In [3]:
```
#Q1 A
import math

class factor:

    def findfactor(self):
        factors = [int(x) for x in input("Enter the Factors\n").split()]
#         print(factors)
        target = int(input("Enter the target element: \n"))
#         print(target)
        for i in range(len(factors)):
            rem = target-factors[i]
            try:
                if factors.index(rem):
                    j = factors.index(rem)
                    print("[{},{}]".format(i,j))
            except:
                pass


ur5 = factor()
ur5.findfactor()
```

```
Enter the Factors
15 25 35 45 55
Enter the target element:
40
[0,1]
```

In [4]:
```
#Q1 B
class sets:
    def factorial(self,a):
        if a==0:
            return 1
        else:
            return self.factorial(a-1)*a

    def powerset(self):
        inp = [int(x) for x in input("Enter the Factors\n").split()]
#         print(inp)
        finalset = []
        pow = len(inp)
        count = 2**pow
#         print(count)
        for x in range(count):
            binnum = str(bin(x)[2:].zfill(pow))
            thisset = []
            for i in range(len(binnum)):
                if binnum[i]=="1":
                    thisset.append(inp[i])
            finalset.append(thisset)
        print(finalset)



ur3 = sets()

ur3.powerset()
```

```
Enter the Factors
4 5 6 7
[[], [7], [6], [6, 7], [5], [5, 7], [5, 6], [5, 6, 7], [4], [4, 7], [4, 6], [4,
6, 7], [4, 5], [4, 5, 7], [4, 5, 6], [4, 5, 6, 7]]
```

In [6]:
```
#Q2
import math

class calc:
    def calculator(self,x,y,opr):
        try:
            if opr=="+":
                return x+y
            elif opr=="-":
                return x-y
            elif opr=="*":
                return x*y
            elif opr=="/":
                return x/y
            elif opr=="%":
                return x//y
            elif opr=="^":
                return x**y
        except:
            pass

cal = calc()
oper = input("Enter the Operation. Seperate by space").split()
# print(oper)
oprans = int(oper[0])
oper.pop(0)
while len(oper)>0:
    operand = oper[0]
    opr2 = int(oper[1])
    oper.pop(1)
    oper.pop(0)
    try:
        oprans = cal.calculator(oprans,opr2,operand)
    except:
        pass

print(oprans)
```

```
Enter the Operation. Seperate by space1 + 2 + 3 + 4 - 5
5
```

In [13]:
```
#Q3
from threading import *
from time import sleep

class thread1(Thread):
    def __init__(self,num):
        super().__init__()
        self.num = num
        self.result = False
    def run(self):
        lim = self.num//2
        for i in range(2,lim):
            if self.num%i==0:
                self.result = False
        self.result = True

class thread2(Thread):
    def __init__(self,num):
        super().__init__()
        self.num = num
    def run(self):
        lim = self.num//2
        for i in range(lim,self.num):
            if self.num%i==0:
                self.result = False
        self.result = True

number = int(input("Enter the number to be checked"))
t1 = thread1(number)
t2 = thread2(number)

t1.start()
t2.start()

t1.join()
t2.join()
if t1.result and t2.result:
    print(str(number)+" is a prime number")
else:
    print(str(number)+" is not a prime number")
```

```
Enter the number to be checked11
True True
11 is a prime number
```

In [ ]: