a.) Proof by using mathematical method

1. Let $f(x) = e^x - 2x - 2$.

(a) Show that $f(x) = 0$ has a root in $[-1, 0]$.

Proof: we have $f(x) = e^x - 2x - 2$

- $x = -1 \therefore f(-1) = e^{-1} - 2(-1) - 2 = \frac{1}{e} - 4 < 0$
- $x = 0 \therefore f(0) = e^0 - 2(0) - 2 = e - 2 > 0$

$\Rightarrow f(-1) \times f(0) = (\frac{1}{e} - 4)(e-2) < 0$, then by using intermediate value theorem, we have the function $f(x) = 0$ has a root in $[-1, 0]$

a.) Proof by using code implementation

In [1]:
```
1  import math as mt
2  def verify(a,b):
3    y1 = mt.exp(-a)-2*a-2
4    y2 = mt.exp(-b)-2*b-2
5    if(y1*y2<0):
6      print(f'f(x) = 0 has a root in the interval[{a,b}]')
7    else:
8      print(f'f(x) has not a root in the interval[{a,b}]')
9    return
10
11 if __name__ == "__main__":
12   verify(-1,0)
```

```
f(x) = 0 has a root in the interval[(-1, 0)]
```

(b) Perform 10 iterations using bisection method.

In [9]:

```python
import pandas as pd
def bisection(f, x0, x1, e):
    step = 1
    condition = True
    df = pd.DataFrame(data={'x':[x0],'f(x)':[f(x0)]}) #
    while condition:
        if step < 11:
            x2 = (x0 + x1) /2
            print(f'Step:{step}, p{step} = {x2:8.16f} and f(p{step}) = {f(x
            if f(x0) * f(x2) < 0:
                x1 = x2
            else:
                x0 = x2
            df.loc[step] = {'x':x0, 'f(x0)':f(x0)}  # use it to test afterw
            step = step + 1
            condition = abs(f(x2)) > e
        else:
            break

    print(f'\nRequired root is : {x2:0.16f}')
    return df
if __name__ == "__main__":
    import math as mt
    def f(x): return mt.exp(x)-2*x-2
    df = bisection(f=f,x0=-1,x1=0,e=1.0e-16)
```

```
Step:1, p1 = -0.5000000000000000 and f(p1) = -0.3934693402873666
Step:2, p2 = -0.7500000000000000 and f(p2) = -0.0276334472589852
Step:3, p3 = -0.8750000000000000 and f(p3) = 0.1668620196785082
Step:4, p4 = -0.8125000000000000 and f(p4) = 0.0687473100810800
Step:5, p5 = -0.7812500000000000 and f(p5) = 0.0203333617716144
Step:6, p6 = -0.7656250000000000 and f(p6) = -0.0037068118659436
Step:7, p7 = -0.7734375000000000 and f(p7) = 0.0082991933323440
Step:8, p8 = -0.7695312500000000 and f(p8) = 0.0022926565630828
Step:9, p9 = -0.7675781250000000 and f(p9) = -0.0007079629204707
Step:10, p10 = -0.7685546875000000 and f(p10) = 0.0007921257201233

Required root is : -0.7685546875000000
```

(c) Perform bisection iterations until $|f(p)| < 10^{-4}$.

