**习1：青蛙过河（frog 蓝桥杯 2022 省 A）**

**问题描述**：小青蛙住在一条河边，它想到河对岸的学校去学习。小青蛙打算经过河里的石头跳到对岸。河里的石头排成了一条直线，小青蛙每次跳跃必须落在一块石头或者岸上。不过，每块石头有一个高度，每次小青蛙从一块石头起跳，这块石头的高度就会下降 1，当石头的高度下降到 0时小青蛙不能再跳到这块石头上（某次跳跃后使石头高度下降到 0 是允许的)。

小青蛙一共需要去学校上 x 天课，所以它需要往返 2x 次。当小青蛙具有一个跳跃能力 y 时，它能跳不超过 y 的距离。

请问小青蛙的跳跃能力至少是多少才能用这些石头上完 x 次课。。

**输入格式：**第一行包含两个整数n,x, 分别表示河的宽度和小青蛙需要去学校的天数。请注意 2x 才是实际过河的次数。

第二行包含 n−1 个非负整数 H1​,H2​,⋯,Hn−1​, 其中Hi>0 表示在河中与小青蛙的家相距 i 的地方有一块高度为 Hi​ 的石头，Hi​=0 表示这个位置没有石头。

**输出格式**：输出一行, 包含一个整数, 表示小青蛙需要的最低跳跃能力。

**输入样例**：

5 1

1 0 1 0

**输出样例**：

4

**样例说明**：由于只有两块高度为 1 的石头，所以往返只能各用一块。第 1 块石头和对岸的距离为 4，如果小青蛙的跳跃能力为 3 则无法满足要求。所以小青蛙最少需要 4 的跳跃能力。

**数据范围**：

对于 30% 的数据，n≤100;

对于 60% 的数据，n≤1000;

对于100%的数据，1≤n≤105,1≤x≤109,0≤Hi≤104 。

https://www.luogu.com.cn/problem/P8775

**习2：快速求和(quicksum，搜索剪枝)**

问题描述： 给定一个数字字符串，在字符串中的某些位置插入一些加号。在所有的加号都插入后，就象做普通加法那样来求值，问最少需要插入多少加号可以使得该值等于一个给定的目标数字。

例如，考虑字符串"303"和目标数字 6，最佳方法不是"3+0+3"，而是"3+03"。能这样做是因为一个数的前导零不会改变它的大小。

编程实现这个算法。

输入格式：第一行包含一个字符串 S( 1 <= S 的长度 <= 40 ) 和一个整数 N( N <= 1000 )，用单个空格分隔。

输出格式：一行一个整数 K，表示需要最少需要插入多少加号。如果无解，则输出 -1。

输入样例：

2222 8

输出样例：

3

数据规模：

对于 80%的数据，1 <= S 的长度 <= 10。

对于 100%的数据，1 <= S 的长度 <= 40,N <= 1000。

分析：对于朴素的回溯搜索，由于字符串长度不超过40，最多有39个位置可以放置加号，这样最多就有239种可能性。显然，不加优化的搜索会出现超时。

如何优化呢？很容易想到的优化有三个：

1）如果搜索到当前位置i时，已经添加的加号个数p大于最优解ans，则不管以后怎样，都不可能得到最优的解，因此剪枝，返回。

2）如果搜索到当前位置i时，已经算出的部分和大于N，则以后的数最小，也不可能恰好凑成N，因此剪枝，返回。

3）如果搜索到当前位置i时，已经算出的部分和再加上i+1～N这一段的最大值（即i+1~N不添加加号），仍然小于N，则再搜索也没有意义，因此剪枝，返回。

另一个要注意的问题是40位的数字串在转化成整数时，会超过int甚至long long的存储范围。而本题的一个特点是要凑的数N本身并不太大（不超过106），所以可以设置一个无穷大INF（值可设为109），当要转化的数字串的值超过INF时，直接把值设为INF就行。这样做不会影响运算结果的正确性，也有效地避免了高精度运算。

有了上述优化，本题完全可以在规定时限内出解。

**习3：最大 M 子段和问题（maxsum HDU1024动态规划）**

**问题描述**：给定一个连续的数字序列A1、A2、A3、A4…Ax…An（1≤x≤n≤1000000，-32768≤Ax≤32767）。我们定义一个函数sum（i,j）=Ai+…+Aj（1≤i≤j≤n）。现在给定一个整数m（m > 0），你的任务是找到m对i和j，它们构成sum（i1,j1）+sum（i2,j2）+sum（i3,j3）+…+sum（im，jm）的最大值（不允许ix≤iy≤jx或ix≤jy≤jx）。输出该最大值。

**输入格式**：

第一行包含两个整数n和m。

第二行包含 n 个用空格分隔的整数，分别表示A1,A2,A3…An。

**输出格式**：

输出一个整数，表示最大M子段和。

**输入样例1**：

3 1

1 2 3

**输出样例1**：

6

**输入样例2**：

6 2

-1 4 -2 3 -2 3

**输出样例2**：

8

**数据范围**：1 ≤ n ≤ 1,000,000, -32768 ≤ a*i* ≤ 32767。

**习4：分金币（coin UVa11300）**

**问题描述**：圆桌旁坐着 n 个人，每人都有一定数量的金币，金币总数能被 n 整除。每个人可以给他左右相邻的人一些金币，最终使得每个人的金币数目相等。你的任务是求出被转手的金币数量的最小值。比如，n=4，且4个人的金币数量分别是1,2,5,4时，只需转移4枚金币（第3个人给第2个人两枚金币，第2个人和第4个人分别给第1个人1枚金币）即可实现每人手中的金币数目相等。

**输入格式**：第一行包含一个整数 n。第二行包含 n 个整数，表示按逆时针顺序给出的每个人拥有的金币数。

**输出格式**：输出被转手金币数量的最小值。输入保证这个值在 64 位无符号整数范围内。

**输入样例**：

4

1 2 5 4

**输出样例**：

4