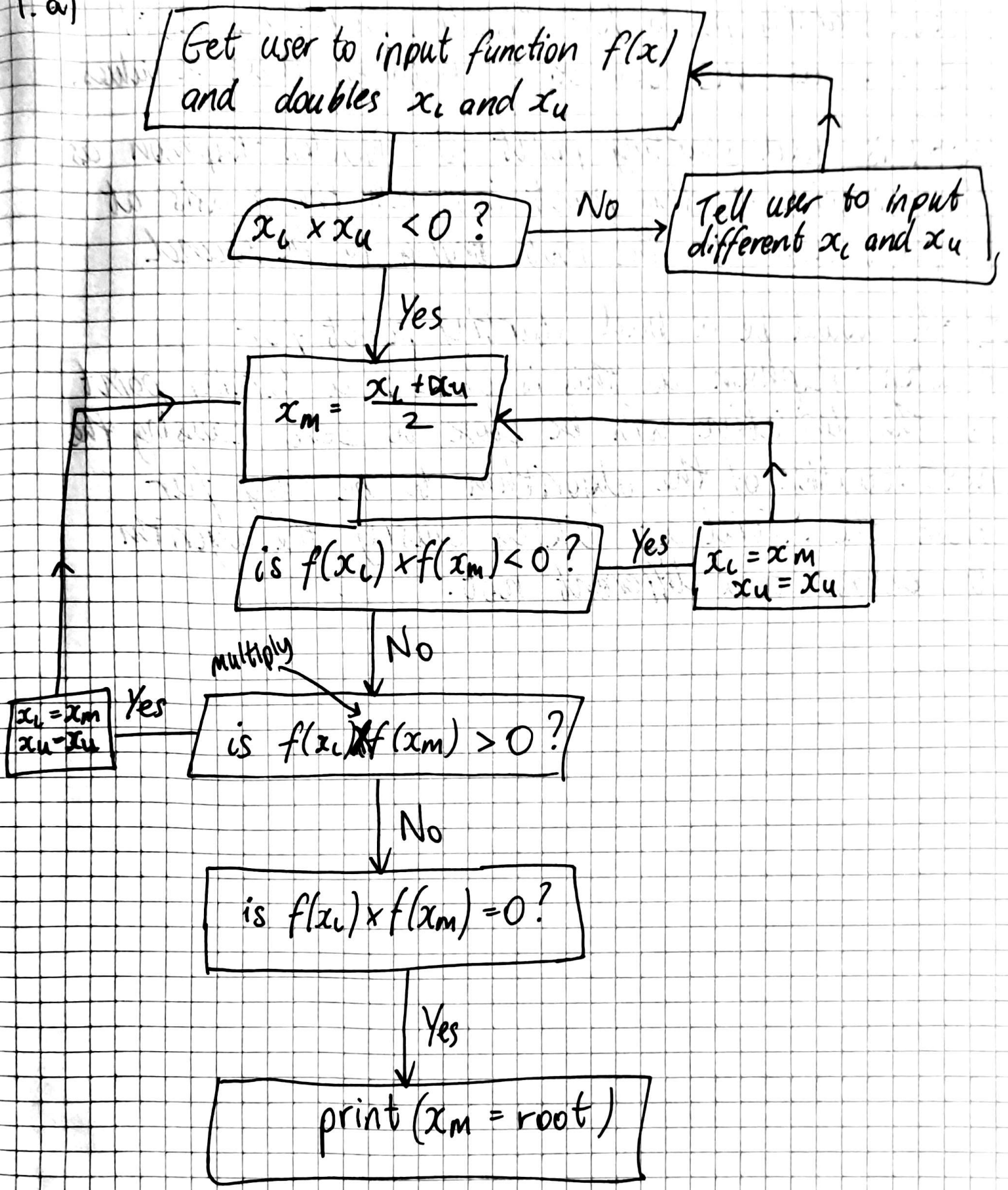
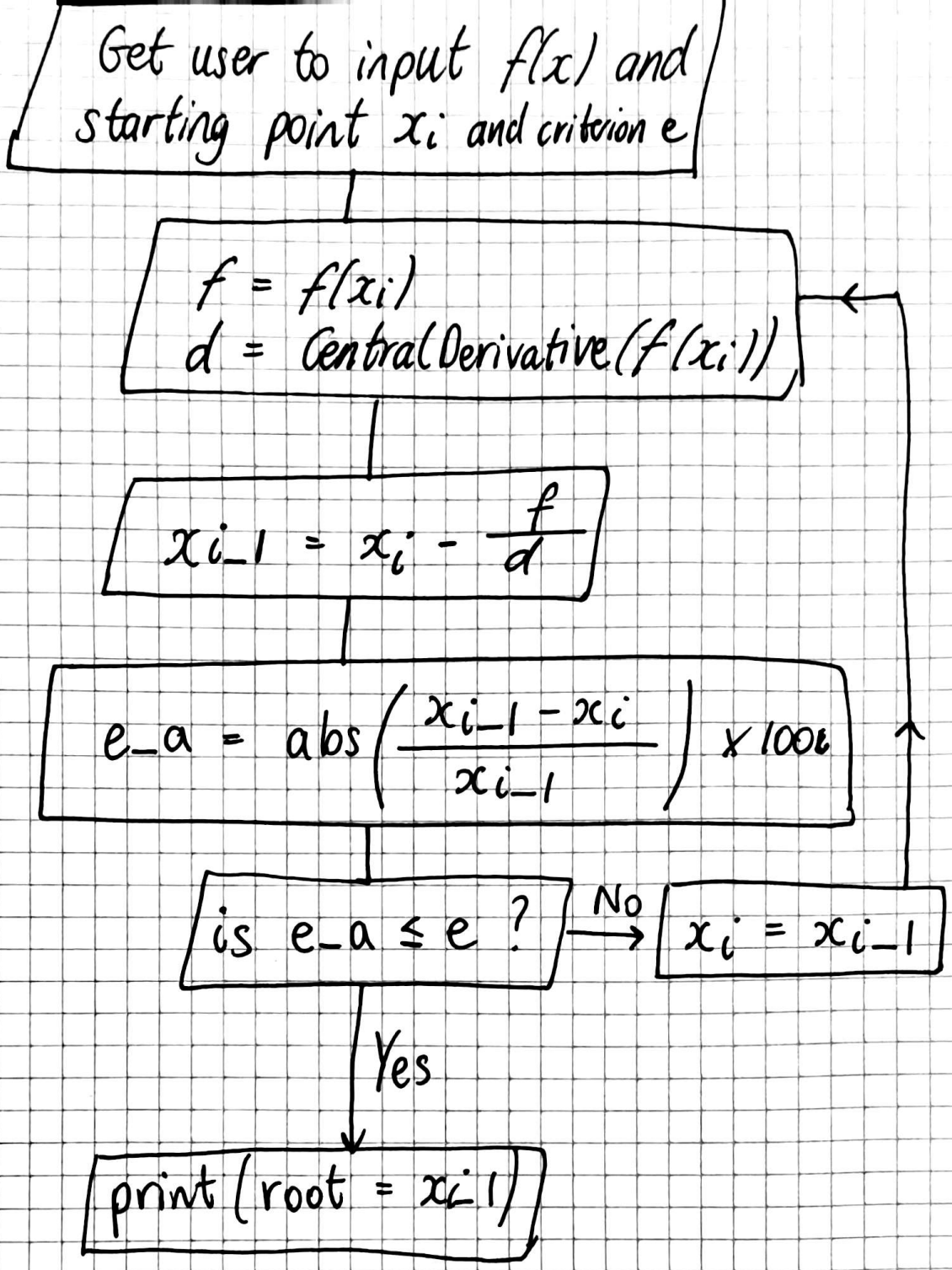
1. a)