In [18]:

```python
import pandas as pd
def bisection(f, x0, x1, e):
    step = 1
    condition = True
    df = pd.DataFrame(data={'x':[x0],'f(x)':[f(x0)]}) #
    while condition:
        x2 = (x0 + x1) /2
        if abs(f(x2))< 10**(-4):
            print(f'Step:{step}, p{step} = {x2:8.16f} and f(p{step}) = {f(x
        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2
        df.loc[step] = {'x':x0, 'f(x0)':f(x0)}  # use it to test afterward
        step = step + 1
        condition = abs(f(x2)) > e
    print(f'\nRequired root is : {x2:0.16f}')
    return df
if __name__ == "__main__":
    import math as mt
    def f(x): return mt.exp(x)-2*x-2
    df = bisection(f=f,x0=-1,x1=0,e=1.0e-16)
```

```
Step:11, p11 = -0.7680664062500000 and f(p11) = 0.0000420260975376
Step:14, p14 = -0.7680053710937500 and f(p14) = -0.0000517285795090
Step:15, p15 = -0.7680358886718750 and f(p15) = -0.0000048514570170
Step:16, p16 = -0.7680511474609375 and f(p16) = 0.0000185872662533
Step:17, p17 = -0.7680435180664062 and f(p17) = 0.0000068678911163
Step:18, p18 = -0.7680397033691406 and f(p18) = 0.0000010082136743
Step:19, p19 = -0.7680377960205078 and f(p19) = -0.0000019216225151
Step:20, p20 = -0.7680387496948242 and f(p20) = -0.0000004567046314
Step:21, p21 = -0.7680392265319824 and f(p21) = 0.0000002757544686
Step:22, p22 = -0.7680389881134033 and f(p22) = -0.0000000904750945
Step:23, p23 = -0.7680391073226929 and f(p23) = 0.0000000926396839
Step:24, p24 = -0.7680390477180481 and f(p24) = 0.0000000010822938
Step:25, p25 = -0.7680390179157257 and f(p25) = -0.0000000446964006
Step:26, p26 = -0.7680390328168869 and f(p26) = -0.0000000218070535
Step:27, p27 = -0.7680390402674675 and f(p27) = -0.0000000103623798
Step:28, p28 = -0.7680390439927578 and f(p28) = -0.0000000046400430
Step:29, p29 = -0.7680390458554029 and f(p29) = -0.0000000017788746
Step:30, p30 = -0.7680390467867255 and f(p30) = -0.0000000003482903
Step:31, p31 = -0.7680390472523868 and f(p31) = 0.0000000003670015
Step:32, p32 = -0.7680390470195562 and f(p32) = 0.0000000000093556
Step:33, p33 = -0.7680390469031408 and f(p33) = -0.0000000001694673
Step:34, p34 = -0.7680390469613485 and f(p34) = -0.0000000000800560
Step:35, p35 = -0.7680390469904523 and f(p35) = -0.0000000000353502
Step:36, p36 = -0.7680390470050042 and f(p36) = -0.0000000000129972
Step:37, p37 = -0.7680390470122802 and f(p37) = -0.0000000000018208
Step:38, p38 = -0.7680390470159182 and f(p38) = 0.0000000000037677
Step:39, p39 = -0.7680390470140992 and f(p39) = 0.0000000000009734
Step:40, p40 = -0.7680390470131897 and f(p40) = -0.0000000000004237
Step:41, p41 = -0.7680390470136444 and f(p41) = 0.0000000000002749
Step:42, p42 = -0.7680390470134171 and f(p42) = -0.0000000000000746
Step:43, p43 = -0.7680390470135308 and f(p43) = 0.0000000000001004
Step:44, p44 = -0.7680390470134739 and f(p44) = 0.0000000000000129
Step:45, p45 = -0.7680390470134455 and f(p45) = -0.0000000000000309
Step:46, p46 = -0.7680390470134597 and f(p46) = -0.0000000000000089
Step:47, p47 = -0.7680390470134668 and f(p47) = 0.0000000000000018
Step:48, p48 = -0.7680390470134633 and f(p48) = -0.0000000000000036
Step:49, p49 = -0.7680390470134650 and f(p49) = -0.0000000000000009
Step:50, p50 = -0.7680390470134659 and f(p50) = 0.0000000000000004
Step:51, p51 = -0.7680390470134655 and f(p51) = 0.0000000000000000

Required root is : -0.7680390470134655
```

(d) Perform bisection iterations until $|p - a| < 10^{-4}$.

