//

Conceptual Programming 과제 #2

//

1. Electric wire is a cylindrical conductor covered by an insulating material. The resistance of a piece of wire is given by the formula

$$R = \frac{\rho L}{A} = \frac{4\rho L}{\pi d^2}$$

where ρ is the resistivity of the conductor, and L, A, and d are the length, cross-sectional area, and diameter of the wire. The resistivity of copper is 1.678×10^{-8} Ω m. The wire diameter, d, is commonly specified by the American wire gauge (AWG), which is an integer, n. The diameter of an AWG n wire is given by the formula

$$d = 0.127 \times 92^{\frac{36-n}{39}}\ mm$$

Write a function

```
def diameter(wireGauge)
```

that accepts the wire gauge and returns the corresponding wire diameter. Write another function

```
def copperWireResistance(length, wireGauge)
```

that accepts the length and gauge of a piece of copper wire and returns the resistance of that wire. The resistivity of aluminum is 2.82×10^{-8} Ω m. Write a third function

```
def aluminumWireResistance(length, wireGauge)
```

that accepts the length and gauge of a piece of aluminum wire and returns the resistance of that wire.
Write a program to test these functions.

Sample output:

```
Enter n (wire gauge): 30
Enter L (length): 6000
The resistance of copper wire is 0.0020
The resistance of aluminum wire is 0.0033
```

2. Implement the following algorithm to construct magic $n \times n$ squares; it works only if n is odd.

```
Set row=n-1, column=n/2.
For k=1 … n*n
  Place k at [row][column].
  Increment row and column.
  If the row or column is n, replace it with 0.
  If the element at [row][column] has already been filled,
    Set row and column to their previous values.
    Decrement row.
```

Here is the 5 × 5 square that you get if you follow this algorithm:-

11	18	25	2	9
10	12	19	21	3
4	6	13	20	22
23	5	7	14	16
17	24	1	8	15

Write a program whose input is the number n and whose output is the magic square of order n if n is odd.

Sample output:

```
Enter n: 5

-----------------------------------
|  11  |  18  |  25  |   2  |   9  |
-----------------------------------
|  10  |  12  |  19  |  21  |   3  |
-----------------------------------
|   4  |   6  |  13  |  20  |  22  |
-----------------------------------
|  23  |   5  |   7  |  14  |  16  |
-----------------------------------
|  17  |  24  |   1  |   8  |  15  |
-----------------------------------
```

Bonus:

** Draw with graphics.py

** Algorithm that works with even n