1. A Gaussian quadrature formula for a weight function \( w(x) > 0 \) takes the form

\[
\int_{a}^{b} f(x) w(x) \, dx = \sum_{i=1}^{n} w_i f(x_i),
\]

and is exact for all polynomials \( f(x) \) of degree \( 2n-1 \) or less. Show that all the weights \( w_i \) are positive.

**Hint:** By considering some suitable test polynomials \( f(x) \), the result follows immediately.

2. Atkinson’s Example 3 on pages 262-263 concerns

\[
\int_{0}^{2\pi} e^{-x^2} dx \approx 7.9549265210128453.\]

One way to understand the extremely high rate of convergence of the trapezoidal rule (and, to a lesser extent, Simpson’s rule) for a periodic function such as this one starts by noting that the integrand can be Fourier expanded. In the present case

\[
e^{-x^2} = \sum_{n=0}^{\infty} \frac{\cos(nx)}{2^nn!}, \quad a_0 = \frac{1}{2\pi} \int_{0}^{2\pi} e^{-x^2} dx, \quad a_n = \frac{1}{\pi} \int_{0}^{2\pi} e^{-x^2} \cos(nx) dx, n > 0.
\]

By asymptotic analysis (the topic of APPM 5480), one can readily show that \( \lim_{n \to \infty} a_n 2^{-n} n! = 1 \). Correctly assuming that this limit result provides good approximations also at low values of \( n \), use this to derive approximations for the trapezoidal and Simpson errors when the original integral is discretized at \( x_i = 2\pi i / 8, i = 0,1,\ldots,8 \). Compare what you obtain against the values for this case that are quoted in Atkinson’s Table 5.7. Also tell how many nodes \( n \) you would need to obtain the very high accuracy of \( 10^{-60} \).

3. Gaussian quadrature is commonly used for accurate approximation of integrals. One useful generalization is to instead apply it to approximate infinite sums:

Determine the nodes \( x_1, x_2 \) and weights \( w_1, w_2 \) so that the formula

\[
\sum_{n=0}^{\infty} \frac{f(n)}{n!} = w_1 f(x_1) + w_2 f(x_2)
\]

becomes exact for polynomials \( f(x) \) of as high degree as possible.

**Hint:** Sums of the form \( \sum_{n=0}^{\infty} \frac{n^n}{n!} \) can be found analytically by considering derivatives of \( e^x, x e^x, \) etc. at \( x = 1 \).
4. The following code (adapted from Trefethen, L.N., Is Gauss quadrature better than Clenshaw-Curtis, SIAM Review 50(2008), 67-87) is a particularly slick implementation of the Golub and Welsch Gaussian quadrature algorithm, here given in the special case of \[ \int_{-1}^{1} f(x) dx, \] i.e. with weight function \( w(x) \equiv 1. \)

```matlab
function [x,w] = GQ(n) % (n+1)-pt Gauss quadrature of f
beta = .5./sqrt(1-(2*(1:n)).^(-2)); % 3-term recurrence coeffs
T = diag(beta,1) + diag(beta,-1); % Jacobi matrix
[V,D] = eig(T); % eigenvalue decomposition
x = diag(D); [x,i] = sort(x); % nodes (= Legendre points)
w = 2*V(1,i).^2; % weights
```

In the case of Clenshaw-Curtis quadrature (just interpolate at the ‘Chebyshev nodes’ and then integrate the interpolant), it is easier not to create nodes and weights separately, but instead evaluate the integral directly using an FFT. The following is a code for this task (also from the article mentioned above):

```matlab
function I = CC(f,n) % (n+1)-pt CC quadrature of f
x = cos(pi*(0:n)'/n); % Chebyshev points
fx = feval(f,x)/(2*n); % f evaluated at these points
g = real(fft(fx([1:n+1 n:-1:2]))); % Fast Fourier Transform
a = [g(1);g(2:n)+g(2*n:-1:n+2); g(n+1)]; % Chebyshev coefficients
w = 0*a'; w(1:2:end)=2./(1-(0:2:n).^2); % weight vector
I = w*a; % the integral
```

Test the two quadrature approached against each other by means of graphically displaying the errors with the two methods, as functions of \( n \), in the cases of (a) \( f(x) = e^x \), and (b) \( f(x) = 1/(1+16x^2) \). Tell what difference you observe between the results in the two cases.