In [21]:

```python
import pandas as pd
def bisection(f, x0, x1, e):
    step = 1
    condition = True
    df = pd.DataFrame(data={'x':[x0],'f(x)':[f(x0)]}) #
    while condition:
        x2 = (x0 + x1) /2
        if abs(x2-x0)< 10**(-4):
            print(f'Step:{step}, p{step} = {x2:8.16f} and f(p{step}) = {f(x
        if f(x0) * f(x2) < 0:
            x1 = x2
        else:
            x0 = x2
        df.loc[step] = {'x':x0, 'f(x0)':f(x0)}  # use it to test afterward
        step = step + 1
        condition = abs(f(x2)) > e
    print(f'\nRequired root is : {x2:0.16f}')
    return df
if __name__ == "__main__":
    import math as mt
    def f(x): return mt.exp(x)-2*x-2
    df = bisection(f=f,x0=-1,x1=0,e=1.0e-16)
```

```
Step:14, p14 = -0.7680053710937500 and f(p14) = -0.0000517285795090
Step:15, p15 = -0.7680358886718750 and f(p15) = -0.0000048514570170
Step:16, p16 = -0.7680511474609375 and f(p16) = 0.0000185872662533
Step:17, p17 = -0.7680435180664062 and f(p17) = 0.0000068678911163
Step:18, p18 = -0.7680397033691406 and f(p18) = 0.0000010082136743
Step:19, p19 = -0.7680377960205078 and f(p19) = -0.0000019216225151
Step:20, p20 = -0.7680387496948242 and f(p20) = -0.0000004567046314
Step:21, p21 = -0.7680392265319824 and f(p21) = 0.0000002757544686
Step:22, p22 = -0.7680389881134033 and f(p22) = -0.0000000904750945
Step:23, p23 = -0.7680391073226929 and f(p23) = 0.0000000926396839
Step:24, p24 = -0.7680390477180481 and f(p24) = 0.0000000010822938
Step:25, p25 = -0.7680390179157257 and f(p25) = -0.0000000446964006
Step:26, p26 = -0.7680390328168869 and f(p26) = -0.0000000218070535
Step:27, p27 = -0.7680390402674675 and f(p27) = -0.0000000103623798
Step:28, p28 = -0.7680390439927578 and f(p28) = -0.0000000046400430
Step:29, p29 = -0.7680390458554029 and f(p29) = -0.0000000017788746
Step:30, p30 = -0.7680390467867255 and f(p30) = -0.0000000003482903
Step:31, p31 = -0.7680390472523868 and f(p31) = 0.0000000003670015
Step:32, p32 = -0.7680390470195562 and f(p32) = 0.0000000000093556
Step:33, p33 = -0.7680390469031408 and f(p33) = -0.0000000001694673
Step:34, p34 = -0.7680390469613485 and f(p34) = -0.0000000000800560
Step:35, p35 = -0.7680390469904523 and f(p35) = -0.0000000000353502
Step:36, p36 = -0.7680390470050042 and f(p36) = -0.0000000000129972
Step:37, p37 = -0.7680390470122802 and f(p37) = -0.0000000000018208
Step:38, p38 = -0.7680390470159182 and f(p38) = 0.0000000000037677
Step:39, p39 = -0.7680390470140992 and f(p39) = 0.0000000000009734
Step:40, p40 = -0.7680390470131897 and f(p40) = -0.0000000000004237
Step:41, p41 = -0.7680390470136444 and f(p41) = 0.0000000000002749
Step:42, p42 = -0.7680390470134171 and f(p42) = -0.0000000000000746
Step:43, p43 = -0.7680390470135308 and f(p43) = 0.0000000000001004
Step:44, p44 = -0.7680390470134739 and f(p44) = 0.0000000000000129
Step:45, p45 = -0.7680390470134455 and f(p45) = -0.0000000000000309
Step:46, p46 = -0.7680390470134597 and f(p46) = -0.0000000000000089
Step:47, p47 = -0.7680390470134668 and f(p47) = 0.0000000000000018
Step:48, p48 = -0.7680390470134633 and f(p48) = -0.0000000000000036
Step:49, p49 = -0.7680390470134650 and f(p49) = -0.0000000000000009
Step:50, p50 = -0.7680390470134659 and f(p50) = 0.0000000000000004
Step:51, p51 = -0.7680390470134655 and f(p51) = 0.0000000000000000

Required root is : -0.7680390470134655
```