1) MULTIPLICATION TABLE OF ANUMBER

In [1]:

```
def mul_table(num):
    """print multiplication table of given number"""
    n=int(input("enter number of terms required"))
    print("MULTIPLICATION TABLE OF",num)
    for i in range(1,n+1):
        print("{} * {} = {}".format(i,num,i*num))
mul_table(3)
```

```
enter number of terms required10
MULTIPLICATION TABLE OF 3
1 * 3 = 3
2 * 3 = 6
3 * 3 = 9
4 * 3 = 12
5 * 3 = 15
6 * 3 = 18
7 * 3 = 21
8 * 3 = 24
9 * 3 = 27
10 * 3 = 30
```

2)TWIN PRIMES BELOW 1000

In [21]:

```
b=[]
for num in range(1,100 + 1):
    if num > 2:
        for i in range(2,num):
            if (num % i) == 0:
                break
        else:
            b.append(num)
for i in range(0,len(b)-1):
    if(b[i+1]-b[i]==2):
        print(b[i],'and',b[i+1])
```

```
3 and 5
5 and 7
11 and 13
17 and 19
29 and 31
41 and 43
59 and 61
71 and 73
```

3) prime factors of a number

In [32]:

```
def prime_factor(n):
    """returns a list of prime factors of n"""
    i=2
    factors=[]
    while n>1:
        if n%i==0:
            factors.append(i)
            n=n//i
        else:
            i=+i
    return factors
result=prime_factors(56)
print("the prime factors are",result)
```

```
the prime factors are [2, 2, 2, 7]
```

4) permutation and combination

In [2]:

```
def factorial(n):
    fact=1
    for i in  range(1,n+1):
        fact=fact*i
    return fact

def permutation(n,r):
    permutation=factorial(n)/factorial(n-r)
    return permutation


def combination(n,r):
    combination=permutation(n,r)/factorial(r)
    return combination
```

In [3]:
```
permutation(5,3)
```

Out[3]:
```
60.0
```

In [4]:
```
combination(5,3)
```

Out[4]:
```
10.0
```

5) decimal to binary conversion

In [5]:
```
def dectobin(n):
    """convert decimal number to binary using recursion"""
    if n>1:
        dectobin(n//2)
    print(n%2,end=" ")
dectobin(56)
```

```
1 1 1 0 0 0
```

6) cubesum() , PrintArmstrong() and isArmstrong()

In [6]:
```
def cubesum(n):
    sum=0
    while n>0:
        r=n%10
        sum=sum+r*r*r
        n=n//10
    return sum
    print(sum)
cubesum(153)
```

Out[6]:

```
153
```

In [7]:

```
def  printamstrong(start,end):
    for i in  range(start,end+1):
        s=cubesum(i)
        if s==i:
            print(s)
printamstrong(1,500)
```

```
1
153
370
371
407
```

In [4]:

```
def isamstrong(number):
    s=cubesum(number)
    if s==number:
        print("it is amstrong")
    else:
        print("it is not amstrong")
isamstrong(1634)
```

```
it is not amstrong
```

7) product of digits of given number

In [10]:

```
def proddigits(n):
    """return product of digits of given number"""
    mul=1
    while n>1:
        r=n%10
        mul=mul*r
        n=n//10
    return(mul)
proddigits(153)
```

Out[10]:

```
15
```

8) to calculate multiplicative digital root and multiplicative persistence

In [116]:

```
count=1
def mdr(n):
    global count
    s=proddigits(n)
    while s>9:
        count=count+1
        s=proddigits(s)
    print("multiplicative digital root  = ",s)

def mr():
    print("multiplicative persistence = ",count)
n=int(input("enter the number"))
mdr(n)
mr()
```

```
enter the number86
multiplicative digital root  =  6
multiplicative persistence =  3
```

9)sum of proper divisors of given number

In [12]:

```
def sumpdivisors(n):
    """sum of proper divisors of given number"""
    sum=0
    for i in range(1,n):
        if n%i==0:
            sum+=i
    return sum
n=int(input("enter the number"))
result=sumpdivisors(n)
print("the sum of proper divisors of {0} = {1}".format(n,result))
```

```
enter the number36
the sum of proper divisors of 36 = 55
```

10) perfect numbers in a range

In [19]:

```
def perfectnumber(start,end):
    """return the perfect numbers on given range"""
    for i in range(start,end+1):
        sum=0
        for j in range(1,i):
            if i%j==0:
                sum+=j
        if i==sum:
            print(i)
perfectnumber(1,8500)
```

```
6
28
496
8128
```

11) Amicable numbers between a range

In [16]:

```
def sumofdiv(x):
    """Amicable numbers in a range"""
    sum=1
    for i in range(2,x):
        if x%i==0:
            sum+=i
    return sum
start=int(input("enter the start limit"))
end=int(input('enter the end limit'))
for num1 in range(start,end):
    for num2 in range(start,end):
         if (num1==sumofdiv(num2)) and (num2==sumofdiv(num1)) and (num1
!=num2):
                                      print(num1,  "and " ,num2)
```

```
enter the start limit1
enter the end limit1000
220 and  284
284 and  220
```

these (11) program take more time to run and produce output . can you please suggest how to reduce complexity of the program

12)

```
In [12]: numbers=[2,1,3,4,5,6,7,8,9,20,21,23,34]
         list(filter(lambda x:x%2==1,numbers))
```

```
Out[12]: [1, 3, 5, 7, 9, 21, 23]
```

13)

```
In [13]: list(map(lambda x:x**3,numbers))
```

```
Out[13]: [8, 1, 27, 64, 125, 216, 343, 512, 729, 8000, 9261, 12167, 39304]
```

14)

In [14]:
```
p=list(filter(lambda x:x%2==0,numbers))
list(map(lambda x:x**3,p))
```

Out[14]:
```
[8, 64, 216, 512, 8000, 39304]
```