```
#include <iostream>
#include <cstdlib>
#include <cmath>
using namespace std;

#include "riemann.h"

double f( double x )
{
   return( sin(x) );
}

int main()
{
    double a = 0.0;
    double b = M_PI / 2;
    int n = 100;
    short iflag = 3;
    char type[16] = "right endpoints";

    cout.setf( ios::scientific );
    cout.precision(6);

    cout << "Riemann sum using " << n << " subintervals and " << type << ": "
         << riemann( f, a, b, n, iflag ) << endl;

    cout << "\n\n"; system( "PAUSE" );
    return( EXIT_SUCCESS );
}
```

---
riemann.h

```
#include <cmath>

// This function returns the Riemann sum for f(x) on [a,b] using n subintervals.
//      iflag = 1 ---> Left endpoints are used
//      iflag = 2 ---> Midpoints are used (default value)
//      iflag = 3 ---> Right endpoints are used
double riemann( double (*f)( double x ), double a, double b, int n = 100,
                short iflag = 2 )
{
    double c, sum, h = ( b - a ) / n;

    if ( n <= 0 ) {
        n = 100;
        cout << "\n\nWarning: Bad number of subintervals in riemann().\n";
        cout << "Proceeding with n = 100.\n\n";
    }

    if ( iflag == 1 ) c = a;
    else if ( iflag == 2 ) c = a + h / 2;
    else if ( iflag == 3 ) c = a + h;
    else {
        c = a + h / 2;
        cout << "\n\nWarning: Bad iflag in riemann().\n";
        cout << "Proceeding with iflag = 2 (Midpoints).\n\n";
    }

    sum = (*f)(c);
    for ( int i = 2; i <= n; i++ ) {
        sum += (*f)( c += h );
```

```
    }
    return( h * sum );
}
```