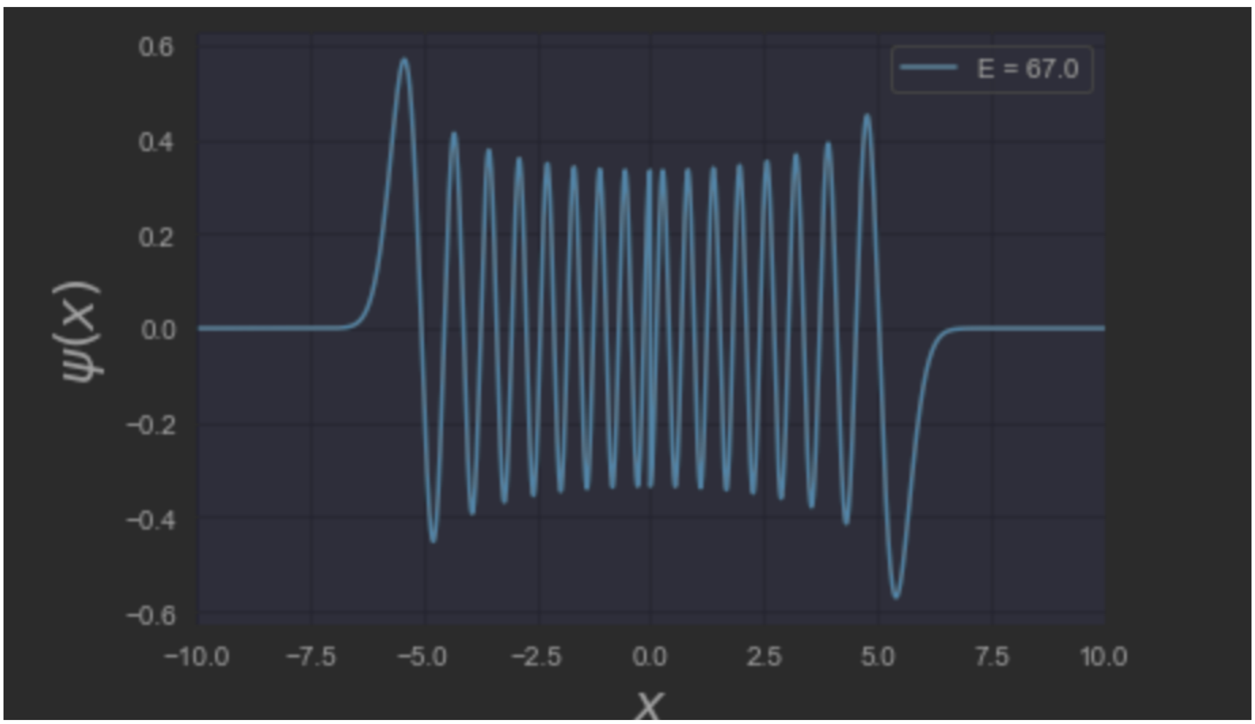
1.
    a. Bound state energies for omega = 1 are [1.5, 3.5, 5.5, 7.5, 9.5, 11.5, 13.5, 15.5, 17.5, 19.5, 21.5, 23.5, 25.5, 27.5, 29.5, 31.5, 33.5, 35.5, 37.5, 39.5, 41.5, 43.5, 45.5, 47.5, 49.5, 51.5, 53.5, 55.5, 57.5, 59.5, 61.5, 63.5, 65.5, 67.5, 69.5]
    b. What do we expect?
        i. We expect them to follow a relationship similar to E_n = hw(n+0.5)
        ii. Checking for odd case, there is good agreement (see above)
        iii. For even case we have; [0.5, 2.5, 4.5 …] again checks out with expected equation derived results.
2. With w = 2, we then have E_n = 2n + 1, we can check that our outputs line up…
    a. Odd:
        i. bound state energies for omega = 2 are [3.0, 7.0, 11.0, 15.0, 19.0, 23.0, 27.0, 31.0, 35.0, 39.0, 43.0, 47.0, 51.0, 55.0, 59.0, 63.0, 67.0]
    b. Even:
        i. bound state energies for omega = 2 are [1.0, 5.0, 9.0, 13.0, 17.0, 21.0, 25.0, 29.0, 33.0, 37.0, 41.0, 45.0, 49.0, 53.0, 57.0, 61.0, 65.0]

