ECON425 Homework 1

Madelyn Caufield

Question 1

In [29]:
```
def factorial(x):
  return 1 if x==0  else x * factorial(x-1)

factorial(10)
```

Out[29]:
```
3628800
```

Question 2

In [30]:
```
#Question 2
def odd_product(nums):
  for i in range(len(nums)):
    for j in range(len(nums)):
      if i != j:
        product = nums[i] * nums[j]
        if product & 1:
          return True
        if product & 0:
          return False

dt1 = [2, 4, 5]
dt2 = [1, 3, 4]
print(odd_product(dt1))
print(odd_product(dt2))
```

```
None
True
```

Question 3

In [31]:
```
def reverse(x):
    if x >= 0:
        answer = int(str(x)[::-1])
    else:
        answer = -int(str(-x)[::-1])
    if -3**25 <= answer <= 3**25:
        return answer
    else:
        return 0

print(reverse(234))
print(reverse(-241))
```

```
432
-142
```

Question 4

In [32]:
```
def nocomma(s):
    comma = ",'"
    no_comma = ""
    for char in s:
        if char not in comma:
            no_comma = no_comma + char
    return no_comma

s = "Sit down, please"
print(nocomma(s))
```

```
Sit down please
```

Question 5

In [33]:
```
chars_left = ["(","{","["]
chars_right = [")","}","]"]

def isValid(my_str):
    stack = []
    for i in my_str:
        if i in chars_left:
            stack.append(i)
        elif i in chars_right:
            pos = chars_right.index(i)
            if ((len(stack) > 0) and
                (chars_left[pos] == stack[len(stack)-1])):
                stack.pop()
            else:
                return "false"
    if len(stack) == 0:
        return "true"
    else:
        return "false"

test1 = "()"
test2 = "({})"
test3 = "(}"
test4 = "({[}])"
print(isValid(test1))
print(isValid(test2))
print(isValid(test3))
print(isValid(test4))
```

```
true
true
false
false
```

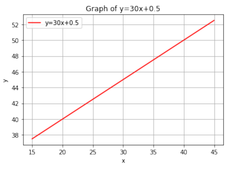
Question 6

In [34]:
```
def merge(a, b):
    merged_list = a + b
    merged_list.sort()
    return(merged_list)

a = [1,3,4]
b = [1,2,6,8]
print(merge(a, b))
```

```
[1, 1, 2, 3, 4, 6, 8]
```

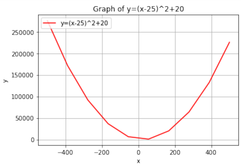
Question 7 Part A

In [35]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(15,45,10)
m = 0.5
b = 30
y = b + m*x
plt.plot(x, y, '-r', label='y=30x+0.5')
plt.title('Graph of y=30x+0.5')
plt.xlabel('x', color='black')
plt.ylabel('y', color='black')
plt.legend(loc='upper left')
plt.grid()
plt.show()
```



Question 7 Part B

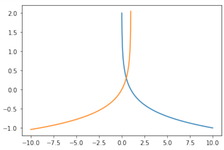
In [36]:
```
x = np.linspace(-500,500,10)
m = 25
b = 20
y = (x-m)**2+b
plt.plot(x, y, '-r', label='y=(x-25)^2+20')
plt.title('Graph of y=(x-25)^2+20')
plt.xlabel('x', color='black')
plt.ylabel('y', color='black')
plt.legend(loc='upper left')
plt.grid()
plt.show()
```



Question 7 Part C

In [37]:
```
x = np.linspace(-10,10,1000)
n = -1
y1 = np.log10(x)/n
y2 = np.log10(1-x)/n
plt.plot(x,y1)
plt.plot(x,y2)
plt.show()
```

```
<ipython-input-37-092d7b9dd795>:3: RuntimeWarning: invalid value encountered in log10
  y1 = np.log10(x)/n
<ipython-input-37-092d7b9dd795>:4: RuntimeWarning: invalid value encountered in log10
  y2 = np.log10(1-x)/n
```



Question 7 Part D

In [38]:
```
x = np.linspace(-10, 10, 100)
z = 1/(1 + np.exp(-x))

plt.plot(x, z)
plt.xlabel("x")
plt.ylabel("Sigmoid(X)")

plt.show()
```