Some Python Programs

In [1]:

```
# 1.program to print multiplication table of a number using function

print("enter the number:")
n = int(input("n="))
i=1

def multable(m):
    """finding multiplication table
    """
    res = m*i
    return(res)
print("the multiplication table of {} is:" .format(n))
for i in range(1 , 11):
    print("{} * {} = {} " .format(n , i , multable(n)))
```

```
enter the number:
n=5
the multiplication table of 5 is:
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
```

In [2]:

```
# 2.program to find TWIN PRIMES less than 1000
lst=[] ; j=0 ; l=0

"""logic to find all prime numbers less than 1000"""
for number in range(2,1001):
    prime=0
    for j in range(2,number):
        if(number%j==0):
            prime=1
    if(prime==0):
        """saving the prime numbers as a list """
        lst += [number]

print("TWIN PRIMES less than 1000 :")
ln= len(lst)
"""logic to find consecutive odd numbers in the list of prime numbers"""
while((l+1)<ln):
    if((lst[l+1]-lst[l])==2):
        print("({} , {})" .format(lst[l],lst[l+1]))
    l += 1
```

```
TWIN PRIMES less than 1000 :
(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 [4]:

```
# 3.program to find prime factors of a number
print("enter the number")
num = int(input("num="))
f=2
print("prime factors of {} :" .format(num))
"""logic to find prime factors: we keep on dividing given number as long as it is
while((num/f)>=1):
    while(num%f==0):
        print(f)
        num /= f
    f += 1
```

```
enter the number
num=63
prime factors of 63 :
3
3
7
```

In [5]:

```
# 4.program to implement permutation and combination

n = int(input("n="))
r = int(input("r="))

def fact(f):
    factorial = 1
    while(f>=1):
        factorial *= f
        f -= 1
    return(factorial)

"""permutation"""
print("p({},{}) = {}" .format(n,r,(fact(n)/fact(n-r))))
"""combination"""
print("c({},{}) = {}" .format(n,r,(fact(n)/(fact(r)*fact(n-r)))))
```

```
n=3
r=2
p(3,2) = 6.0
c(3,2) = 3.0
```

In [6]:

```
# 5.program to covert decimal numbers to binary

print("enter the number")
def conv(n):
    lst = [] ; ln=0
    r = 0
    """finding remainder at each stage and inorder to print them in reverse order
    while(n>=1):
        r = n%2
        lst += [int(r)]
        n /= 2
    ln = len(lst)
    for i in range(ln-1 , -1 , -1):
        print(lst[i] , end='')

n = int(input("n="))
print("the binary equivalent of {} = " .format(n) , end='')
conv(n)
```

```
enter the number
n=4
the binary equivalent of 4 = 100
```

In [11]:

```
# 6.program to implement cubesum() , printarmstrong() , isarmstrong()
print("enter the number")
n = int(input("n="))

def cubesum(i):
    """fuunction to find sum of cube of digits"""
    r = 0 ; cube = 0 ; sum = 0
    while(i>0):
        r = i%10
        cube = r**3
        sum = sum + cube
        i = int(i/10)
    return(sum)
k=cubesum(n)

def printarmstrong(i):
    """function prints the given number if it is armstrong"""
    if(i==k):
        return(i)
def isarmstrong(i):
    """funtion to check whether given number is armstrong , if so we print it usi
    if(i==k):
        print("{} is armstrong" .format(printarmstrong(n)))
    else:
        print("given number is not armstrong" .format(i))
isarmstrong(n)
```

```
enter the number
n=153
153 is armstrong
```

In [2]:

```
# 7. Funtion proDigits() to find product of digits of that number

print("enter the number")
n = int(input("n="))

def proDigits(i):
    pro = 1
    while(i>0):
        k = i%10
        pro *= k
        i = int(i/10)
    return pro
print("the product of digits of {0} is {1}" .format(n , proDigits(n)))
```

```
enter the number
n=456
the product of digits of 456 is 120
```

In [4]:

```
#8. function sumPdivisors() to print sum of proper divisors of given number

print("enter the number")
n = int(input("n="))

def sumPdivisors(num):
    lst = [] ; i=1 ; sum = 0
    for i in range(1 , int(num/2)+1):
        if((num%i)==0):
            lst += [i]
    for ele in lst :
        sum += ele
    return sum
print("the sum of proper divisors of {} are {}" . format(n , sumPdivisors(n)))
```

```
enter the number
n=36
the sum of proper divisors of 36 are 55
```

In [8]:

```
#9 . program to print perfect numbers in a range

print(" enter the required range r1 and r2")
r1 = int(input("r1="))
r2 = int(input("r2="))
k=0 ; lst1 = []

for k in range(r1 , r2 + 1):
    if(k == sumPdivisors(k)):
        lst1 += [k]
print(lst1)
```

```
 enter the required range r1 and r2
r1=1
r2=10000
[6, 28, 496, 8128]
```

In [6]:

```
#10 . program to print all amicable numbers in a given range
"""function to find sum of divisors"""
def sumPdivisors(num):
    lst = [] ; i=1 ; sum = 0
    for i in range(1 , int(num/2)+1):
        if((num%i)==0):
            lst += [i]
    for ele in lst :
        sum += ele
    return sum

print("enter the ranges r1 and r2")
r1 = int(input("r1="))
r2 = int(input("r2="))
lsta = []
for p in range(r1 , r2 + 1):
    for q in range(r1 , r2 + 1):
        if((p == sumPdivisors(q))and(q == sumPdivisors(p))and(p!=q)):
            lsta += [(p , q)]
print("amicable pairs in given range {}" .format(lsta))
```

```
enter the ranges r1 and r2
r1=1184
r2=1220
amicable pairs in given range [(1184, 1210), (1210, 1184)]
```