Q 1)Multiplication Table of Given Number in the given Range

In [0]:
```python
def __mul_table(n,x):
    for i in range(1,x+1):
        print("{0} * {1} = {2}".format(n,i,n*i))


n=int(input("Enter Integer number for multiplication Table:" ))
x=int(input("enter the range of multiplication Table:"))
print("The multplication of the given number {0} for given range {1}:".format(n,x))
__mul_table(n,x)
```

```
Enter Integer number for multiplication Table:10
enter the range of multiplication Table:25
The multplication of the given number 10 for given range 25
10 * 1 = 10
10 * 2 = 20
10 * 3 = 30
10 * 4 = 40
10 * 5 = 50
10 * 6 = 60
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100
10 * 11 = 110
10 * 12 = 120
10 * 13 = 130
10 * 14 = 140
10 * 15 = 150
10 * 16 = 160
10 * 17 = 170
10 * 18 = 180
10 * 19 = 190
10 * 20 = 200
10 * 21 = 210
10 * 22 = 220
10 * 23 = 230
10 * 24 = 240
10 * 25 = 250
```

Q 2) Twin Primes below 1000

In [0]:
```python
print("The Twin primes below 1000 are:")
def __prime(num):
    for x in range(2,num):
        if num%x== 0:
            return False
    return True


def __Twins_prime__(f,l):
    for i in range(f,l):
        j=i+2

        if(__prime(i) and __prime(j)):
            print("[{0},{1}]".format(i,j))


__Twins_prime__(2,1000)
```

```
The Twin primes below 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]
```

Q 3) Prime Factors of a given number

In [0]:
```python
num=int(input("Please enter Positive Integer:"))
print("The prime factors of given number are :")

def __prime_fact(n):

    fact = []
    i=2
    while i<= n:
        if n % i:
            i += 1
        else:
            n //= i
            fact.append(i)
    return fact

print(__prime_fact(num))
```

```
Please enter Positive Integer:56
The prime factors of given number are :
[2, 2, 2, 7]
```

Q 4) Permutations and combinations

In [0]:
```python
def fact(n):
    return 1 if n == 1 or n == 0 else (n * factorial(n-1))

def __permut(n,r):
    return(int(fact(n)/fact(n-r)))

def __Comb(n,r):
  #return(int(fact(n)/(fact(r)*fact(n-r))))
  return(int((__permut(n,r))/(fact(r))))

n=int(input("Enter the value n :" ))
r=int(input("Enter the value r :" ))
print("The Permutations for given values:{0}".format(__permut(n,r)))
print("The combinations for given values:",__Comb(n,r))
```

```
Enter the value n :8
Enter the value r :3
The Permutations for given values:336
The combinations for given values: 56
```

Q:5)Decimal Number to Binary

In [0]:
```python
def Dec_Bin(num):
    if num > 1:
        decimalToBinary(num // 2)
    print(num % 2,end='')


num = int(input("Enter any Decimal number: "))
print("Binary form of given number:")
Dec_Bin(num)

print("\nBinary form with inbuilt function:",bin(num))
```

```
Enter any Decimal number: 25
Binary form of given number:
11001
Binary form with inbuilt function: 0b11001
```

Q 6) Cube sum and IsArmstrong Verification

In [0]:
```python
def cubesum(num):
    sum=0
    while num:
        #l=len(str(num))
        x=num%10
        num=num//10
        sum+=x**3
    return sum
def isArmstrong(num):
    if(num == cubesum(num)):
        PrintArmstrong(num)
    else:
        print("Given Number {0} is not an Armstrong".format(num))


def PrintArmstrong(num):
    print("Given number {0} is an Armstrong".format(num))


num=int(input("Enter number:"))
print("The sum of cube of given number is:",cubesum(num))
isArmstrong(num)
```

```
Enter number:407
The sum of cube of given number is: 407
Given number 407 is an Armstrong
```

In [0]:
```python
##########Armstrong for n digit number#######
def nsum(num):
    sum=0
    l=len(str(num))
    while num:
        x=num%10
        num=num//10
        sum+=x**l
    return sum
def isArmstrong(num):
    if(num == nsum(num)):
        PrintArmstrong(num)
    else:
        print("Given Number {0} is not Armstrong".format(num))


def PrintArmstrong(num):
    print("Given number {0} is an Armstrong".format(num))




num=int(input("Enter number:"))
print("The sum of power of given number is:",nsum(num))
isArmstrong(num)
```

```
Enter number:35641594208964132
The sum of power of given number is: 35641594208964132
Given number 35641594208964132 is an Armstrong
```

