1、ACM 编程 执行时长

题目描述：为了充分发挥 GPU 算力，需要尽可能多的将任务交给 GPU 执行，现在有一个任务数组，数组元素表示在这 1 秒内新增的任务个数且每秒都有新增任务，假设 GPU 最多一次执行 n 个任务，一次执行耗时 1 秒，在保证 GPU 不空闲情况下，最少需要多长时间执行完成

输入描述：第一个参数为 GPU 一次最多执行的任务个数，取值范围[1, 10000]

第二个参数为任务数组长度，取值范围[1, 10000]

第三个参数为任务数组，数字范围[1, 10000]

输出描述：执行完所有任务最少需要多少秒

```
示例

示例1
输入：3
      5
      1 2 3 4 5
输出：6
说明：一次最多执行3个任务，最少耗时6s

示例2
输入：4
      5
      5 4 1 1 1
输出：5
说明：一次最多执行4个任务，最少耗时5s
```

2、ACM 编程 字符串序列判定

题目描述：输入两个字符串 S 和 L，都只包含英文小写字母。S 长度<=100，L 长度<=500,000。判定 S 是否是 L 的有效字串。

判定规则：S 中的每个字符在 L 中都能找到（可以不连续），且 S 在 L 中字符的前后顺序与 S 中顺序要保持一致。（例如，S="ace"是 L="abcde"的一个子序列且有效字符是 a、c、e，而"aec"不是有效子序列，且有效字符只有 a、e）

输入描述：输入两个字符串 S 和 L，都只包含英文小写字母。S 长度<=100，L 长度<=500,000。先输入 S，再输入 L，每个字符串占一行。

输出描述：S 串最后一个有效字符在 L 中的位置。（首位从 0 开始计算，无有效字符返回-1）

```
示例

示例1
输入：ace
      abcde
输出：4
说明：

示例2
输入：fgh
      abcde
输出：-1
说明：
```

3、完善核心代码编程 路口最短时间问题

题目描述：假定街道是棋盘型的，每格距离相等，车辆通过每格街道需要时间均为 timePerRoad；街道的街口（交叉点）有交通灯，灯的周期 T(=lights[row][col])各不相同；车辆可直行、左转和右转，其中直行和左转需要等相应 T 时间的交通灯才可通行，右转无需等待。

现给出 n*m 个街口的交通灯周期，以及起止街口的坐标，计算车辆经过两个街口的最短时间。

其中：

1）起点和终点的交通灯不计入时间，且可以任意方向经过街口
2）不可超出 n*m 个街口，不可跳跃，但边线也是道路（即 lights[0][0] -> lights[0][1] 是有效路径）
入口函数定义:

```
/**
* lights : n*m 个街口每个交通灯的周期，值范围[0,120]，n 和 m 的范围为[1,9]
* timePerRoad : 相邻两个街口之间街道的通过时间,范围为[0,600]
* rowStart : 起点的行号
* colStart : 起点的列号
* rowEnd : 终点的行号
* colEnd : 终点的列号
* return : lights[rowStart][colStart] 与 lights[rowEnd][colEnd] 两个街口之间的最短通行时间
*/
int calcTime(int[][] lights,int timePerRoad,int rowStart,int colStart,int rowEnd,int colEnd)
```

```
示例

示例1
输入：[[1,2,3],[4,5,6],[7,8,9]],60,0,0,2,2
输出：245
说明：行走路线为 (0,0) -> (0,1) -> (1,1) -> (1,2) -> (2,2) 走了4格路，2个右转，1个左转，共耗时 60+0+60+5+60+0+60=245
```

1、ACM 编程 最多购买宝石数目

题目描述：橱窗里有一排宝石，不同的宝石对应不同的价格，宝石的价格标记为 gems[i],0<=i<n, n = gems.length
宝石可同时出售 0 个或多个，如果同时出售多个，则要求出售的宝石编号连续；例如客户最大购买宝石个数为 m，购买的宝石编号必须为
gems[i],gems[i+1]...gems[i+m-1](0<=i<n,m<=n)
假设你当前拥有总面值为 value 的钱，请问最多能购买到多少个宝石,如无法购买宝石，则返回 0.
输入描述：第一行输入 n，参数类型为 int，取值范围：[0,10]，表示橱窗中宝石的总数量。之后 n 行分别表示从第 0 个到第 n-1 个宝石的价格，即 gems[0]到 gems[n-1]的价格，类型为 int，取值范围：(0,1000]。
之后一行输入 v，类型为 int，取值范围：[0,10]表示你拥有的钱。
输出描述：输出 int 类型的返回值，表示最大可购买的宝石数量。

```
示例1
输入：7
      8
      4
      6
      3
      1
      6
      7
      10
输出：3
说明：
      gems = [8,4,6,3,1,6,7], value = 10

      最多购买的宝石为gems[2]至gems[4]或者gems[3]至gems[5]
```