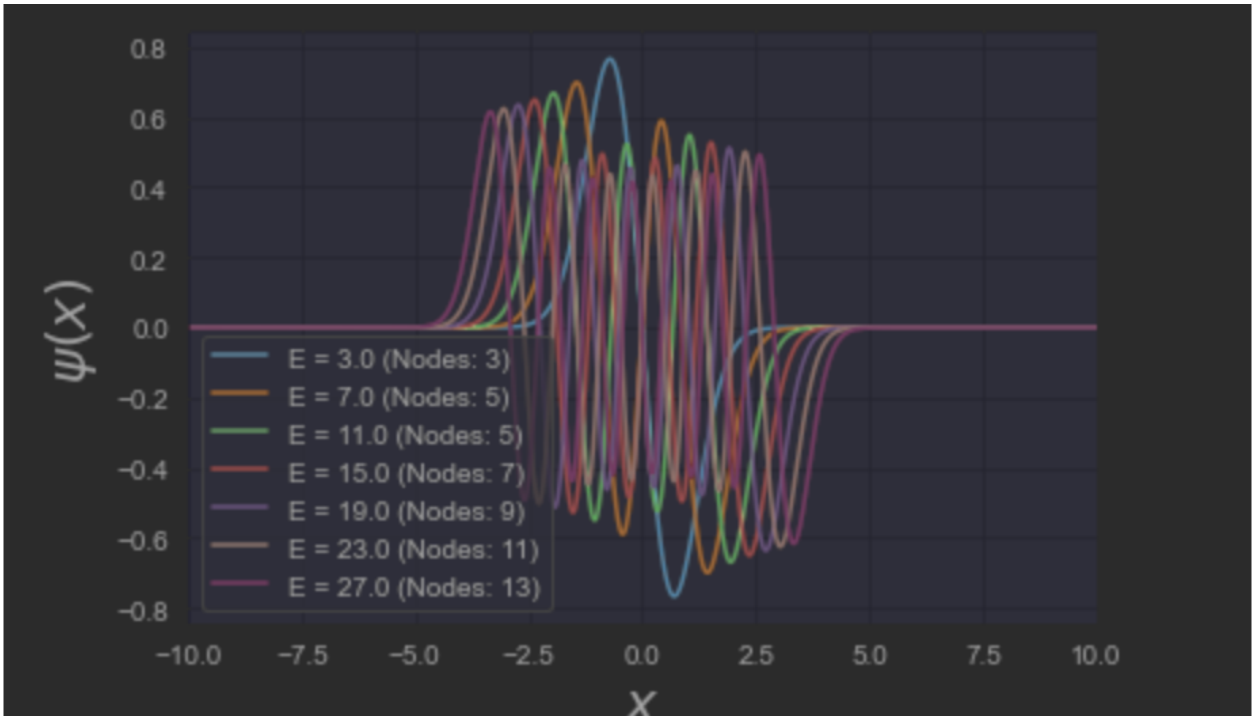
3. Considering the 17th odd state with w = 2, how many zeros does this function have?



We can take smaller magnitude examples to try and build a general model for the relationship between the function number and the corresponding number of 0's.

We can visual inspect these lower order functions to draw the conclusion that the relationship between the number of nodes and the function number is 2n + 1.

This is demonstrated with the small logic function below which shows this relationship graphically.

