所有代码参考文件为 chat gpt4

1.

```
def Print_values(a,b,c):
    if(a>b):
        if(b>c):
            print(a,b,c)
        else:
            if(a>c):
                print(a,c,b)
            else:
                print(c,a,b)
    else:
        if(b>c):
            if (a > c):
                print(b, a, c)
            else:
                print(b, c, a)
        else:
            print(c,b,a)

if __name__ == '__main__':
    a, b, c = map(int, input().split())
    Print_values(a,b,c)
```

2.

```
import random
import numpy

def Matrix_multip(M1,M2):
    M = numpy.zeros((5, 5))
    for i in range(5):
        for j in range(5):
            for k in range(10):
                M[i][j]+=M1[i][k]*M2[k][j]
    return M

if __name__ == '__main__':
    M1=numpy.zeros((5,10))
    M2=numpy.zeros((10,5))
    for i in range(5):
        for j in range(10):
            M1[i][j]=random.randint(0,50)
            M2[j][i]=random.randint(0,50)
    for i in range(5):
```

```
            for j in range(10):
                print(M1[i][j],end=" ")
            print('\n')
        print('\n')
        for i in range(10):
            for j in range(5):
                print(M2[i][j],end=" ")
            print('\n')
        print('\n')
        M=Matrix_multip(M1,M2)
        for i in range(5):
            for j in range(5):
                print(M[i][j],end=" ")
            print('\n')
```

3.

```
def Pascal_triangle(k):
    list=[0,1,0]
    list2=[]
    for i in range(k):
        list2.append(0)
        left=0
        right=1
        while right<len(list):
            list2.append(list[left]+list[right])
            left+=1
            right+=1
        list2.append(0)
        list.clear()
        for item in list2:
            list.append(item)
        list2.clear()
    return list[1:-1]

if __name__ == '__main__':
    k=int(input())
    ret=Pascal_triangle(k)
    print(ret)

def mymin(x,y):
    if(x<=y):
        return x
    else:
        return y
```

print Pascal_triangle(100):
print Pascal_triangle(200):

4.

```
def Least_moves(x):
    dp=[0]*(x+1)
    dp[1]=0
    for i in range(2,x+1):
        if(i%2==0):
            dp[i]=mymin(dp[i-1],dp[int(i/2)])+1
        else:
            dp[i]=dp[i-1]+1
    return dp[x]

if __name__ == '__main__':
    x=int(input())
    ret=Least_moves(x)
    print(ret)
```

5.

5.1

```
origin=['1','2','3','4','5','6','7','8','9']

def str2int(list,i,j):
    ret=0
    for idx in range(i,j+1):
        ret=ret*10+int(list[idx])
    return ret

def cal(list):
    ret=0
    left=0
    flag=0
    length=len(list)
    for i in range(length):
        if(list[i]=='+' or list[i]=='-'):
            if(flag==0):
                ret=ret+str2int(list,left,i-1)
            elif(flag==1):
                ret=ret-str2int(list,left,i-1)
            left=i+1
            if(list[i]=='+'):
                flag=0
            if(list[i]=='-'):
                flag=1
```

```
    if (flag == 0):
        ret = ret + str2int(list,left, length-1)
    elif (flag == 1):
        ret = ret - str2int(list,left, length - 1)
    return ret

number=[]
data=[]
result=[]
5.2
def insert(list,idx,type):
    if(idx>=len(list)):
        if(type==0):
            data.append(list)
            result.append(cal(list))
    else:
        if(type==1):
            list.insert(idx,'+')
        elif(type==2):
            list.insert(idx, '-')
        tmp1 = list.copy()
        tmp2 = list.copy()
        tmp3 = list.copy()
        if(type!=0):
            insert(tmp1, idx + 2, 0)
            insert(tmp2, idx + 2, 1)
            insert(tmp3, idx + 2, 2)
        else:
            insert(tmp1, idx + 1, 0)
            insert(tmp2, idx + 1, 1)
            insert(tmp3, idx + 1, 2)
        if(type!=0):
            del list[idx]

def bti():
    for i in range(9):
        number.append([])
        for j in range(9):
            number[i].append(str2int(origin,i,j))
    insert(['1','2','3','4','5','6','7','8','9'],1,0)
    insert(['1','2','3','4','5','6','7','8','9'], 1, 1)
    insert(['1','2','3','4','5','6','7','8','9'], 1, 2)

def Find_expression(x):
```

```
        list=[]
        length=len(result)
        for idx in range(length):
            tmp=''
            if(x==result[idx]):
                for item in data[idx]:
                    tmp+=item
                tmp=tmp+'='+str(x)
                list.append(tmp)
        return list

if __name__ == '__main__':
        bti()
        x=Find_expression(50)
        for item in x:
            print(item)
        Total_solutions=[]
        for i in range(1,101):
            Total_solutions.append(len(Find_expression(i)))
        min=Total_solutions[0]
        max=Total_solutions[0]
        for item in Total_solutions:
            if(item<min):
                min=item
            if(item>max):
                max=item
        for i in range(100):
            if(Total_solutions[i]==min):
                print(i+1,end=" ")
        print()
        for i in range(100):
            if(Total_solutions[i]==max):
                print(i+1,end=" ")
```