示例2
输入：0
1
输出：0
说明：

gems = [], value = 1

因为没有宝石，所以返回0

示例3
输入：9
6
1
3
1
8
9
3
2
4
15
输出：4
说明：

gems = [6, 1, 3, 1, 8, 9, 3, 2, 4], value = 15

最多购买的宝石为gems[0]至gems[3]

示例4
输入：9
1
1
1
1
1
1
1
1
1
10
输出：9
说明：

gems = [1, 1, 1, 1, 1, 1, 1, 1, 1], value = 10

最多购买的宝石为gems[0]至gems[8]，即全部购买

2、ACM 编程 用连续自然数之和来表达整数

题目描述：一个整数可以由连续的自然数之和来表示。给定一个整数，计算该整数有几种连续自然数之和的表达式，且打印出每种表达式。

输入描述：一个目标整数 T (1 <=T<= 1000)

输出描述：该整数的所有表达式和表达式的个数。如果有多种表达式，输出要求为：

1.自然数个数最少的表达式优先输出

2.每个表达式中按自然数递增的顺序输出，具体的格式参见样例。在每个测试数据结束时，输出一行"Result:X"，其中 X 是最终的表达式个数。

示例 展开

示例1
输入：9
输出：9=9
9=4+5
9=2+3+4
Result:3
说明：整数 9 有三种表示方法，第1个表达式只有1个自然数，最先输出，第2个表达式有2个自然数，第2次序输出，第3个表达式有3个自然数，最后输出。每个表达式中的自然数都是按递增次序输出的。
数字与符号之间无空格

示例2
输入：10
输出：10=10
10=1+2+3+4
Result:2
说明：

3、ACM 编程 5G 网络建设

题目描述：现需要在某城市进行 5G 网络建设，已经选取 N 个地点设置 5G 基站，编号固定为 1 到 N，接下来需要各个基站之间使用光纤进行连接以确保基站能互联互通，不同基站之间架设光纤的成本各不相同，且有些节点之间已经存在光纤相连，请你设计算法，计算出能联通这些基站的最小成本是多少。

注意：基站的联通具有传递性，入基站 A 与基站 B 架设了光纤，基站 B 与基站 C 也架设了光纤，则基站 A 与基站 C 视为可以互相联通

输入描述：第一行输入表示基站的个数 N，其中 0<N<=20

第二行输入表示具备光纤直连条件的基站对的数目 M，其中 0<M<N*(N-1)/2

从第三行开始连续输入 M 行数据，格式为 X Y Z P，其中 X Y 表示基站的编号，0<X<=N，0<Y<=N 且 x 不等于 y，Z 表示在 X Y 之间架设光纤的成本，其中 0<Z<100，P 表示是否已存在光纤连接，0 表示未连接，1 表示已连接

