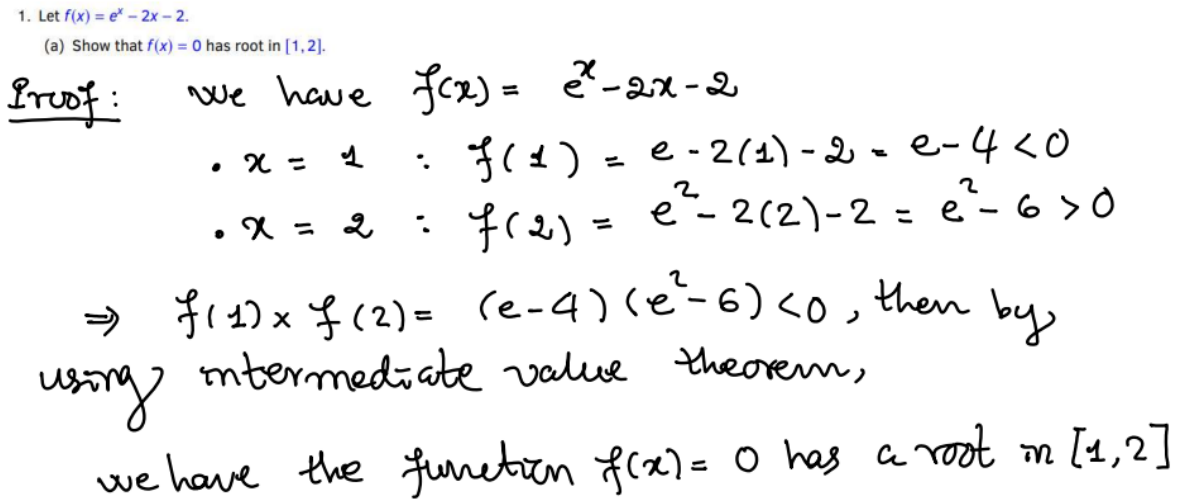
a.) Proof by using mathematical method



a.) Proof by using code implementation

In [24]:

```
import math as mt
def verify(a,b):
  y1 = mt.exp(-a)-2*a-2
  y2 = mt.exp(-b)-2*b-2
  if(y1*y2<0):
    print(f'f(x) = 0 has a root in the interval{a,b}')
  else:
    print(f'f(x) has no a root in the interval{a,b}')
  return

if __name__ == "__main__":
  verify(1,2)
```

```
f(x) has no a root in the interval(1, 2)
```

(b) Perform 10 iterations using Secant method.

In [25]:

```python
from math import pi,cos,exp
def SecantMethod(f,p0,p1,TOL=1e-10,N=100):
  print(f'i={0:2d}, x0={p0:0.16f}, f(x0)={f(p0):0.16f}')
  for i in range(1,N,1):
    p=p1-f(p1)*(p1-p0)/(f(p1)-f(p0))
    print(f'i={i+1:2d}, x{i}={p:0.16f}, f(x{i})={f(p):0.16f}')
    if(abs(p-p1)<TOL):
      return p
    p0 = p1
    p1 = p
  return None

if __name__ == "__main__":
  def f(x): return exp(x)-2*x-2
  p0 = 1
  p1= 2
  p=SecantMethod(f,p0,p1,TOL=1e-16)
  print('The request root is : ',p)
```

```
i= 0, x0=1.0000000000000000, f(x0)=-1.2817181715409549
i= 2, x1=1.4799050918348966, f(x1)=-0.5672814093807768
i= 3, x2=1.6307176031026089, f(x2)=-0.1538966168898508
i= 4, x3=1.6868627146073407, f(x3)=0.0287794611761112
i= 5, x4=1.6780174072530005, f(x4)=-0.0011060175757591
i= 6, x5=1.6783447590574712, f(x5)=-0.0000074886339503
i= 7, x6=1.6783469906034445, f(x6)=0.0000000019696538
i= 8, x7=1.6783469900166597, f(x7)=-0.0000000000000036
i= 9, x8=1.6783469900166608, f(x8)=0.0000000000000004
i=10, x9=1.6783469900166605, f(x9)=0.0000000000000000
i=11, x10=1.6783469900166605, f(x10)=0.0000000000000000
The request root is :  1.6783469900166605
```

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

In [26]:

```python
from math import pi,cos,exp
import math as mt
def SecantMethod(f,p0,p1,TOL=1e-10,N=100):
  print(f'i={0:2d}, x0={p0:0.16f}, f(x0)={f(p0):0.16f}')
  for i in range(1,N,1):
    p=p1-f(p1)*(p1-p0)/(f(p1)-f(p0))
    if abs(f(p))<mt.pow(10,-4):
      print(f'i={i+1:2d}, x{i}={p:0.16f}, f(x{i})={f(p):0.16f}')
    if(abs(p-p1)<TOL):
      return p
    p0 = p1
    p1 = p
  return None

if __name__ == "__main__":
  def f(x): return exp(x)-2*x-2
  p0 = 1
  p1 = 2
  p=SecantMethod(f,p0,p1,TOL=1e-16)
  print('The request root is : ',p)
```

```
i= 0, x0=1.0000000000000000, f(x0)=-1.2817181715409549
i= 6, x5=1.6783447590574712, f(x5)=-0.0000074886339503
i= 7, x6=1.6783469906034445, f(x6)=0.0000000019696538
i= 8, x7=1.6783469900166597, f(x7)=-0.0000000000000036
i= 9, x8=1.6783469900166608, f(x8)=0.0000000000000004
i=10, x9=1.6783469900166605, f(x9)=0.0000000000000000
i=11, x10=1.6783469900166605, f(x10)=0.0000000000000000
The request root is :  1.6783469900166605
```

(d) Perform Secant iterations until $|x - x_1| < 10^{-4}$.

In [27]:

```python
from math import pi,cos,exp
def SecantMethod(f,p0,p1,TOL=1e-10,N=100):
  print(f'i={0:2d},p0={p0:0.16f},f(p0)={f(p0):0.16f}')
  for i in range(1,N,1):
    p=p1-f(p1)*(p1-p0)/(f(p1)-f(p0))
    if abs(p-p0)<mt.pow(10,-4):
      print(f'i={i+1:2d},p{i}={p:0.16f},f(p{i})={f(p):0.16f}')
    if(abs(p-p1)<TOL):
      return p
    p0 = p1
    p1 = p
  return None

if __name__ == "__main__":
  def f(x): return exp(x)-2*x-2
  p0 = 1
  p1= 2
  p=SecantMethod(f,p0,p1,TOL=1e-16)
  print('The request root is : ',p)
```

```
i= 0,p0=1.0000000000000000,f(p0)=-1.2817181715409549
i= 8,p7=1.6783469900166597,f(p7)=-0.0000000000000036
i= 9,p8=1.6783469900166608,f(p8)=0.0000000000000004
i=10,p9=1.6783469900166605,f(p9)=0.0000000000000000
i=11,p10=1.6783469900166605,f(p10)=0.0000000000000000
The request root is :  1.6783469900166605
```