Homework 1 Writeup

Christian Diangco
AMATH 301 B

In [7]:
```
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
```

Problem 1

In [8]:
```
# Problem 1
y1,y2,y3,y4 = [0,0,0,0]
for i in np.arange(1,100000001):
    if i <= 100000:
        y1 += 0.1
    y2 += 0.1
    y3 += 0.25
    y4 += 0.5
```

In [9]:
```
x1 = abs(10000 - y1)
x2 = abs(y2 - 10000000)
x3 = abs(25000000 - y3)
x4 = abs(y4 - 50000000)
print("x1 =", x1)
print("x2 =", x2)
print("x3 =", x3)
print("x4 = ", x4)
```

```
x1 = 1.8848368199542165e-08
x2 = 0.018870549276471138
x3 = 0.0
x4 =  0.0
```

The values computed for `x1, x2, x3, x4` are displayed above.
The value computed for `x2` is *larger* than the value computed for `x1`. `x2` is larger because the error when computing for `y2` is larger than that of `y1`. Both `x3` and `x4` are exactly 0 while `x1` and `x2` are not. Notably, both `x1`'s and `x2`'s corresponding `y` variables were computed using 0.1.
I think that only some are equal to 0 because some numbers are easier for the computer to work with. I know that computers store numbers in binary, so I'm guessing that it's easier to store 0.25 and 0.5 in binary than 0.1 i.e. 0.1 doesn't have an exact binary counterpart.

Problem 2

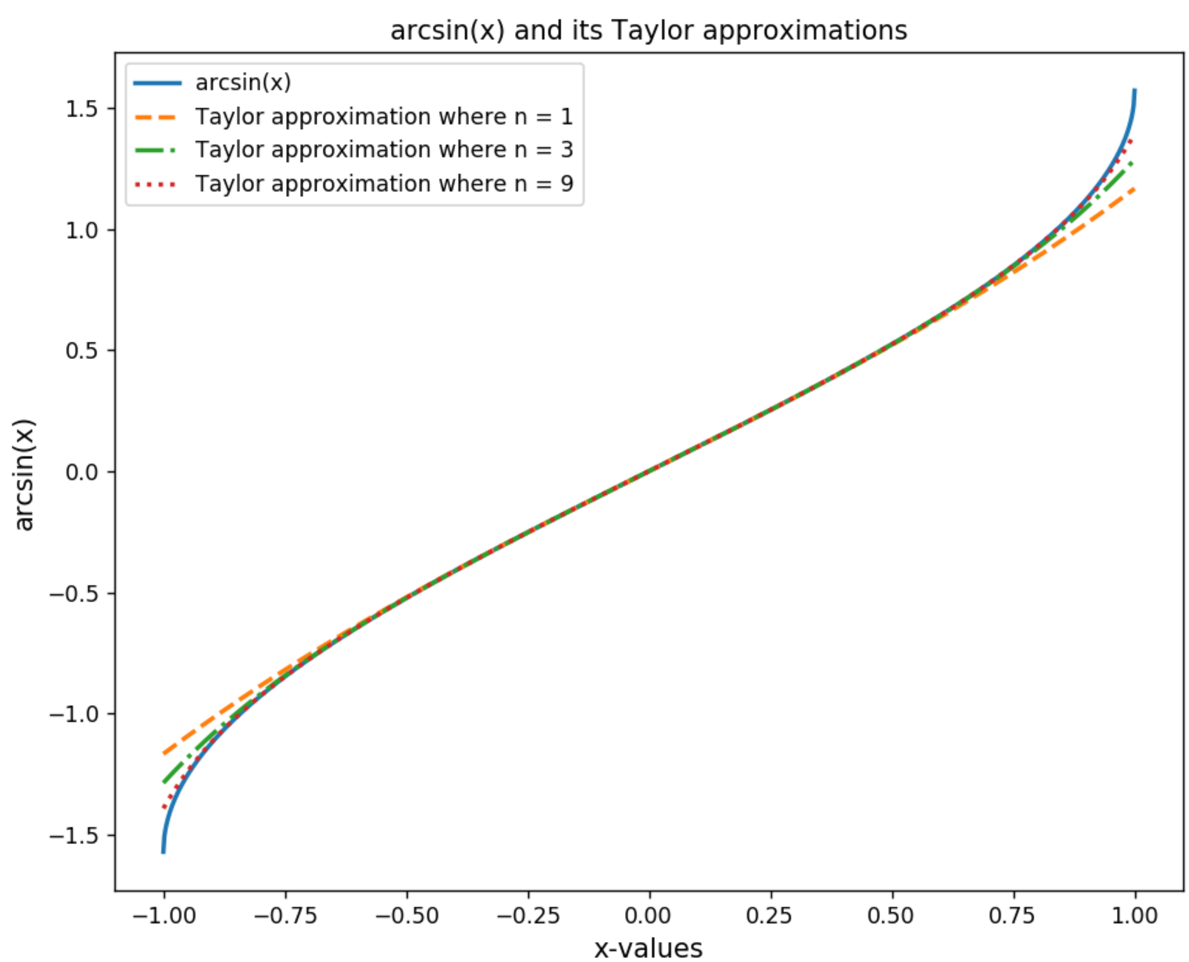
In [14]:
```python
# Problem 2
x = np.linspace(-1,1,1000)
y = np.arcsin(x)
fig, ax = plt.subplots()

# plot arcsin(x) and label graph
ax.plot(x,y,linewidth="2",label="arcsin(x)")
ax.set_xlabel("x-values", fontsize="12")
ax.set_ylabel("arcsin(x)", fontsize="12")
ax.set_title("arcsin(x) and its Taylor approximations", fontsize="12")

# plot Taylor series
t1 = np.zeros(1000)
t3 = np.zeros(1000)
t9 = np.zeros(1000)

for k in range(10):
        if k <= 1:
            t1 += np.math.factorial(2 * k) / (4**k * np.math.factorial(k)**2 * (2 * k + 1)
        if k <= 3:
            t3 += np.math.factorial(2 * k) / (4**k * np.math.factorial(k)**2 * (2 * k + 1)
        t9 += np.math.factorial(2 * k) / (4**k * np.math.factorial(k)**2 * (2 * k + 1)) * 

ax.plot(x,t1,ls="dashed",linewidth="2",label="Taylor approximation where n = 1")
ax.plot(x,t3,ls="dashdot",linewidth="2",label="Taylor approximation where n = 3")
ax.plot(x,t9,ls="dotted",linewidth="2",label="Taylor approximation where n = 9")
ax.legend()
```



```
Out[14]: <matplotlib.legend.Legend at 0x21829d18dc8>
```