Get user to input function f(x) and doubles $x_l$ and $x_u$
$x_l \times x_u < 0$ ?
No
Tell user to input different $x_l$ and $x_u$
Yes
$x_m = \frac{x_l + x_u}{2}$
is $f(x_l) \times f(x_m) < 0$ ?
Yes
$x_l = x_m$
$x_u = x_u$
No
multiply
$x_l = x_m$
$x_u = x_u$
Yes
is $f(x_l) \times f(x_m) > 0$ ?
No
is $f(x_l) \times f(x_m) = 0$ ?
Yes
print ($x_m$ = root)

1. a)


Get user to input f(x) and starting point $x_i$ and criterion e
$f = f(x_i)$
$d$ = CentralDerivative($f(x_i)$)
$x_{i\_1} = x_i - \frac{f}{d}$
$e\_a = abs\left(\frac{x_{i\_1} - x_i}{x_{i\_1}}\right) \times 100$
is $e\_a \leq e$ ?
No
$x_i = x_{i\_1}$
Yes
print (root = $x_{i\_1}$)

Good starting points for Bisection would be $x_l = 1$ $x_r = 2$ as the root exists between these $x$ values.

$x_i = 1$ is good starting point for Newton-Raphson as the derivative at this point intersects the $x$ axis at a point closer to the root trying to be found

$x_i = 4$ would be a bad starting point for Newton-Raphson as this is close to a turning point so the derivative will be close to zero causing the next iteration of the algorithm to be very far from the initial $x_i$ likley causing the algorithm to converge to a different root.

c) Discussion:

Newton-Raphson converges the fastest with secant taking just a few more iterations to reach the zero error tolerance level (as defined in the code). Bisection as expected takes quite a lot more iterations to find the root.