Exercício 5

Considere $f(x) = tan(\pi x) - 6$ Determine um intervalo que contenha um zero de f e utilize os métodos a seguir, para aproximar uma raiz no intervalo que você determinou,

(a) Método da Bisseção

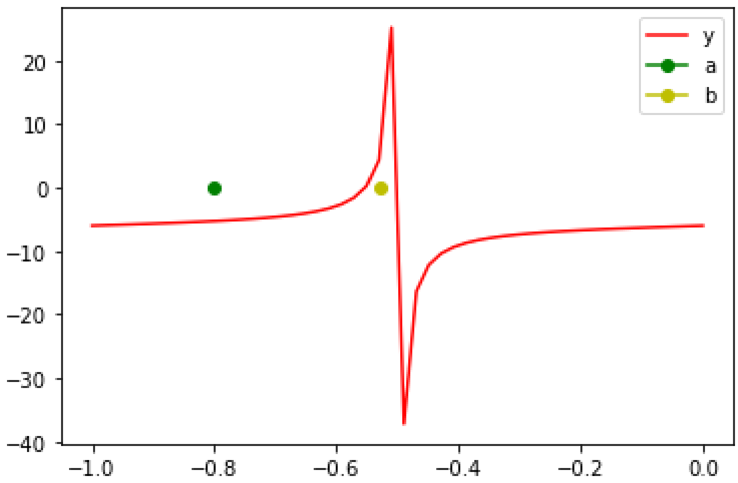
(b) Método da Falsa Posição

(c) Método de Newton

(d) Método Secante

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import math

xx = np.linspace(-1,0,50)
plt.figure(1)
y= np.tan(np.pi*xx) -6
plt.plot(xx,y,'-r',label = 'y')
plt.plot(-0.8,0,'-g',label = 'a',marker = 'o')
plt.plot(-0.53,0,'-y',label = 'b',marker = 'o')
plt.legend()
```

Out[1]: <matplotlib.legend.Legend at 0x1df6f7c4df0>



In [2]:
```
def fx(x):
    return math.tan(math.pi*x) -6
```

In [3]:
```
print(fx(-0.8)*fx(-0.53))
```

```
-24.146608016453754
```

In [4]:
```
a = -0.8
b = -0.53
precisao = 10**-5
k = (math.log(b-a,10) - math.log(precisao,10)) / math.log(2,10)
k
```

Out[4]: 14.72067178682556

O intervalo escolhido foi [-0.8;-0.53]

Resposta (a) Método da bisseção:

In [5]:

```
x = a
k = 14
for i in range(1,k):
    x=a+((b-a)/2)

    if fx(a)*fx(x) < 0:
        b=x
    else:
        a=x
print("x= ", x,"f(x) = ",fx(x))
```

```
x=  -0.5525769042968751 f(x) =  -0.0009817819663000549
```

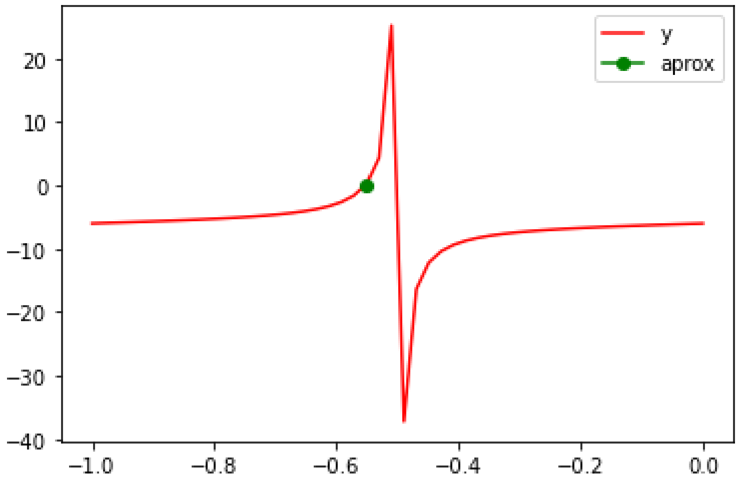
O resultado encontrado foi de $x \approx -0.55257$ para $y \approx 0.00093$

In [6]:

```
xx = np.linspace(-1,-0,50)
plt.figure(1)
y= np.tan(np.pi*xx) -6
plt.plot(xx,y,'-r',label = 'y')
plt.plot(-0.55256,0.00093,'-g',label = 'aprox',marker = 'o')
plt.legend()
```

Out[6]:

```
<matplotlib.legend.Legend at 0x1df6ff5f610>
```



Resposta (b) Método da falsa posição:

In [7]:

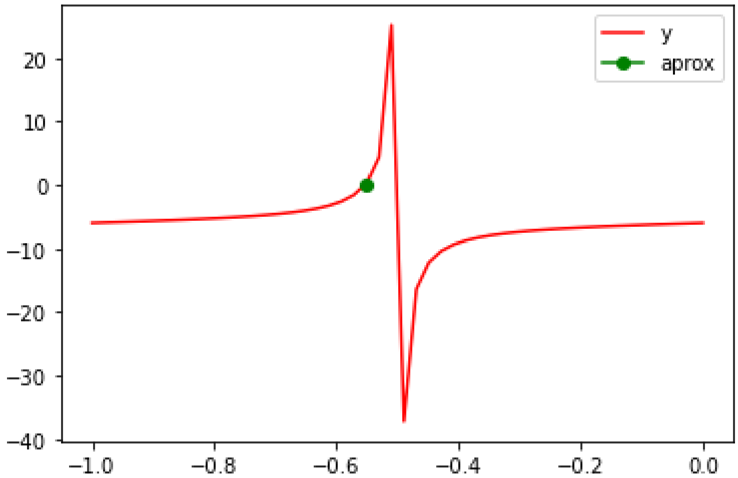
```
a = -0.8
b = -0.53
precisao = 10**-5
x = a
k = 15

for i in range(1,k):

    x= a - (b-a)*fx(a)/(fx(b)-fx(a))

    if fx(a)*fx(x) < 0:
        b=x
    else:
        a=x
print("x= ", x, "f(x) = ",fx(x))
```

```
x=  -0.5525700463205683 f(x) =  -0.00018476894667696087
```

In [8]:

```
xx = np.linspace(-1,-0,50)
plt.figure(1)
y= np.tan(np.pi*xx) -6
plt.plot(xx,y,'-r',label = 'y')
plt.plot(-0.55257,-0.00018,'-g',label = 'aprox',marker = 'o')
plt.legend()
```

Out[8]: <matplotlib.legend.Legend at 0x1df6ffd7700>



O resultado encontrado foi de $x \approx -0.55257$ para $y \approx -0.00018$

Resposta (c) Método de Newton:

In [9]:

```
import  sympy as sp
_a,_x,_y = sp.symbols('_a _x _y')
```

In [10]:

```
fxLinha = sp.diff(sp.tan(sp.pi*_x) -6)
print(fxLinha)
fxLinha
```

```
pi*(tan(pi*_x)**2 + 1)
```

Out[10]: $\pi\left(\tan^2\left(\pi_x\right) + 1\right)$

In [11]:

```
def dFx(x):
    return math.pi*(math.tan(math.pi*x)**2 + 1)
```

In [12]:

```
x =  -0.53
k = 15

for i in range(1,k):
    x = x-fx(x)/dFx(x)

print("x= ", x, "f(x) = ",fx(x))
```

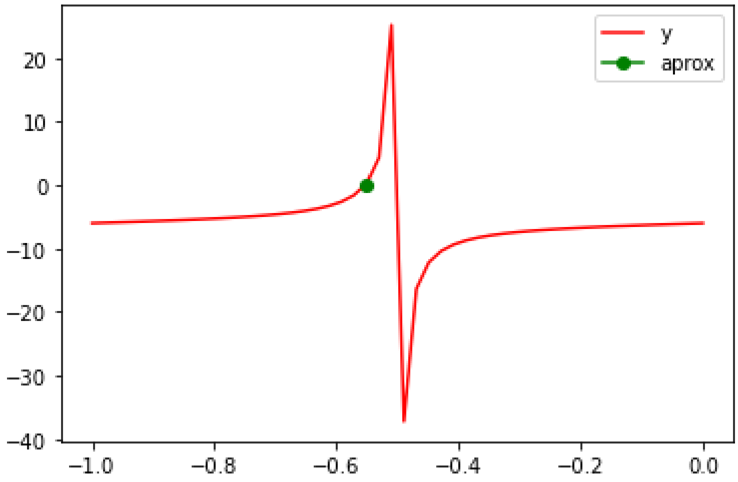
```
x=  -0.552568456711253 4 f(x) =  0.0
```

In [13]:

```
xx = np.linspace(-1,-0,50)
plt.figure(1)
y= np.tan(np.pi*xx) -6
plt.plot(xx,y,'-r',label = 'y')
```

```
plt.plot(-0.55257,0,'-g',label = 'aprox',marker = 'o')
plt.legend()
```

Out[13]: <matplotlib.legend.Legend at 0x1df714b8df0>



O resultado encontrado foi de $x \approx -0.55257$ para $y \approx -0.00018$

Resposta (d) Método Secante:

In [14]:

```
xAnt0 = -0.53
xAnt1 = -0.535

precisao = 10**-5

while True:
    xAtual = xAnt1 - (xAnt0-xAnt1)*fx(xAnt1)/(fx(xAnt0)-fx(xAnt1))

    Er = abs((xAtual-xAnt1)/xAtual)
    if Er < precisao:
        break
    xAnt0,xAnt1 = xAnt1,xAtual

print("x= ", xAtual, "f(x) = ",fx(xAtual))
```

```
x=  -0.5525684566760563 f(x) =  4.091286420759843e-09
```

O resultado encontrado foi de $x \approx -0.55257$ para $y \approx 0.00000$