This programming assignment is to write an adaptive quadrature routine to integrate a function \( f(x) \) from \( a \) to \( b \), to within a given tolerance \( tol \). Your program should ask for inputs \( a \) and \( b \), and the tolerance. It should output the estimated integral, the estimated error, and the number of function evaluations it took.

Your program should use either the trapezoid rule or Simpson’s rule, your choice. (The ambitious student should try both; if your program is modularly written, this should be a very small change). An error estimate is obtained by using the rule for a sub-interval \([x_i, x_{i+1}]\) the two-subintervals \([x_i, x_{i+1}/2] \) and \([x_{i+1}/2, x_{i+1}]\), and comparing the two. If the estimate exceeds the tolerance (weighted by the ratio of the sub-interval to the whole interval), then the interval should be divided in half, and each half will be integrated anew. Note that you might want to start with a minimum number of intervals, since starting from one interval might lead to premature termination.

You must write your program in a way that does not use more than the minimum number of function evaluations. If an interval if divided in half, you should not have to re-evaluate the function at the endpoints where it has already been calculated. There are several ways to do this, e.g. store all the function values for reuse, or a careful use of recursion, are two.

Apply your program on the following functions:

1. \( f(x) = \cos(a \cdot x^2) \) on \([0,1]\). Try \( a=1 \) and \( a=1000 \). (If your computer can’t handle \( a=1000 \), back off to \( a=100 \).)
2. \( f(x) = \sqrt{x} \) on \([0,1]\)
3. \( f(x) = \begin{cases} 1 & \text{for } x \in \left[\frac{1}{\sqrt{10}}, \frac{1}{\sqrt{10}}\right] \\ 0 & \text{otherwise} \end{cases} \) on the interval \([-1,1]\).

For each function, comment on the results. To help with this, your program should output a graph of the function \( f \), marking the \( x \) coordinates where the function was evaluated. This will give you an idea of the local stepsize used in each part of the interval.

Also to hand in: A description of how you tested your program. What robustness measures have you included?