In [1]:
```
'''Write a function that inputs a number and prints the multiplication table of tha
t number'''

def mul(a):
    for i in range(13):
        print("{0}*{1}={2}".format(a,i,a*i))
num=int(input("Please enter a number: "))
mul(num)
```

```
Please enter a number: 5
5*0=0
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50
5*11=55
5*12=60
```

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


def chk(p):
    count=1
    for i in range(1,p):
        if (p>1):
            if p%i==0:
                count+=1
    if count==2:
        isPrime=True
        return(p)
    else:
        isPrime=False
l=list(range(1,1001))
plist=list(filter(chk,l))
print("The List of twin primes within 1000 are : ")
for index,i in enumerate(plist):
    try:
        if(plist[index+1]-i==2):
            print(i,plist[index+1])

    except:
        print('')
```

```
The List of twin primes within 1000 are :
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 [3]:

```
'''3. Write a program to find out the prime factors of a number. Example: prime fac
tors of 56 -
2, 2, 2, 7'''
n=int(input("Please enter a number : "))
result=[]
for i in range(2,n):
    while n % i == 0:
        n = n/i
        result.append(i)

    if n == 1:
        break

if n > 1:
    result.append(n)
print(result)
```

```
Please enter a number : 56
[2, 2, 2, 7]
```

In [4]:
```
"""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
r of
combinations of n objects taken r at a time is: c(n, r) = n! / (r!*(n-r)!) = p(n,r)
/ r!"""

n=int(input("Please enter the Value of N :"))
r=int(input("Please enter the Value of r :"))


def factorial(n):
    if n>0:
        if n == 1:
            return(1)
        else:
            return(n*factorial(n-1))
    else:
        return(1)

def compute(n,r):
    if n>=r and r!=0:
        val_perm=factorial(n)/factorial(n-r)
        print("Number of permutations = ",val_perm)
        val_comb=factorial(n)/(factorial(r)*factorial(n-r))
        print("Number of comnbinations = ",val_comb)
compute(n,r)
```

```
Please enter the Value of N :5
Please enter the Value of r :4
Number of permutations =  120.0
Number of comnbinations =  5.0
```

In [5]:
```
#5. Write a function that converts a decimal number to binary number

n=int(input("Please enter a decimal number :"))
l=[]
a=[]
while n!=1:
    rem=n%2
    if rem==0:
        a=l.append('0')
        n//=2
    else:
        a=l.append('1')
        n//=2
    if n==1:
        a=l.append('1')
binary = ''.join(l)
print(binary)
```

```
Please enter a decimal number :120
0001111
```

In [1]:
```
'''Write a function cubesum() that accepts an integer and returns the sum of the cu
bes of
individual digits of that number. Use this function to make functions PrintArmstron
g() and
isArmstrong() to print Armstrong numbers and to find whether is an Armstrong numbe
r.'''
import math

num=int(input("Please enter a number to check if it is an Amrstrong Number : "))
ran=int(input("Please enter the range to find all the Armstrong numbers with it :
"))
l=math.floor(math.log10(num))+1
def cubesum(num):
    r=num
    sum=0
    while num!=0:
        rem=num%10
        sum=sum+math.pow(rem,l)
        num=num//10
    else:
        print("")
    if sum==r:
        isArmstrong=True
    else:
        isArmstrong=False
    return(isArmstrong)
n_list=list(range(1,ran+1))
cubesum(num)
if cubesum(num) == True :
    print("The number {} is an Armstrong number".format(num))

if cubesum(num) == False :
    print("The number {} is not an Armstrong number".format(num))
armlist=list(filter(cubesum,n_list))
print(armlist)
```

```
Please enter a number to check if it is an Amrstrong Number : 153
Please enter the range to find all the Armstrong numbers with it : 1000


The number 153 is an Armstrong number
```

In [8]:
```
''' 7.Write a function prodDigits() that inputs a number and returns the product of
digits of that
number. '''
from functools import reduce
num=int(input("Enter a number : "))
def prodDigits(num):
    a=[]
    while num!=0:
        rem=num%10
        a.append(rem)
        num//=10
    a=a[::-1]
    product = reduce((lambda x, y: x * y), a)
    return(product)
print(prodDigits(num))
```

```
Enter a number : 123
6
```

In [9]:
```
'''8. If all digits of a number n are multiplied by each other repeating with the p
roduct, the one
digit number obtained at last is called the multiplicative digital root of n. The n
umber of
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)'''

import math
def mult_per(num):
    n= prodDigits(num)
    l=math.floor(math.log10(n))+1
    count=2
    while l>1:
        val=prodDigits(n)
        l=math.floor(math.log10(n))+1
        if l>1:
            count+=1
            if val==0:break
            else:val=prodDigits(val)
            break
    else:
        val=n

    return[val,count]
num=int(input("Please Enter a Number : "))
print("The MDR is :",mult_per(num)[0])
print("The MPersistence is :",mult_per(num)[1])
```

```
Please Enter a Number : 1253
The MDR is : 0
The MPersistence is : 3
```

In [10]:
```
'''9. Write a function sumPdivisors() that finds the sum of proper divisors of a nu
mber. Proper
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'''

num=int(input("Please enter a number : "))
def prop_div(num):
    sum=0
    i=1
    while i<num:
        rem=num%i
        if rem==0:
            sum=sum+i
        i+=1
    return(sum)
print("The sum of proper divisors is: ",prop_div(num))
```

```
Please enter a number : 36
The sum of proper divisors is:  55
```

In [11]:
```
'''10. A number is called perfect if the sum of proper divisors of that number is e
qual to the
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'''

ran=int(input("Please enter a range: "))
ran_list=list(range(1,ran+1))
vsum=0
def chk_Perfect(num):
    reslist=[]
    vsum=0
    i=1
    while i<num:
        rem=num%i
        if rem==0:
            reslist.append(i)
            vsum=vsum+i
        i+=1
    if vsum==num:
        is_Perfect=True
    else:
        is_Perfect=False
    return(is_Perfect)

per_list=list(filter(chk_Perfect,ran_list))
print("Perfect numbers with in the range {} are : {}".format(ran,per_list))
```

```
Please enter a range: 1000
Perfect numbers with in the range 1000 are : [6, 28, 496]
```

In [19]:
```
'''11. Two different numbers are called amicable numbers if the sum of the proper d
ivisors of
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'''


ran=int(input("Please enter a range : "))
def amicable(ran):
    sum=[]
    for i in range(1,ran):
        a=prop_div(i)
        b=prop_div(prop_div(i))
        if i==b:
            if chk_Perfect(i)==False:
                sum.append(i+a)
                if sum.count(i+a)>1:
                    print(i,a)
print("The Amicable numbers within the range {} are : ".format(ran))
amicable(ran)
```

```
Please enter a range : 5000
The Amicable numbers within the range 5000 are :
284 220
1210 1184
2924 2620
```

In [20]:
```
'''12. Write a program which can filter odd numbers in a list by using filter funct
ion'''

num=int(input("Please enter a number : "))
nlist=list(range(1,num+1))
reslist=list(filter(lambda x:x%2==0,nlist))
nlist=set(nlist)
reslist=set(reslist)
print(nlist.difference(reslist))
```

```
Please enter a number : 50
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41,
43, 45, 47, 49}
```

In [1]:
```
'''13. Write a program which can map() to make a list whose elements are cube of el
ements in
a given list'''
import math
num=int(input("Please enter a number : "))
nlist=list(range(1,num+1))
reslist=list(map(lambda x:int(math.pow(x,3)),nlist))

print(reslist)
```

```
Please enter a number : 50
[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, 1331, 1728, 2197, 2744, 3375,
4096, 4913, 5832, 6859, 8000, 9261, 10648, 12167, 13824, 15625, 17576, 19683,
21952, 24389, 27000, 29791, 32768, 35937, 39304, 42875, 46656, 50653, 54872,
59319, 64000, 68921, 74088, 79507, 85184, 91125, 97336, 103823, 110592, 117649,
125000]
```

In [2]:
```
'''14. Write a program which can map() and filter() to make a list whose elements a
re cube of
even number in a given list'''


num=int(input("Please enter a number : "))
nlist=list(range(1,num+1))
reslist=list(filter(lambda x:x%2==0,nlist))
cube_list=list(map(lambda x:int(math.pow(x,3)),reslist))
print(cube_list)
```

```
Please enter a number : 50
[8, 64, 216, 512, 1000, 1728, 2744, 4096, 5832, 8000, 10648, 13824, 17576,
21952, 27000, 32768, 39304, 46656, 54872, 64000, 74088, 85184, 97336, 110592,
125000]
```

In [ ]:

In [ ]: