf_s("QC Percent Error      = %+13.10lf\n",
149             100.0 * fabs(-4.0 / (3.0 * pi) + 0.5) / 0.5);
150 }
151
152 int main()
153 {
154     double alpha1 = 0.0, alpha2 = 0;
155     double pi = 4.0 * atan(1.0);
```

```
156     double a1 = 1.45;
157     double b1 = 1.50;
158     double m1 = minin(&alpha1, &a1, &b1, Energy1, tol);
159     double e1 = Energy1(alpha1);
160     double error1 = 100.0 * fabs(-0.5 - e1) / 0.5;
161     PrintResults(e1, alpha1, error1, 1);
162     double a2 = 0.25;
163     double b2 = 0.50;
164     double m2 = minin(&alpha2, &a2, &b2, Energy2, tol);
165     double e2 = Energy2(alpha2);
166     double error2 = 100.0 * fabs(-0.5 - e2) / 0.5;
167     PrintResults(e2, alpha2, error2, 2);
168     return 0;
169 }
```