```
> # HW - Bisection Method
  f(x) := exp(x) - 3 x - 5; x[0] := 2; x[1] := 3.2;
```

$$f := x \mapsto e^{x} - 3 \cdot x - 5$$

$$x_0 := 2$$

$$x_1 := 3.2 \quad \textbf{(1)}$$

```
> # Endpoints of shrinking interval containing (?) root
  a := x[0]; b := x[1];

   # Check IVP condition
  if evalf(f(a)·f(b)) > 0 then
     printf("Error: Intermediate Value Theorem not applicable here; f(a) and f(b) same sign");
    quit(1);
  end if
```

$$a := 2$$

$$b := 3.2 \quad \textbf{(2)}$$

```
> # LOOP: Check for roots on interval endpnts a,b. IF fail, bisect interval and store new endpnt in
      x[n]
  for n from 2 to 6 do
    if evalf(f(a)) = 0 then printf("%f is a root", a); quit(0);
    elif evalf(f(b)) = 0 then printf("%f is a root", b); quit(0);
    end if;

    # Store new interval endpnt, then update a,b
    x[n] := (a + b)/2;
    if evalf(f(a)·f(x[n])) < 0 then b := x[n]
    else a := x[n]   # EITHER f(x[n]) opp. sign to f(b) OR x[n] is a root
    end if;
  end do
```

$$x_2 := 2.600000000$$

$$x_3 := 2.300000000$$

$$x_4 := 2.450000000$$

$$x_5 := 2.525000000$$

$$x_6 := 2.562500000 \quad \textbf{(3)}$$

```
>
>
```