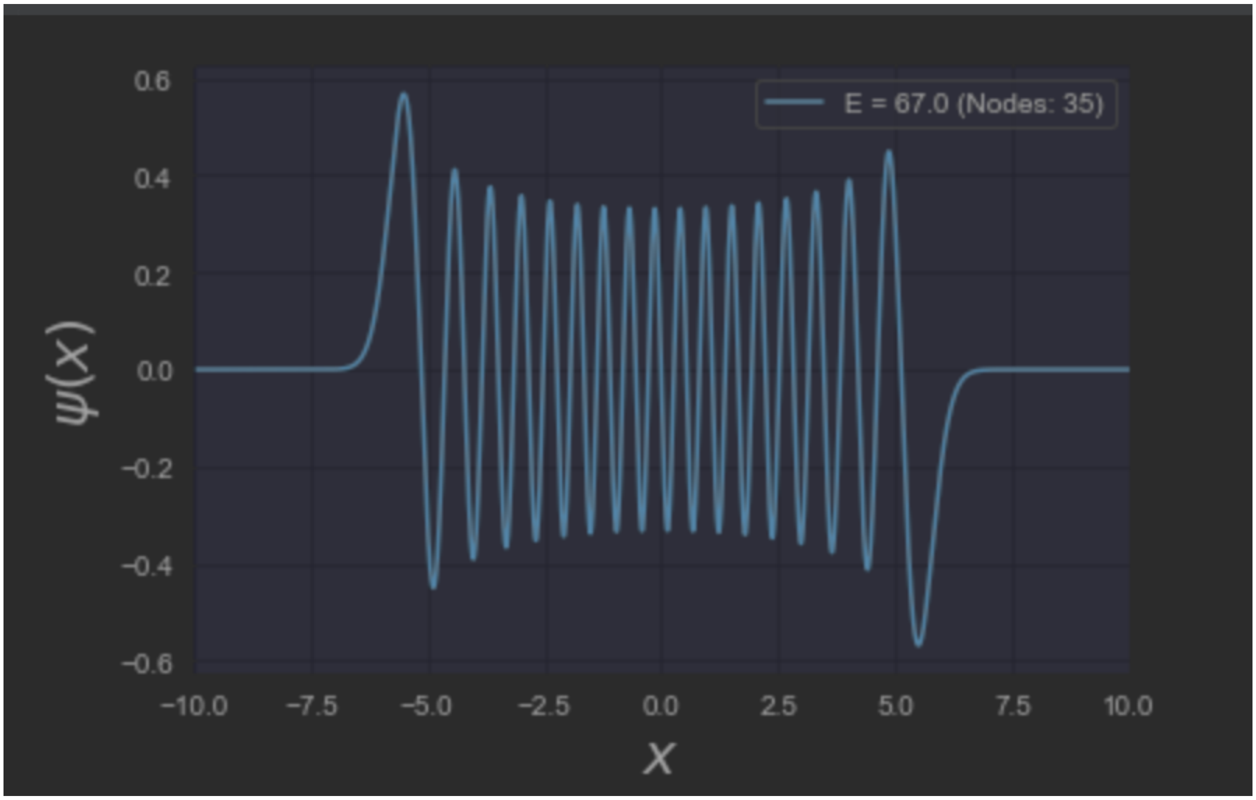


```python
def count_zero_crossings(data):
    zero_crossings = 0
    for i in range(len(data)-1):
        if data[i] * data[i+1] < 0:
            zero_crossings += 1
    return zero_crossings

x = linspace(-b, 0, N)
xall =linspace(-b,b,2*N-1)
h=b/(N-1)
for E in E_zeroes[0:7]:
    Wave_function(E)
    psil = psi[:,0]
    psir = -psil[N:0:-1]
    psilr = np.append(psil,psir)
    psilr=psilr/sqrt(h*np.sum(np.square(psilr)))
    # plt.plot(xall, psilr, label="E = " + str(E))
    #check how many times each wavefunction crosses zero
    zero_crossings = count_zero_crossings(psilr)
    plt.plot(xall, psilr, label="E = " + str(E) + f" (Nodes: {zero_crossings})")


plt.xlabel(r'$x$',size=20)
plt.ylabel(r'$\psi(x)$',size=20)
plt.legend()
plt.xlim(-10,10)
```

As such it is possible to verify that the function with excited state number = 17 should have 2(17)+1 nodes, i.e. 35 nodes.



Further, we can derive the classical turning points and probabilities as the following.

Classical turning point: 5.787918451395113
Probability outside well: 0.041309286870913224

All relevant code can be found in the following github repository.
https://github.com/udiram/physics3mm3_mcmaster/tree/master/udi