Problem 1

定义一个叫 Print_values 的函数，包含 3 个参数：a，b，和 c。首先判断 a 是否大于 b，再判断 b 和 c，a 和 c 之间的大小关系，按照 a、b、c 的大小关系，把三个数按照从大到小的顺序打印出来。

Problem 2（夏侯露钰 explained to me what append() mean）

2.1：定义一个空列表 M，第一次循环利用 append()将 n 个元素为 0 的列表添加到 M 中 m 次，下一个 for 循环使用 M[i][j]=random.randint(0,50)将每个元素替换为 0 到 50 的随机整数。

2.2：rows 和 cols 表征 M1 的行数和 M2 的列数，后创建一个行数和列数为 rows 和 cols 的零矩阵，最后 for 循环计算 M1 的第 i 行和 M2 的第 j 列的元素的点积，输出矩阵乘法运算结果。（只能计算 M1 的行数与 M2 的列数相等的矩阵乘法）

Problem 3

Pascal's triangle 每一项都是二项式系数，定义一个初始为 1 的列表，利用二项式系数公式$C(n,k) == C(n,k-1) * (n-k+1)/k$。进入 for 循环，在每次循环中，第 i 项等于上一行的第 i 个元素乘以(k - i + 1) // i，利用 append()将计算结果添加到列表中。

Problem 4

求解 1 到 x 的最小步骤数，即求解 x 到 1 的最小步骤数。当 x=1，2，3 时，固定步骤数为 0，1，2；当 x 大于 3，判断 x 是奇数还是偶数。对于奇数，只能选择减 1 这个操作故增加一步；对于偶数，选择 i-1 或 i/2 这两个操作，增加一步判断 i-1 还是 i/2 更小，输出最小步骤数。

Problem 5

5.1：先定义 1-9 数字的集合，再定义运算符号即 list_symbol = ["0","1","2"]，其中 0 代表不做运算，1 代表加，2 代表减。利用循环穷举所有的符号组合，123456789 里面一共可以插入 8 个字符，依次计算所有的组合，即把每种符号组合添加进数字中，并转为加减运算，判断结果是否符合要求，输出符合的等式与等式数目。

5.2：初始化 Total_solutions，把 1 至 100 匹配等式数目添加进入 Total_solutions，寻找 Total_solutions 的最大值与最小值，并找到对应的数，并绘图（图如下）。（the usage of enumerate() and Matplotlib Pyplot from https://www.runoob.com/python3/python3-tutorial.html）