In [ ]:
```
# Write a python program for a fixed t, in the interval (-3 π to 3 π) for any rar
X(t) = A cos(wt) + B sin(wt)
```

In [ ]:
```
# A = 5 7 420 69 45
# B = 3 4 88  18 77
# w = 1400 2000 3000 9999 21000
```

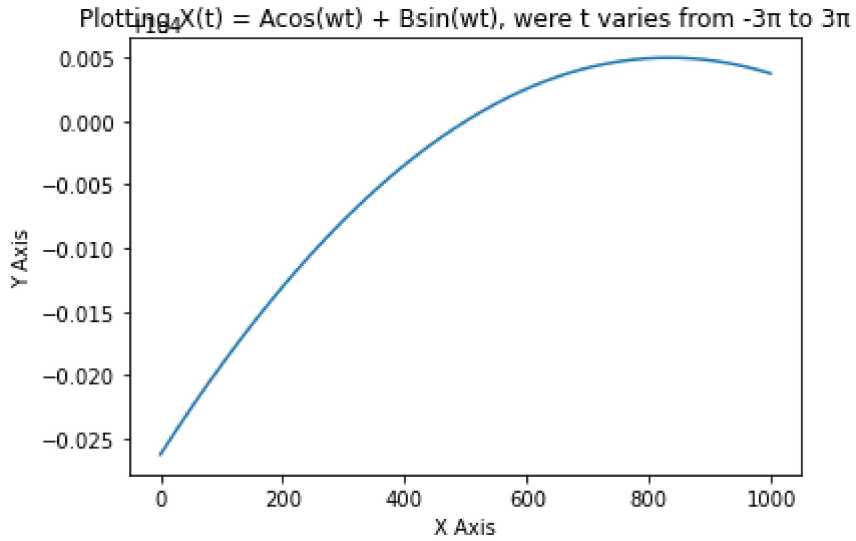
In [75]:
```
# Import Libraries
import numpy as np
import matplotlib.pyplot as plt
import math
import cmath
from sympy import *
```

In [76]:
```
"""
Target for today:
1. Plotting a graph for X(t) = Acos(wt) + Bsin(wt), where  varies from -3π to 3π
"""
```

Out[76]:
```
'\nTarget for today:\n1. Plotting a graph for X(t) = Acos(wt) + Bsin(wt), where
varies from -3π to 3π using matplotlib and numpy\n'
```

In [78]:
```
# 1. Plotting a graph for X(t) = Acos(wt) + Bsin(wt), where  varies from -3π to 3

t = np.linspace(-3 * np.pi, 3 * np.pi, 1000) # Plotting the graph for t = sin(t)

# from -3π to 3π; 1000 denotes Linearly spaced numbers

run = True
while(run):
    f = int(input("Enter the value for Frequency : "))
    A = int(input("Enter the value for A : "))
    B = int(input("Enter the value for B : "))

    w = 1/(2 * np.pi * f)

    plot = A * np.cos(w * t) + B * np.sin(w * t)

    plt.title("Plotting X(t) = Acos(wt) + Bsin(wt), were t varies from -3π to 3π'
    plt.xlabel("X Axis")
    plt.ylabel("Y Axis")

    plt.plot(plot)
    plt.show()
    run = int(input("Enter 1 for continuing, 0 to quit : "))
```

```
Enter the value for Frequency : 1000
Enter the value for A : 10000
Enter the value for B : 10
```




```
Enter 1 for continuing, 0 to quit : 0
```

