1.

In [9]:

```
def multiply(num):
    i=1
    for i in range(1,11):
        print(num*i)
```

In [10]:

```
multiply(10)
```

```
10
20
30
40
50
60
70
80
90
100
```

2

In [59]:

```
def twinPrimes():
    temp = 0
    for i in range(3,1000,2):
        isDiv = False
        for j in range(2,i):
            if(i%j==0):
                isDiv = True
        if(isDiv == False):
           # print(i)
            if((i-temp) == 2):
                print(temp,i)
            temp = i
```

In [60]:

```
twinPrimes()
```

```
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
```

3

In [73]:

```
def factors(num):
    for i in range(2,num):
        while(num%i==0):
            num = num/i
            print(i)
```

In [76]:

```
factors(100)
```

```
2
2
5
5
```

4

In [137]:

```
def fact(num):
    if (num == 1):
        return(1)
    else:
        num = num * fact(num-1)

    return(num)


def perm(n,r):
    res = fact(n)/fact(n-r)
    return(int(res))

def comb(n,r):
    res = perm(n,r)/fact(r)
    return(int(res))
```

In [139]:

```
print(perm(4,2))
print(comb(4,2))
```

```
12
6
```

5

In [191]:

```
def dec2bin(num):
    rem = []
    while(num!=0):
        rem.append(num % 2)
        num = int(num/2)
    rem.reverse()

    for ele in rem:
        print(ele)
```

In [192]:

```
dec2bin(8)
```

```
1
0
0
0
```

6

In [2]:

```
def cubesum(num):
    sum= 0
    while(num != 0):
        rem = num % 10
        num = int(num/10)
        sum = sum + rem ** 3
    return(sum)


def printArmstrong(num):
    for i in range(1,num):
        if(i == cubesum(i)):
            print(i)


def isArmstrong(num):
    if(num == cubesum(num)):
        print(num,"is an Armstrong number")
    else:
        print(num,"is not an Armstrong number")
```

In [12]:

```
print('cubesum:',cubesum(123))
print('********')
print('Armstrong number between 1 and 1000')
printArmstrong(1000)
print('********')
isArmstrong(407)
```

```
cubesum: 36
********
Armstrong number between 1 and 1000
1
153
370
371
407
********
407 is an Armstrong number
```

7

In [13]:

```
def prodDigits(num):
    prod = 1
    while(num != 0):
        rem = num % 10
        num = int(num/10)
        prod = prod * rem
    return(prod)
```

In [14]:

```
prodDigits(125)
```

Out[14]:

```
10
```

8

In [226]:

```
def MDR(num):
    while((num % 10) != num):
        num = prodDigits(num)
    return(num)
```

In [228]:

```
MDR(341)
```

Out[228]:

```
2
```

In [231]:

```
def MPersistence(num):
    i=0
    while((num % 10) != num):
        num = prodDigits(num)
        i = i+1
    return(i)
```

In [233]:

```
MPersistence(86)
```

Out[233]:

```
3
```

9

In [240]:

```
def sumPdivisors(num):
    lst = []
    for i in range(1,num):
        if(num % i == 0):
            lst.append(i)
    return(lst)
```

In [252]:

```
sumPdivisors(28)
```

Out[252]:

```
[1, 2, 4, 7, 14]
```

10

In [253]:

```
def perfect(num):
    for i in range(1,num):
        if(i == sum(sumPdivisors(i))):
            print(i)
```

In [255]:

```
perfect(50)
```

```
6
28
```

11

In [282]:

```
def amicable(num):
    for i in range(1,num):
        temp = sum(sumPdivisors(i))
        temp2 = sum(sumPdivisors(temp))
        if(i == temp2 and i != temp and i < temp):
            print(i,temp)
```

In [283]:

```
amicable(300)
```

```
220 284
```

12

In [324]:

```
def odd(lst):
    abc = filter(lambda x:x%2,lst)
    return(list(abc))
```

In [325]:

```
print(odd([1,2,3,4]))
```

```
[1, 3]
```

13

In [328]:

```
def cube(lst):
    abc = map(lambda x:x**3,lst)
    return(list(abc))
```

In [330]:

```
cube([1,2,3,4])
```

Out[330]:

```
[1, 8, 27, 64]
```

14

In [335]:

```
def evenCube(lst):
    abc = map(lambda x:x**3,filter(lambda x:x%2==0,lst))
    return(list(abc))
```

In [334]:

```
evenCube([1,2,3,4])
```

Out[334]:

```
[8, 64]
```