In [1]:

```
'Week 2 Practice Programming Assignment'

def intreverse(val):
    'Finding Reverse of Given Integer'
    res = 0
    while val>0:
        res = res*10+val%10
        val //=10
    return res

def matched(st):
    'Finding if given string has Matching & Balanced parentheses'
    c = 0
    for i in st:
        if i=='(':
            c+=1
        if i==')':
            c-=1
        if c<0:
            return False
    return True if c==0 else False

def sumprimes(L):
    'Sum of primes number in given list, (2,infinity) is range of primes'
    mx = max(L)+1
    prime_index = [1]*mx
    prime_index[0:2] = [0,0]
    for i in range(2,round(mx**0.5)+1):
        if prime_index[i]:
            for j in range(i*2,mx,i):
                prime_index[j] = 0
    return sum(i*prime_index[i] for i in L if i>=2)
```

In [2]:

```
_ = '''DO NOT COPY THIS CODE'''

a = '''Test Case 1 intreverse(31511) 11513
Test Case 2 intreverse(4) 4
Test Case 3 intreverse(15135324234235) 53243242353151
Test Case 4 matched("a3qw3;4w3(aasdgsd)((agadsgdsgag)agaga)") True
Test Case 5 matched("(ag(Gaga(agag)Gaga)GG)a)33)cc(") False
Test Case 6 matched("(adsgdsg(agaga)a") False
Test Case 7 matched("15ababa.agaga[][[[") True
Test Case 8 sumprimes([101,93,97,44]) 198
Test Case 9 sumprimes([1001,393,743,59]) 802
Test Case 10 sumprimes([11,11,11,13,11,-11]) 57'''


for i1,i2,i3,j,k in [x.split(' ') for x in a.split('\n')]:
    print(eval(j),eval(k))

_ = '''NON-COPYING REGION END'''
```

```
11513 11513
4 4
53243242353151 53243242353151
True True
False False
False False
True True
198 198
802 802
57 57
```

In [ ]:

```

```

In [3]:

```
'Week 2 Programming Assignment'

def primepartition(val):
    'Finding if exist two primes p,q such that p+q==val'
    mx = val+1
    prime_index = [1]*mx
    prime_index[0:2] = [0,0]
    for i in range(2,round(mx**0.5)+1):
        if prime_index[i]:
            for j in range(i*2,mx,i):
                prime_index[j] = 0
    primes = set(i for i in range(mx) if prime_index[i])
    for i in primes:
        if val-i in primes:
            return True
    else:
        return False

def nestingdepth(st):
    'Maximum nested depth of parentheses, -1 if invalid'
    mx,c = 0,0
    for i in st:
        if i=='(':
            c+=1
        if i==')':
            c-=1
        if c<0:
            return -1
        if c>mx:
            mx = c
    return mx if c==0 else -1

def rotatelist(L,k):
    'Left rotated list, for positive k only'
    return L[k%len(L):]+L[:k%len(L)] if k>0 else L
```

In [4]:

```
_ = '''DO NOT COPY THIS CODE'''

a = '''Test Case 1 primepartition(7) True
Test Case 2 primepartition(185) False
Test Case 3 primepartition(3432) True
Test Case 4 nestingdepth("zb%78") 0
Test Case 5 nestingdepth("(7)(a") -1
Test Case 6 nestingdepth("a)*(?") -1
Test Case 7 nestingdepth("((jkl)78(A)&l(8(dd(FJI:),):)?)") 4
Test Case 8 rotatelist([1,2,3,4,5],1) [2,3,4,5,1]
Test Case 9 rotatelist([1,2,3,4,5],3) [4,5,1,2,3]
Test Case 10 rotatelist([1,2,3,4,5],12) [3,4,5,1,2]'''

for i1,i2,i3,j,k in [x.split(' ') for x in a.split('\n')]:
    print(eval(j),eval(k))

_ = '''NON-COPYING REGION END'''
```

```
True True
False False
True True
0 0
-1 -1
-1 -1
4 4
[2, 3, 4, 5, 1] [2, 3, 4, 5, 1]
[4, 5, 1, 2, 3] [4, 5, 1, 2, 3]
[3, 4, 5, 1, 2] [3, 4, 5, 1, 2]
```

In [ ]:

```

```