

```

1 Sub = ['s', 'p', 'd', 'f', 'g', 'h', 'i', 'j', 'k'
, 'l'] # Sub electron shell
2 Main = ['K', 'L', 'M', 'N', 'O', 'P', 'Q'] # Main
electron shell
3 Num = []
4 Sequence = []
5 k = []
6 h = []
7 Period = ['H', 'He', 'Li', 'Be', 'B', 'C', 'N', 'O'
, 'F', 'Ne', 'Na', 'Mg', 'Al', 'Si', 'P', 'S', 'Cl'
, 'Ar', 'K', 'Ca',
8           'Sc', 'Ti', 'V', 'Cr', 'Mn', 'Fe', 'Co',
'Ni', 'Cu', 'Zn', 'Ga', 'Ge', 'As', 'Se', 'Br', 'Kr'
, 'Rb', 'Sr', 'Y',
9           'Zr', 'Nb', 'Mo', 'Tc', 'Ru', 'Rh', 'Pd'
, 'Ag', 'Cd', 'In', 'Sn', 'Sb', 'Te', 'I', 'Xe', '
Cs', 'Ba', 'La', 'Ce',
10          'Pr', 'Nd', 'Pm', 'Sm', 'Eu', 'Gd', 'Tb'
, 'Dy', 'Ho', 'Er', 'Tm', 'Yb', 'Lu', 'Hf', 'Ta', '
W', 'Re', 'Os', 'Ir',
11          'Pt', 'Au', 'Hg', 'Tl', 'Pb', 'Bi', 'Po'
, 'At', 'Rn', 'Fr', 'Ra', 'Ac', 'Th', 'Pa', 'U', '
Np', 'Pu', 'Am', 'Cm',
12          'Bk', 'Cf', 'Es', 'Fm', 'Md', 'No', 'Lr'
, 'Rf', 'Db', 'Sg', 'Bh', 'Hs', 'Mt', 'Ds', 'Rg', '
Cn', 'Nh', 'Fl', 'Mc',
13          'Lv', 'Ts', 'Og']
14 First = []
15 Second = []
16 Atom = []
17
18 def Raw():
19     for a in range(0, 20):
20         Num.append(str(a+1))
21         if a >= 4:
22             h.append(a)
23             if a >= 10 and a % 2 == 0:
24                 k.append(a)
25             for b in range(0, len(Sub)):
26                 if b <= a and b < 2:
27                     Sequence.append(Num[a] + Sub[b])

```

```

28
29 def Process():
30     for i in range(0, len(Sequence)):
31         if Sequence[i][1] != ('0' or '1' or '2' or
32         '3' or '4' or '5' or '6' or '7' or '8' or '9'):
33             if int(Sequence[i][0]) >= 4 and
34             Sequence[i][1] == 's':
35                 Sequence.insert(i+1, Sequence[i-1][
36                 0]+'d')
37                 if int(Sequence[i][0]) >= 6 and
38                 Sequence[i][1] == 's':
39                     Sequence.insert(i+1, Sequence[i-2][
40                     0]+'f')
41                     if int(Sequence[i][0]) >= 8 and
42                     Sequence[i][1] == 's':
43                         Sequence.insert(i+1, Sequence[i-3][
44                         0]+'g')
45
46     for p in range(29, len(Sequence)+60):
47         if len(Sequence[p]) == 2:
48             continue
49         else:
50             if int(Sequence[p][0])*10+int(Sequence[
51             p][1]) >= 10 and Sequence[p][2] == 's':
52                 for n in range(1,4):
53                     if len(Sequence[p-n]) == 2:
54                         Sequence.insert(p+1,
55                         Sequence[p-n][0]+Sub[n+1])
56                     else:
57                         Sequence.insert(p+1,
58                         Sequence[p-n][0]+Sequence[p-n][1]+Sub[n+1])
59                 for u in range(0,5):
60                     if int(Sequence[p][0])*10+int(
61                     Sequence[p][1]) >= k[u] and Sequence[p][2] == 's':
62                         if len(Sequence[p-h[u]]) == 2:
63                             Sequence.insert(p+1,
64                             Sequence[p-h[u]][0]+Sub[h[u]+1])
65                         else:
66                             Sequence.insert(p+1,
67                             Sequence[p-h[u]][0]+Sequence[p-h[u]][1]+Sub[h[u]+1
68                             ])

```

```
55
56 def Check():
57     atom = input("please input the atomic number or
    atomic symbol of the element: ")
58     if atom in Period:
59         atom = Period.index(atom) + 1
60     else:
61         atom = int(atom)
62     return atom
63
64 def Run(atom):
65     atom_w = Period[atom - 1]
66     Atom = []
67     for j in range(0, len(Sequence)):
68         for v, u in zip(range(2, 19, 4), range(0, 5
    )):
69             if Sub[u] in Sequence[j] and atom - v
    >= 0:
70                 Atom.append(Sequence[j] + str(v))
71                 atom -= v
72                 break
73             else:
74                 if atom > 0:
75                     Atom.append(Sequence[j] + str(atom
    ))
76                 break
77     print("The electron configuration of", atom_w,
    'is', Atom)
78     return Atom
79
80 m = Check()
81
82 Raw()
83 Process()
84 Run(m)
85
86 # print(len(Sequence))
```