Q 7)Product of Given number

In [0]:
```python
def prodDigits(num):
    prod=1
    while num:
        x=num%10
        num=num//10
        prod*=x
    return prod


num=int(input("Enter number:"))
print("The product of digits of Given number:",prodDigits(num))
```

```
Enter number:222222
The product of digits of Given number: 64
```

Q 8) multiplicative digital and multiplicative persistence

In [0]:
```python
def MDR(num):
    #print(,num)
    l=1
    lst=[num]
    x=num
    count=0
    while l:
        x=prodDigits(x)
        l=len(str(x))
        count=count+1
        lst.append(x)
        if(l==1):
            print("The multiplicative persistence of n:",count)
            return lst
            break

num=int(input("Enter number:"))
print("The multiplicative digital root of n:",MDR(num))
```

```
Enter number:86
The multiplicative persistence of n: 3
The multiplicative digital root of n: [86, 48, 32, 6]
```

Q 9) Perfect Divisors

In [0]:
```python
def sumPdivisors(num):
    div=[]
    for i in range(1,num//2+1):
        if(num%i==0):
            div.append(i)
    return div


num=int(input("Enter number::"))
print("The perfect divisors are::",sumPdivisors(num))
```

```
Enter number::496
The perfect divisors are:: [1, 2, 4, 8, 16, 31, 62, 124, 248]
```

Q 10) Perfect Number

In [0]:
```python
def perfectdiv(f,l):
    for i in range(f,l):
        lst=sumPdivisors(i)
        if(i == sum(lst)):
             print("{0} is a perferct Number".format(i))


f=int(input("Enter minrange::"))
l=int(input("Enter maxrange::"))
perfectdiv(f,l)
```

```
Enter minrange::4
Enter maxrange::10000
6 is a perferct Number
28 is a perferct Number
496 is a perferct Number
8128 is a perferct Number
```

Q 11) Amicable Numbers

In [0]:
```python
def amicnum(f,l):
  for i in range(f,l):
      lst=sum(sumPdivisors(i))
      x=sum(sumPdivisors(lst))
      #print("{0} and {1} ".format(lst,x))
      if(x==i) and (x!=lst):
          print("{0} and {1} are amicable".format(x,lst))


f=int(input("Enter minrange::"))
l=int(input("Enter maxrange::"))

amicnum(f,l)
```

```
Enter minrange::220
Enter maxrange::6369
220 and 284 are amicable
284 and 220 are amicable
1184 and 1210 are amicable
1210 and 1184 are amicable
2620 and 2924 are amicable
2924 and 2620 are amicable
5020 and 5564 are amicable
5564 and 5020 are amicable
6232 and 6368 are amicable
6368 and 6232 are amicable
```

Q12) Filter Function

In [146]:
```python
print("Enter list with spaces")
lst  = [int(i) for i in input().split()]
odd=filter(lambda x : x % 2 , lst)

print("odd Numbers in the list:",list(odd))
```

```
Enter list with spaces
12 46 78 90 30 12 1 3 4 76 89 34 43 20 121 23
odd Numbers in the list: [1, 3, 89, 43, 121, 23]
```

Q13) Map function

In [147]:
```python
print("Enter list with spaces")
lst  = [int(i) for i in input().split()]
cube=map(lambda x : x **3 , lst)
print("Cube of Given List:",list(cube))
```

```
Enter list with spaces
2 3 4 6 8 9
Cube of Given List: [8, 27, 64, 216, 512, 729]
```

Q14) Filter and Map Functions

In [148]:
```python
print("Enter list with spaces")
lst  = [int(i) for i in input().split()]
even=filter(lambda x : (x%2==0) , lst)
cube=map(lambda x : x **3 , even)
print("Cube of the even numbers in given list:",list(cube))
```

```
Enter list with spaces
2 3 5 6 8 9 10 20 30
Cube of the even numbers in given list: [8, 216, 512, 1000, 8000, 27000]
```

In [0]: