

In [3]:

```
# (1) Function that input a number and print the multiplication of number

Num = int(input('Enter a number: '))

print('The multiliplication table of number is:')

for i in range (1,11):

    print (Num, 'X', i, '=', Num*i)
```

```
Enter a number: 2900
The multiliplication table of number is:
2900 X 1 = 2900
2900 X 2 = 5800
2900 X 3 = 8700
2900 X 4 = 11600
2900 X 5 = 14500
2900 X 6 = 17400
2900 X 7 = 20300
2900 X 8 = 23200
2900 X 9 = 26100
2900 X 10 = 29000
```

In [5]:

```
# (2) Program to print Twin Prime less than 1000.

def prime(n):

    for i in range(2,n):

        if n%i ==0 :
            return False
    return True

NUM_1 = int(input('Enter Initial Number: '))

NUM_2 = int(input('Enter Final Number: ')) # Here, Final number must be 1000 as per quest
ion

print('Twin prime numbers are ')

for i in range(NUM_1 , NUM_2 + 1):

    if prime(i) and prime(i+2):

        print(i,i+2)
```

```
Enter Initial Number: 2
Enter Final Number: 1000
Twin prime numbers 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 [6]:

```
# (3) Program to find the prime factor of a number

num = int(input('Enter a number: '))

print('The Factors of Required Number are: ')
for i in range (1,num+1):

    if num % i == 0 :

        print(i)
```

Enter a number: 20  
The Factors of Required Number are:  
1  
2  
4  
5  
10  
20

In [4]:

```
# (4) a) Program to implement formulae of Permutation and Combination.

# Permutations with certain number of elements

import itertools

values = [1,2,3,4]

perm = itertools.permutations(values,2)

for val in perm:
    print(*val)
```

1 2  
1 3  
1 4  
2 1  
2 3  
2 4  
3 1  
3 2  
3 4  
4 1  
4 2  
4 3

In [5]:

```
# (4) b) Program to implement formulae of Permutation and Combination.
```

```
# Combinations with certain set of numbers
```

```
import itertools
```

```
values=[3,4,5,6]
```

```
comb = itertools.combinations(values,2)
```

```
for val in comb:  
    print(*val)
```

```
3 4  
3 5  
3 6  
4 5  
4 6  
5 6
```

In [2]:

```
# (5) function that converts a decimal number to binary number
```

```
num = int(input('Enter a number: '))
```

```
num1 = num
```

```
print(bin(num1))
```

Enter a number: 678

0b1010100110

In [8]:

```
# (6) Function to print an armstrong number.
```

```
# An Armstrong number also called Narcissistic number, is a number that is equal to the sum of the cube of its own digits.
```

```
n = int(input("Enter a num : "))
```

```
org = n
```

```
s = 0
```

```
while n>0 :
```

```
    r = n % 10
```

```
    s = s + (r**3)
```

```
    n = n//10
```

```
if org == s :
```

```
    print('It is an armstrong number.')
```

```
else :
```

```
    print ('It is not an Armstrong number.')
```

Enter a num : 407

It is an armstrong number.

In [19]:

```
# (7) function prodDigits() that inputs a number and returns the product of digits of that number.
```

```
def prodDigits(num):
```

```
    prod = 1
```

```
    while (num > 0):
```

```
        rem = num % 10
```

```
        prod = prod * rem
```

```
        num = num // 10
```

```
    return prod
```

```
num = int(input("Enter a number: "))
```

```
print(prodDigits(num))
```

Enter a number: 56

product of num is = 30

In [5]:

```
# (8)Write Multiplicative Digital Root & Multiplicative Persistence of n

def prodDigits(num):
    temp=num
    c=0
    while 1:
        p=1
        c=c+1
        while temp!=0:
            rem=temp%10
            p=p*rem
            temp=int(temp/10)
        if p<10:
            print("Multiplicative Persistence:",c)
            print("Multiplicative Digital Root:",p)
            break
        temp=p
num=int(input("Enter any number: "))
prodDigits(num)
```

Enter any number: 456  
Multiplicative Persistence: 2  
Multiplicative Digital Root: 0

In [4]:

```
#..(9) Function that finds the sum of proper divisors of a number.

def sumPdivisors(a):
    i=1
    sum=0
    for i in range(1,a):
        if a%i==0:
            sum=sum+i
            i+=1
    return sum
a= int(input('Enter a number: '))
print("Sum of proper divisor of a number is",sumPdivisors(a))
```

Enter a number: 65  
Sum of proper divisor of a number is 19

In [16]:

```
# (10) print all the perfect number in a given range

lower = int(input('Enter the lower number: '))
upper = int(input('Enter the upper limit: '))

for num in range(lower,upper + 1):

    result = 0

    for i in range(1,num):

        if (num%i)==0:

            result= result+i

    if num == result:
        print(num)
```

Enter the lower number: 02  
Enter the upper limit: 100  
6  
28

In [28]:

```
# (11) function to pairs of Amicable numbers in a range
```

```
a = int(input('Enter First Number: '))
b = int(input('Enter Second Number: '))

sum = 0
sum1 = 0
for i in range(a,b+1):
    for i in range(1,int(a/2)+1):
        if(a%i ==0):
            sum = sum +i

    for i in range(1, int(b/2)+1):

        if(b%i == 0):
            sum1 = sum1 +i

if (sum==b and sum1==a):

    print(a, ' and', b, 'are amicable Number')

else:

    print(a, ' and ', b, 'are not Amicable Number')
```

```
Enter First Number: 2
Enter Second Number: 100
2 and 100 are not Amicable Number
```

In [18]:

```
# (12) program which can filter odd numbers in a list by using filter function
```

```
def odd_num(x):

    '''
    This function give odd numbers from a list
    '''

    if x % 2 !=0 :

        return x

number_list = range(1,100)

print(list(number_list))

odd_num_list = list(filter(odd_num,number_list))

print(odd_num_list)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70,
 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,
 93, 94, 95, 96, 97, 98, 99]
[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, 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 [19]:

```
# (13) Program which can map() to make a list whose elements are cube of elements in a given list
```

```
numbers = range(1,15)

def power_of_Three(num):

    return num**3
```

```
cube = list(map(power_of_Three, numbers))
```

```
print(cube)
```

```
[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, 1331, 1728, 2197, 2744]
```

In [20]:

```
# (14) Write a program which can map() and filter() to make a list whose elements are cube of even number in a given list
```

```
def power_cube(num):
```

```
    return num**3
```

```
numbers = range(2,20)
```

```
print (list(numbers))
```

```
def even_number(nums):
```

```
    if nums % 2 == 0:
```

```
        return nums
```

```
even_num= list(filter(even_number,numbers))
```

```
print(even_num)
```

```
cube = list(map(power_cube, even_num))
```

```
print(cube)
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
```

```
[8, 64, 216, 512, 1000, 1728, 2744, 4096, 5832]
```