In [5]:

```
# ASSIGNMENT PYTHON PRACTICE #
```

In [6]:

```
''' 1. Write a function that inputs a number and prints the multiplication table of



NUM = int(input("Enter the number"))
for i in range(11):
    p = i*NUM
    print(p)
```

```
Enter the number5
0
5
10
15
20
25
30
35
40
45
50
```

In [7]:

```
""" 2. Write a program to print twin primes less than 1000. If two consecutive odd
both prime then they are known as twin primes"""



def prime(num):
    Flag = True
    for i in range(2,num):
        if num%i==0:
            Flag = False
            break
    return Flag


for i in range(1000):
    if(i%2!=0):
        if prime(i)==True and prime(i+2)==True:
            print(i,i+2)
```

```
1 3
3 5
5 7
11 13
17 19
29 31
41 43
59 61
71 73
101 103
107 109
137 139
149 151
179 181
191 193
197 199
227 229
239 241
269 271
281 283
311 313
347 349
419 421
431 433
461 463
521 523
569 571
599 601
617 619
641 643
659 661
809 811
821 823
827 829
857 859
881 883
```

In [14]:

```
# 3. Write a program to find out the prime factors of a number. Example: prime fact



Number = int(input("Enter the number"))
Num = Number
i=2
while(i<=(Num//2)):
    if(Number%i==0):
        Number = Number//i
        print(i)
    else:
        i = i + 1
```

```
Enter the number56
2
2
2
7
```

In [18]:

```
''' 4. Write a program to implement these formulae of permutations and combinations
Number of permutations of n objects taken r at a time: p(n, r) = n! / (n-r)!. Numbe
combinations of n objects taken r at a time is: c(n, r) = n! / (r!*(n-r)!) = p(n,r)




N = int(input("Enter value of n"))
R = int(input(("Enter the value of r")))

def factorial(n):
    if n==1:
        return 1
    else:
        return n*factorial(n-1)
def Pnr(n,r):
    return (factorial(n)//factorial(n-r))
def Cnr(n,r):
    return(factorial(n)//(factorial(r)*factorial(n-r)))


print("The value of permutation is {}".format(Pnr(N,R)))
print("The value of combination is {}".format(Cnr(N,R)))
```

```
Enter value of n10
Enter the value of r4
The value of permutation is 5040
The value of combination is 210
```

In [30]:

```
''' 5. Write a function that converts a decimal number to binary number '''



Num = int(input("Enter the value of decimal number"))
lists = []
while(Num!=0):
    if Num%2==0:
        lists.append(0)
    else:
        lists.append(1)
    Num = Num//2
lists.reverse()
def binary(lis):
    n=0
    for i in lis:
        n=n*10+i
    print(n)
binary(lists)
```

```
Enter the value of decimal number10
1010
```

In [43]:

```
''' 6. Write a function cubesum() that accepts an integer and returns the sum of th
individual digits of that number. Use this function to make functions PrintArmstron
isArmstrong() to print Armstrong numbers and to find whether is an Armstrong number

N = int(input("Enter the Range  "))

def cubesum(num):
    s = 0
    while(num!=0):
        s = s + (num%10)**3
        num = num//10
    return s
def isArmstrong(nu):
    if nu==cubesum(nu):
        return True
    else:
        return False
def PrintArmsrong(n):
    for i in range(1,n+1):
        if(isArmstrong(i)):
            print(i)
PrintArmsrong(N)
```

```
Enter the Number  500
1
153
370
371
407
```

In [48]:

```
''' 7. Write a function prodDigits() that inputs a number and returns the product o
number.'''



def proDigits():
    N = int(input("Enter the number "))
    prod = 1
    while N!=0:
        prod*= N%10
        N=N//10
    print(prod)
proDigits()
```

```
Enter the number 89
72
```

