In [1]:

```
def ct(L):
    M = [ v*10 for v in L ]
    N1 = [ x**2 for x in L if x > L[0] ]
    L = sorted(L)
    N2 = [ x**2 for x in L if x > L[0] ]
    print(L + M)
    return N1 + N2 + [max(L)] * (1 + min(L))
L = [4, 1, 6]
print(ct(L))
print(L)
```

```
[1, 4, 6, 40, 10, 60]
[36, 16, 36, 6, 6]
[4, 1, 6]
```

In [3]:

```
def ct(L):
    M = [ v*10 for v in L ]
    N1 = [ x**2 for x in L if x > L[0] ]
    L.sort()
    N2 = [ x**2 for x in L if x > L[0] ]
    print(L + M)
    return N1 + N2 + [max(L)] * (1 + min(L))
L = [4, 1, 6]
print(ct(L))
print(L)
```

```
[1, 4, 6, 40, 10, 60]
[36, 16, 36, 6, 6]
[1, 4, 6]
```

In [2]:

```
def ct(L):
    M = list(range(len(L)))
    for i in range(0, len(L), 2):
        j = L[i]
        v = L[i+1]
        M[-j] += v
    return M
print(ct([1, 1, 3, 3]))
```

```
[0, 4, 2, 4]
```

In [4]:

```
def ct(L):
    M = [ ]
    while sum(L) > sum(M):
        M.extend(L[:2])
    v = M.pop(0)
    v += M.pop()
    M.insert(1, v)
    print(M)
print(ct(list(range(4, 7))))
```

```
[5, 9, 4]
None
```

In [5]:

```
import copy
def ct(L):
    M = L
    N = copy.copy(L)
    x = L.append(-4)
    print(x)
    M[1] = -5
    N[1] = -3
    print(M)
    return N
```

```
L = [2, 6]
print(ct(L))
print(L)
```

```
None
[2, -5, -4]
[2, -3]
[2, -5, -4]
```

In [6]:

```
def ct(L):
    M = [ abs(v) for v in L ]
    N1 = [ L[i]+10 for i in range(len(L)) ]
    L.append(L[2])
    N2 = [ L[i]+10 for i in range(1, len(L)) if L[i] > L[i-1] ]
    print(M)
    return N1 + N2
L = [-5, -1, 8]
print(ct(L))
print(L)
```

```
[5, 1, 8]
[5, 9, 18, 9, 18]
[-5, -1, 8, 8]
```

In [7]:

```
def ct(L):
    M = list(range(len(L)))
    for i in range(1, len(M)):
        L[i] += M[i]
        M[-i] += M[i]
    return M
L = [-5, -4, -4, 2]
print(ct(L))
print(L)
```

```
[0, 5, 4, 4]
[-5, -3, -2, 6]
```

In [8]:

```
def ct(L):
    M = [ 1, 2 ]
    while len(M) < max(L):
        N = M[:2]
        M.extend(N)
        M[0] *= 2
    L.extend([1, 2])
    v = L.pop(0)
    v += L.pop()
    M.insert(0, v)
    print(M)
L = list(range(1, 5))
print(ct(L))
print(L)
```

```
[3, 2, 2, 1, 2]
None
[2, 3, 4, 1]
```

In [9]:

```
import copy
def ct(L):
    M = L
    N = copy.copy(L)
    M.append(-3)
    M = M + [-2]
    v = L.pop(0)
    print(N.extend([v, -5]))
    print(M)
    return N
```

```
L = [2, 6]
print(ct(L))
print(L)
```

```
None
[2, 6, -3, -2]
[2, 6, 2, -5]
[6, -3]
```

In [10]:

```
import copy
def ct(L):
    M = L
    N = [M.pop(1) * 5] * 2
    L += N
    M = M + N
    print(M)
    return N
L = [4, 1]
print(ct(L))
print(L)
```

```
[4, 5, 5, 5, 5]
[5, 5]
[4, 5, 5]
```

In [11]:

```
def ct(t):
    L = [ ]
    for x,y in t:
        L.append(x+y)
    return tuple(L) + t[2]
t = ((-5, 3), (1, -4), (-1, -1))
print(ct(t))
```

```
(-2, -3, -2, -1, -1)
```

In [12]:

```
def ct(L):
    t1 = (-3, -4) * len(L)
    t2 = tuple()
    for i in range(len(L)):
        if L[i] == t1[i]:
            t2 += (i,)
        else:
            t2 += (sum(L[i]),)
    return t2

print(ct([(-3, -4), (1, 3), (3, -4), (-3, -4)]))
```

```
(-7, 4, -1, -7)
```

In [13]:

```
def ct(L):
    M = [ (L[i-1], L[i]) for i in range(1, len(L)) ]
    t = tuple(M)
    t += ((min(L), max(L)),)
    return t
print(ct([1, 4, 3, -2]))
```

```
((1, 4), (4, 3), (3, -2), (-2, 4))
```

In [14]:

```
def ct1(L, M):
    N = L
    L += [1]
    M += [2]
    N = N + [3]
    return L, M, N
L = [4]
```

```
L, M, N = ct1(L, L+[5])
```

```
[4, 1] [4, 5, 2] [4, 1, 3]
```

In [15]:

```
def ct2(L):
    M = sorted(L)
    N = list(reversed(M))
    n = L.pop(0) * 10
    L.remove(1)
    L.sort()
    L.append(n)
    L.extend(L[:1])
    return (L.index(3), M.index(3), N.index(3))
L = [1,2,3,2,1]
M = ct2(L)
print(L, M, type(M) == list)
```

```
[2, 2, 3, 10, 2] (2, 4, 0) False
```

In [16]:

```
def ct3(s):
    L = [int(v) for v in s.split(',')]
    M = [
     L > L[:-1],
     sum(L) - max(L),
     L[-1:]*L[-1]
     ]
    N = [str(v) for v in M]
    return '-'.join(N)
print(ct3('1,23,4'))
```

```
True-5-[4, 4, 4, 4]
```

In [17]:

```
def ct1(L, M):
    N = L
    L += [2]
    M += [1]
    N = N + [0]
    return L, M, N
L = [3]
L, M, N = ct1(L, L+[4])
print(L, M, N)
```

```
[3, 2] [3, 4, 1] [3, 2, 0]
```

In [18]:

```
def ct2(L):
    M = sorted(L)
    N = list(reversed(M))
    n = L.pop(0) * 10
    L.remove(3)
    L.sort()
    L.append(n)
    L.extend(L[:1])
    return (L.index(1), M.index(1), N.index(1))
L = [3,2,1,2,3]
M = ct2(L)
print(L, M, type(M) == tuple)
```

```
[1, 2, 2, 30, 1] (0, 0, 4) True
```

In [19]:

```
def ct3(s):
    L = [int(v) for v in s.split(',')]
    M = [
     L > L[:-1],
     L[-1:]*L[-1],
     sum(L) - min(L)
     ]
```

```
    N = [str(v) for v in M]
    return '-'.join(N)
print(ct3('12,4,3'))
```

```
True-[3, 3, 3]-16
```

In [ ]: