Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
resolution = 0.0001
x = np.arange(-np.pi,np.pi,resolution)  # -pi to pi with the interval of 0.0001

square = np.array(x)
square[range(x.size)] = 0
square[range(int(x.size/2))] = 1
square[range(int(x.size/2), int(x.size))]= 0

np.trapz(square,x) # integration of f(x)
a0 = np.trapz(square,x)
a0 /np.pi # dividing by pi which is present out side the integration

n=1
harm = np.sin(n*x)
mult1 = square*harm
fund = np.trapz(mult1,x)
a_1 = np.trapz(mult1,x)
a_1/np.pi

n=3
harm = np.sin(n*x)
mult2 = square*harm
third = np.trapz(mult2,x)
np.trapz(mult2,x)
a_3 = np.trapz(mult2,x)
a_3/np.pi

20*np.log10(abs(third/fund)) # 3rd Harmonic Distortion
plt.subplot(311)
plt.plot(x,square)
plt.xlabel('(x)')
plt.ylabel('f(x)')
plt.title('SIGNAL', fontsize=18)

plt.subplot(312)
plt.plot(x,mult1)
plt.plot(x,square)
plt.xlabel('(x)')
plt.ylabel('sin(1*x)*f(x)')
plt.subplot(313)
plt.plot(x,mult2)
plt.plot(x,square)
plt.xlabel('(x)')
plt.ylabel('sin(3*x)*f(x)')
plt.show()
```

Output:

```
>>> np.trapz(square,x) # integration of f(x)

3.1415500000066299

>>> a0 /np.pi

0.99998642294279794

>>> np.trapz(mult1,x)

-1.9999999986736692

>>> a_1/np.pi

-0.63661977194539721

>>> np.trapz(mult2,x)

-0.66666666268767438

>>> a_3/np.pi

-0.21220658952264121

>>> 20*np.log10(abs(third/fund))

-9.5424251404747
```