```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
# Problem1
import random
def Print_values(a,b,c):
    if type(a)==str or type(b)==str or type(c)==str:
        print(type(a),type(b),type(c))
        print ("请输入整数")
    else:
        if a>b:
            if b>c:
                print(a,b,c)
            elif a>c:
                print(a,c,b)
            else:
                print(c,a,b)
        elif a>c:
            print(b,a,c)
        elif b>c:
            print(b,c,a)
        else:
            print(c,b,a)

for i in range(10):
  a = random.randint(1, 10)
  b=random.randint(1, 10)
  c=random.randint(1, 10)
  Print_values(a,b,c)

#Problem2
import numpy as np
M1 = np.random.randint(1,50,size=(5,10))
M2 = np.random.randint(1,50,size=(10,5))
print(M1)
print(M2)
def Matrix_multip(M1,M2):
    a=M1.shape[0]
    b=M1.shape[1]
    c=M2.shape[1]
    M3=np.empty(shape=(a,c))
    for i in range(a):
        for j in range(c):
            for k in range(b):
                M3[i][j]=M3[i][j]+M1[i][k]*M2[k][j]
    return M3
print(Matrix_multip(M1,M2))

#Problem 3
def Pascal_triangle(k):
    triangle = []
```

```
    for i in range(k):
        row = [1] * (i + 1)
        if i > 1:
            for j in range(1, i):
                row[j] = triangle[i - 1][j - 1] + triangle[i - 1][j]
        triangle.append(row)
    return triangle

def print_pascals_triangle(triangle):
    for row in triangle:
        print(" ".join(map(str, row)).center(len(triangle[-1]) * 3))

k= 5  # 指定要生成的行数
pascals_triangle = Pascal_triangle(k)
print_pascals_triangle(pascals_triangle)

#problem4
import random
import math
def Least_moves(y):
    for i in range(2,100):
        z=math.pow(2, i)
        if y>z:
            i=i+1
        else:
            break
    s=y-(2**(i-1))+(i-1)
    return s

for i in range(10):
  y = random.randint(1, 100)
  p=Least_moves(y)
  print(p)

#probelm5 [1]
def make_50(self) -> list:
    str = list(self)
    spr = ['+', '-', '']
    sum_50 = []
    for a in spr:
        for b in spr:
            for c in spr:
                for d in spr:
                    for e in spr:
                        for f in spr:
                            for g in spr:
                                for h in spr:
                                    sum = str[0] + a + str[1] + b + str[2] + c +
str[3] + d + str[4] + e + str[5] + f + str[6] + g + str[7] + h + str[8]
                                    if eval(sum) == 50:
                                        print(sum+"=50")
                                        sum_50.append(sum)
    return sum_50
```

```
if __name__ == "__main__":
    results = make_50("123456789")
    for result in results:
        assert eval(result) == 50
    print("OK")

#probelm5 [2]
def make_50(self,x) -> list:
    str = list(self)
    spr = ['+', '-', '']
    i=0
    for a in spr:
        for b in spr:
            for c in spr:
                for d in spr:
                    for e in spr:
                        for f in spr:
                            for g in spr:
                                for h in spr:
                                    sum = str[0] + a + str[1] + b + str[2] + c +
str[3] + d + str[4] + e + str[5] + f + str[6] + g + str[7] + h + str[8]
                                    if eval(sum) == x:
                                        i=i+1
    return i

Total_solutions={}
for x in range(1,101):
    results = make_50("123456789",x)
    Total_solutions[x] = results
print(tinydict)
min_Total_solutions = min(zip(Total_solutions.values(), Total_solutions.keys()))
max_Total_solutions = max(zip(Total_solutions.values(), Total_solutions.keys()))
print(min_Total_solutions)
print(max_Total_solutions)
```