Cyclical figurate numbers

Problem 61 (http://projecteuler.net/problem=061)

Triangle, square, pentagonal, hexagonal, heptagonal, and octagonal numbers are all figurate (polygonal) numbers and are generated by the following formulae:

Figure	Formula	Numbers
Triangle	$P_{3,n}$=n(n+1)/2	1, 3, 6, 10, 15, …
Square	$P_{4,n}$=n²	1, 4, 9, 16, 25, …
Pentagonal	$P_{5,n}$=n(3n−1)/2	1, 5, 12, 22, 35, …
Hexagonal	$P_{6,n}$=n(2n−1)	1, 6, 15, 28, 45, …
Heptagonal	$P_{7,n}$=n(5n−3)/2	1, 7, 18, 34, 55, …
Octagonal	$P_{8,n}$=n(3n−2)	1, 8, 21, 40, 65, …

The ordered set of three 4-digit numbers: 8128, 2882, 8281, has three interesting properties.

- The set is cyclic, in that the last two digits of each number is the first two digits of the next number (including the last number with the first).
- Each polygonal type: triangle (P3,127=8128), square (P4,91=8281), and pentagonal (P5,44=2882), is represented by a different number in the set.
- This is the only set of 4-digit numbers with this property.

Find the sum of the only ordered set of six cyclic 4-digit numbers for which each polygonal type: triangle, square, pentagonal, hexagonal, heptagonal, and octagonal, is represented by a different number in the set.

Solution

In [1]:

```
from itertools import permutations
from itertools import product
from copy import deepcopy

def fignum(x,n = None):
    i = 0
    while (n is None) or (i < n):
        i += 1
        yield i * ((x - 2) * i + 4 - x) // 2

lst=[]
for x in range(3,9):
    lst.append([])
    fn = fignum(x)
    n = ''
    while len(n) < 5:
        if len(n) == 4:
            lst[-1].append(n)
        n = str(next(fn))
c = 1
for z in permutations(lst):
    p = list(deepcopy(z))
    pl = []
    newpl = list(map(len,p))
    while any(len(x)>2 for x in p) and (newpl != pl):
        pl = newpl
        newlst=[]
        for i in range(len(p)):
            a, b = p[i], p[(i + 1) % len(p)]
            newlst.append([x for x in a if any(x[2:]==y[:2] for y in b)])
        p = newlst
        newpl = list(map(len,p))

    if all(len(x)>0 for x in p):
        break

for z in product(*p):
    for i in range(len(p)):
        a, b = z[i], z[(i + 1) % len(z)]
        if a[2:] != b[:2]:
            break
    else:
        print(*z)
        print(sum(map(int, z)))
```

```
8256 5625 2512 1281 8128 2882
28684
```