```cpp
#include<iostream>
using std::cin; using std::cout; using std::cerr; using std::endl;

#include<cmath>
using std::exp; using std::sin;

double f( double x)
{
    return exp( sin( x)) - (x - 1)*(x - 6);
}

void Wertetabelle( double a, double b, int N)
{
    double h= (b - a)/N,
           x= a;

    cout << "x\tf(x)\n----------------------\n";

    for (int i=0; i<=N; i++)
    {
        cout << x << "\t" << f(x) << "\n";
        x+= h;
    }
    cout << endl;
}

double MPRegel( double a, double b, int N)
// Mittelpunktsregel zur Approximation des Integrals
{
    double h= (b - a)/N,
           sum= 0;

    for (int i=1; i<=N; i++)
    {
        double xi= a + (i - 0.5)*h;
        sum+= f(xi);
    }

    return sum*h;
}

bool Einschluss( double fa, double fb)
// Praedikat: Vorzeichenwechsel bei fa, fb ?
{
    return fa*fb <= 0;
}

double Bisektion( double a, double b, double eps)
// Bisektionsverfahren zur iterativen Nullstellenbestimmung
{
    double fa= f(a),
           fb= f(b);

    if ( !Einschluss( fa, fb) ) // kein Einschluss
    {
        cerr << "Das␣Intervall␣[␣" << a << ",␣" << b << "␣]␣liefert␣keinen␣"
             << "Einschluss!␣Abbruch!\n\n";
        return 999;
    }

```

```
61      while ( b-a > eps )
62      {
63          double m= (a + b)/2,
64                 fm= f(m);     // nur eine Funktionsauswertung pro Iteration!
65
66          if (Einschluss( fa, fm)) // linkes Intervall [a,m]
67          {
68              b= m;     fb= fm;
69          }
70          else                     // rechtes Intervall [m,b]
71          {
72              a= m;     fa= fm;
73          }
74      }
75
76      return (a + b)/2;
77  }
78
79  int main()
80  {
81
82      double a, b;
83      int N;
84      cout << "a, b, N = ";     cin >> a >> b >> N;
85
86      Wertetabelle( a, b, N);
87      cout << "MPRegel liefert " << MPRegel( a, b, N) << endl;
88
89      Wertetabelle( 0, 7, 7);
90
91      double x1= Bisektion( 0, 1, 1e-8),
92             x2= Bisektion( 6, 7, 1e-8);
93
94      cout << "1. Nullstelle: " << x1 << ", f(x) = " << f(x1) << endl
95           << "2. Nullstelle: " << x2 << ", f(x) = " << f(x2) << endl;
96
97      cout << "\nAnzahl Unterteilungen = ";    cin >> N;
98
99      cout << "Annaeherung des Integrals: " << MPRegel( x1, x2, N) << endl;
100 }
```