In [8]:

```python
def lesser_of_two_evens(a,b):
    if a%2 == 0 and b%2 == 0:
        if a > b:
            result= b
        elif b > a:
            result= a
        else:
            pass
    else:
        if a  > b:
            result= a
        elif b > a:
            result = b
        else:
            pass
    return result

```

In [10]:

```python
lesser_of_two_evens(2,5)
```

Out[10]:

```
5
```

In [48]:

```python
def animal_crackers(text):
    letter = text.lower().split()
    print(letter)
    
    first = letter[0]
    second = letter[1]
    
    if first[0] == second[0]:
        return True
    else:
        False
```

In [49]:

```python
animal_crackers('Appke ant')
```

```
['appke', 'ant']
```

Out[49]:

```
True
```

In [ ]:

```python

```

In [50]:

```python
def myfunc(n1,n2):
    return n1+n2 == 20 or n1 == 20 or n2 == 20

```

In [51]:

```python
myfunc(10,3)
```

Out[51]:

```
False
```

In [60]:

```
def old_macdonald(name):
    first = name[0]
    between =name [1:3]
    second = name[3]
    rest = name[4:]
    return first.upper()+between+second.upper()+rest
```

In [62]:

```
old_macdonald("nannfsdjkdvsdkdj")
```

Out[62]:

```
'NanNfsdjkdvsdkdj'
```

In [80]:

```
def master_yoda(text):
    word_list = text.split()
    vulta_list = word_list[::-1]
    return ''. join(vulta_list)
```

In [81]:

```
master_yoda('Nithin Sai Reddy')
```

Out[81]:

```
'ReddySaiNithin'
```

In [92]:

```
def almost_there(n):
    return (100-n) <= 10 or (200-n) <= 10
```

In [93]:

```
almost_there(92)
```

Out[93]:

```
True
```

In [96]:

```
def almost_there(n):
    return (abs(100-n) <= 10) or (abs(200-n) <= 10)
```

In [97]:

```
almost_there(91)
```

Out[97]:

```
True
```

In [101]:

```
def has_33(nums):
    for i in range(0,len(nums)-1):

        if i == 3 and i +1 == 3:
            return True
    else:
        return False
```

In [102]:

```
has_33([1, 3, 1, 3])
```

Out[102]:

```
False
```

In [103]:

```
has_33([3, 1, 3])
```

Out[103]:

```
False
```

In [104]:

```
has_33([1, 3, 3])
```

Out[104]:

```
False
```

In [113]:

```
def paper_doll(text):
    word = ''
    for char in text:
        word += char*3
    return word
```

In [114]:

```
paper_doll('Mississippi')
```

Out[114]:

```
'MMMiiissssssiiissssssiiippppppiii'
```

In [127]:

```
def blackjack(a,b,c):
    if (a+b+c) <= 21:
        return a+b+c
    elif 11 in (a,b,c) and sum(a,b,c) <= 31:
        return (a+b+c)-10
    else:
        return 'BUST'
```

In [128]:

```
blackjack(5,6,7)
```

Out[128]:

```
18
```

In [129]:

```
blackjack(9,9,9)
```

Out[129]:

```
'BUST'
```

In [134]:

```
blackjack(1,11,9)
```

Out[134]:

```
21
```

In [136]:

```
def summer_69(arr):
    total = 0
    add = True

    for num in arr:
        while add:
            if num != 6:
                total += num
                break
            else:
                add = False
        while not add:
            if num != 9:
                break
            else:
                add = True
                break
    return total

```

In [137]:

```
summer_69([2, 1, 6, 9, 11])
```

Out[137]:

```
14
```

In [144]:

```
def spy_game(nums):
    out = [0,0,7,'x']
    for num in nums:
        if num == out[0]:
            out.pop(0)
    return len(out)==1

```

In [145]:

```
spy_game([1,2,4,0,0,7,5])
```

Out[145]:

```
True
```

In [1]:

```
def count_primes(num):
    primes = [2]
    if num < 2:
        return 0
    x = 3
    while num >= 2:
        for y in range(3,x,2):
            if y%x == 0:
                x += 2
                break
            else:
                primes.append(x)
                x += 2
    print(primes)
    return len(primes)

```

In [2]:

```
count_primes(100)
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-2-248cdde9552f> in <module>
----> 1 count_primes(100)

<ipython-input-1-d163becfeb2e> in count_primes(num)
      5     x = 3
      6     while num >= 2:
----> 7         for y in range(3,x,2):
      8             if y%x == 0:
      9                 x += 2

KeyboardInterrupt: 
```

In [3]:

```
def count_primes(num):
    primes = [2]
    x = 3
    if num < 2:  # for the case of num = 0 or 1
        return 0
    while x <= num:
        for y in range(3,x,2):  # test all odd factors up to x-1
            if x%y == 0:
                x += 2
                break
        else:
            primes.append(x)
            x += 2
    print(primes)
    return len(primes)
```

In [4]:

```
count_primes(100)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 7
3, 79, 83, 89, 97]
```

Out[4]:

```
25
```

In [ ]:

```

```