In [57]:

```
''' 8. If all digits of a number n are multiplied by each other repeating with the
digit number obtained at last is called the multiplicative digital root of n. The n
times digits need to be multiplied to reach one digit is called the multiplicative
persistance of n.
Example: 86 -> 48 -> 32 -> 6 (MDR 6, MPersistence 3)
341 -> 12->2 (MDR 2, MPersistence 2)
Using the function prodDigits() of previous exercise write functions MDR() and
MPersistence() that input a number and return its multiplicative digital root and
multiplicative persistence respectively'''

N = int(input("Enter a number"))
def proDigits(N):
    prod = 1
    while N!=0:
        prod*= N%10
        N=N//10
    return prod



def MDR(n):
    i=0
    while n//10!=0:
        n = proDigits(n)
        i+=1
    return (n,i)
ANS = MDR(N)
print("MDR : {} and MPersistence : {}".format(ANS[0],ANS[1]))
```

```
Enter a number341
MDR : 2 and MPersistence : 2
```

In [60]:

```
''' 9. Write a function sumPdivisors() that finds the sum of proper divisors of a n
divisors of a number are those numbers by which the number is divisible, except the
number itself. For example proper divisors of 36 are 1, 2, 3, 4, 6, 9, 18'''



N = int(input("Enter a number"))
divisor = []
def sumPdivisors(N):
    for i in range(1,N//2 + 2):
        if N%i==0:
            divisor.append(i)
sumPdivisors(N)
print(divisor)
```

```
Enter a number36
[1, 2, 3, 4, 6, 9, 12, 18]
```

In [61]:

```
''' 10. A number is called perfect if the sum of proper divisors of that number is
number. For example 28 is perfect number, since 1+2+4+7+14=28. Write a program to
print all the perfect numbers in a given range'''


s = int(input("Enter start "))
e = int(input("Enter end"))
def sumPdivisors(N):
    divisor = []
    for i in range(1,N//2 + 2):
        if N%i==0:
            divisor.append(i)
    return divisor
def perfect(start,end):
    for i in range(start,end+1):
        if sum(sumPdivisors(i))==i:
            print(i)
perfect(s,e)
```

```
Enter start 1
Enter end50
1
6
28
```

In [64]:

```
''' 11. Two different numbers are called amicable numbers if the sum of the proper
each is equal to the other number. For example 220 and 284 are amicable numbers.

Sum of proper divisors of 220 = 1+2+4+5+10+11+20+22+44+55+110 = 284
Sum of proper divisors of 284 = 1+2+4+71+142 = 220
Write a function to print pairs of amicable numbers in a range'''


s = int(input("Enter start "))
e = int(input("Enter end"))
def sumPdivisors(N):
    divisor = []
    for i in range(1,N//2 + 2):
        if N%i==0:
            divisor.append(i)
    return sum(divisor)
def amicable(start,end):
    for i in range(start,end+1):
        n = sumPdivisors(i)
        l = sumPdivisors(n)
        if(i==l):
            print(i,n)
amicable(s,e)
```

```
Enter start 100
Enter end300
220 284
284 220
```

In [65]:

```
''' 12. Write a program which can filter odd numbers in a list by using filter func

lis = list(range(100))

new_lis = list(filter(lambda x:(x%2!=0),lis))
print(new_lis)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 3
7, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71,
73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]
```

In [67]:

```
''' 13. Write a program which can map() to make a list whose elements are cube of e
a given list'''


lis = [1,2,3,4,5,6,7,8,9]
def cube(n):
    return n**3
cube0 = list(map(cube,lis))
print(cube0)
```

```
[1, 8, 27, 64, 125, 216, 343, 512, 729]
```

In [69]:

```
'''14. Write a program which can map() and filter() to make a list whose elements a
even number in a given list'''


lis = [1,2,3,4,5,6,7,8,9]
def cube(n):
    if n%2==0:
        return n**3
cube0 = list(filter(cube,lis))
cube1 = list(map(cube,cube0))
print(cube0)
print(cube1)
```

```
[2, 4, 6, 8]
[8, 64, 216, 512]
```

In [ ]:

```

```