In [79]:
```
print(t)
```

```
-0.82077646 -0.80190803 -0.78303961 -0.76417119 -0.74530276 -0.72643434
-0.70756591 -0.68869749 -0.66982906 -0.65096064 -0.63209222 -0.61322379
-0.59435537 -0.57548694 -0.55661852 -0.53775009 -0.51888167 -0.50001325
-0.48114482 -0.4622764  -0.44340797 -0.42453955 -0.40567112 -0.3868027
-0.36793427 -0.34906585 -0.33019743 -0.311329   -0.29246058 -0.27359215
-0.25472373 -0.2358553  -0.21698688 -0.19811846 -0.17925003 -0.16038161
-0.14151318 -0.12264476 -0.10377633 -0.08490791 -0.06603949 -0.04717106
-0.02830264 -0.00943421  0.00943421  0.02830264  0.04717106  0.06603949
 0.08490791  0.10377633  0.12264476  0.14151318  0.16038161  0.17925003
 0.19811846  0.21698688  0.2358553   0.25472373  0.27359215  0.29246058
 0.311329    0.33019743  0.34906585  0.36793427  0.3868027   0.40567112
 0.42453955  0.44340797  0.4622764   0.48114482  0.50001325  0.51888167
 0.53775009  0.55661852  0.57548694  0.59435537  0.61322379  0.63209222
 0.65096064  0.66982906  0.68869749  0.70756591  0.72643434  0.74530276
 0.76417119  0.78303961  0.80190803  0.82077646  0.83964488  0.85851331
 0.87738173  0.89625016  0.91511858  0.93398701  0.95285543  0.97172385
 0.99059228  1.0094607   1.02832913  1.04719755  1.06606598  1.0849344
 1.10380282  1.12267125  1.14153967  1.1604081   1.17927652  1.19814495
 1.21701337  1.23588179  1.25475022  1.27361864  1.29248707  1.31135549
 1.33022392  1.34909234  1.36796077  1.38682919  1.40569761  1.42456604
```

In [80]:
```
print(plot)
```

```
 9999.99860837  9999.99864228  9999.9986761   9999.99870982
 9999.99874346  9999.99877701  9999.99881047  9999.99884383
 9999.99887711  9999.9989103   9999.99894339  9999.9989764
 9999.99900931  9999.99904214  9999.99907487  9999.99910752
 9999.99914007  9999.99917254  9999.99920491  9999.9992372
 9999.99926939  9999.9993015   9999.99933351  9999.99936544
 9999.99939727  9999.99942901  9999.99946067  9999.99949223
 9999.9995237   9999.99955508  9999.99958638  9999.99961758
 9999.99964869  9999.99967971  9999.99971065  9999.99974149
 9999.99977224  9999.9998029   9999.99983347  9999.99986395
 9999.99989434  9999.99992464  9999.99995485  9999.99998497
10000.000015   10000.00004494 10000.00007479 10000.00010455
10000.00013422 10000.0001638  10000.00019329 10000.00022269
10000.000252   10000.00028122 10000.00031034 10000.00033938
10000.00036833 10000.00039719 10000.00042596 10000.00045463
10000.00048322 10000.00051172 10000.00054012 10000.00056844
10000.00059667 10000.0006248  10000.00065285 10000.0006808
10000.00070867 10000.00073645 10000.00076413 10000.00079173
10000.00081923 10000.00084665 10000.00087397 10000.00090121
```

In [32]:
```
# Integrate
# 1. f(x) = x^2 from -2 to 5
# 2. f(x) = x^2 * e^-x from 0 to ∞
```

In [81]:
```
x, y = symbols('x y')
integrate(x ** 2, (x, -2, 5)) # Limits of Integral of x from 0 to 5
```

Out[81]:
$$\frac{133}{3}$$

In [82]:
```
integrate(x ** 2 * exp(-x), (x, 0, np.inf)) # Limits of Integral of x from 0 to 5
```

Out[82]: 2

In [83]:
```
# Fourier Transform and Inverse Fourier Transform
s, t, w = symbols('s t w')

"""f(x) =
        1, |x| < 1
        0, otherwise
"""
```

Out[83]: 'f(x) = \n   1, |x| < 1\n   0, otherwise\n'

In [84]:
```
fourier_transform(abs(t), t, w)
```

Out[84]: $\mathcal{F}_t\left[|t|\right](w)$

In [55]:
```
print(fourier_transform)
```

```
<function fourier_transform at 0x0000018AEB7158B0>
```

In [58]:
```
# f = -2π * |t|
fourier_transform(-2 * np.pi * abs(t), t, w)
```

Out[58]: $-6.28318530717959\mathcal{F}_t\left[|t|\right](w)$

In [72]:
```
j = sqrt(-1)
1/sqrt(2*np.pi) * integrate(1 * exp(-j * w * x), (x, -1, 1))
```

Out[72]: $$0.398942280401433\left(\begin{cases} -\frac{ie^{iw}}{w} + \frac{ie^{-iw}}{w} & \text{for } w > -\infty \wedge w < \infty \wedge w \neq 0 \\ 2 & \text{otherwise} \end{cases}\right)$$

In [ ]:
```
# End of the Practical Session
```