

```

> # Define your function
f := x → ex − 3 x − 5 :

# Initial endpoints: pick one where f>0, one where f<0
P := 3.1 : # because f(3.1) > 0
N := 1 : # because f(1) < 0

# Iteration loop
for n from 2 to 16 do
  x[n] := evalf( (P + N) / 2 ); # midpoint

  if evalf( f(x[n]) ) > 0 then
    P := x[n]; # replace the positive endpoint
  elif evalf( f(x[n]) ) < 0 then
    N := x[n]; # replace the negative endpoint
  else
    break;
  end if;
  print( X, n, "Midpoint =", x[n], "f(x) =", f(x[n]) );
end do:

```

```

X, 2, "Midpoint =", 2.050000000, "f(x) =", −3.382098894
X, 3, "Midpoint =", 2.575000000, "f(x) =", 0.406317160
X, 4, "Midpoint =", 2.312500000, "f(x) =", −1.837857770
X, 5, "Midpoint =", 2.443750000, "f(x) =", −0.815104580
X, 6, "Midpoint =", 2.509375000, "f(x) =", −0.230883120
X, 7, "Midpoint =", 2.542187500, "f(x) =", 0.080875610
X, 8, "Midpoint =", 2.525781250, "f(x) =", −0.076686160
X, 9, "Midpoint =", 2.533984375, "f(x) =", 0.001670665
X, 10, "Midpoint =", 2.529882813, "f(x) =", −0.037613319
X, 11, "Midpoint =", 2.531933594, "f(x) =", −0.017997782
X, 12, "Midpoint =", 2.532958985, "f(x) =", −0.008170175
X, 13, "Midpoint =", 2.533471680, "f(x) =", −0.003251410
X, 14, "Midpoint =", 2.533728028, "f(x) =", −0.000790784
X, 15, "Midpoint =", 2.533856202, "f(x) =", 0.000439844
X, 16, "Midpoint =", 2.533792115, "f(x) =", −0.000175495

```

(1)