输出描述：如果给定条件，可以建设成功互联互通的 5G 网络，则输出最小的建设成本；如果给定条件，无法建设成功互联互通的 5G 网络，则输出-1

示例

示例1
输入：3
3
1 2 3 0
1 3 1 0
2 3 5 0
输出：4
说明：只需要在1,2以及2,3基站之间铺设光纤，其成本为3+1=4

示例2
输入：3
1
1 2 5 0
输出：-1
说明：3基站无法与其他基站连接，输出-1

示例3

输入：3
3
1 2 3 0
1 3 1 0
2 3 5 1
输出：1
说明：2,3基站已有光纤相连，只有要再1,3站点之间铺设光纤，其成本为1

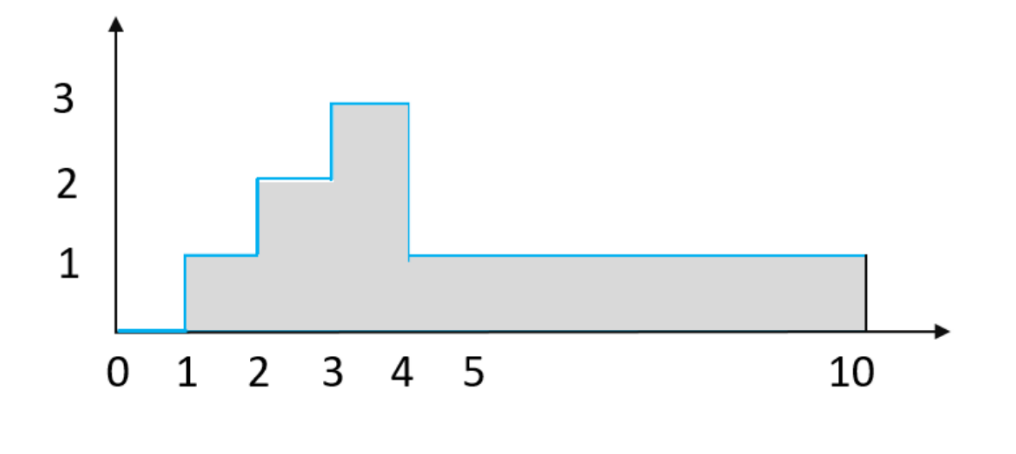
1、ACM 编程 计算面积

题目描述：绘图机器的绘图笔初始位置在原点（0,0），机器启动后其绘图笔按下面规则绘制直线：

1）尝试沿着横向坐标轴正向绘制直线，直到给定的终点值 E。
2) 期间可通过指令在纵坐标轴方向进行偏移，并同时绘制直线，偏移后按规则 1 绘制直线；
指令的格式为 X offsetY，表示在横坐标 X 沿纵坐标方向偏移，
offsetY 为正数表示正向偏移，为负数表示负向偏移。
给定了横坐标终点值 E、以及若干条绘制指令，请计算绘制的直线和横坐标轴、以及 X=E 的直线组成图形的面积。
输入描述：首行为两个整数 N E，表示有 N 条指令，机器运行的横坐标终点值 E。
接下来 N 行，每行两个整数表示一条绘制指令 X offsetY，用例保证横坐标 X 以递增排序方式出现，且不会出现相同横坐标 X。
取值范围：0 < N <= 10000, 0 <= X <= E <=20000, -10000 <= offsetY <= 10000。
输出描述：一个整数，表示计算得到的面积，用例保证，结果范围在 0~4294967295 内

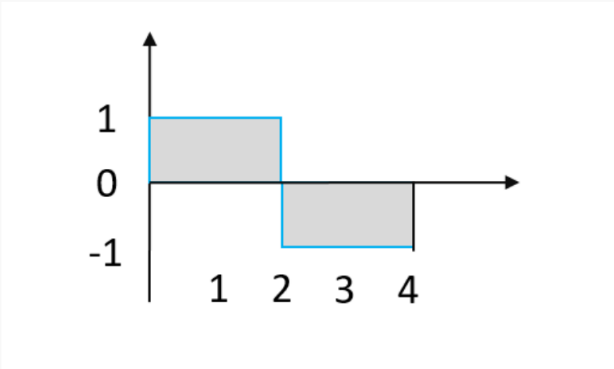
示例

示例1
输入：4 10
1 1
2 1
3 1
4 -2
输出：12
说明：通过操作机器最后绘制了如下图形（蓝色为绘制笔绘制的直线）



计算图中阴影部分面积，其值为1*1+2*1+3*1+1*6=12

示例2
输入：2 4
0 1
2 -2
输出：4
说明：通过操作机器最后绘制了如下图形



计算图中阴影部分面积，其值为1*2+1*2=4

2、ACM 编程 游戏分组

题目描述：部门准备举办一场王者荣耀表演赛，有 10 名游戏爱好者参与，分为两队，每队 5 人。每位参与者都有一个评分，代表着他的游戏水平。为了表演赛尽可能精彩，
我们需要把 10 名参赛者分为实力尽量相近的两队。一队的实力可以表示为这一队 5 名队员的评分总和。
现在给你 10 名参与者的游戏水平评分，请你根据上述要求分队，最后输出这两组的实力差绝对值。
例：10 名参赛者的评分分别为 5 1 8 3 4 6 7 10 9 2，分组为（1 3 5 8 10）（2 4 6 7 9），两组实力差最小，差值为 1。有多种分法，但实力差的绝对值最小为
1。
输入描述：10 个整数，表示 10 名参与者的游戏水平评分。范围在[1, 10000]之间
输出描述：1 个整数，表示分组后两组实力差绝对值的最小值。

```
示例

示例1
输入：1 2 3 4 5 6 7 8 9 10
输出：1
说明：10名队员分成两组，两组实力差绝对值最小为1。
```

3、ACM 编程 电脑病毒感染

题目描述：一个局域网内有很多台电脑，分别标注为 0 - N-1 的数字。相连接的电脑距离不一样，所以感染时间不一样，感染时间用 t 表示。
其中网络内一个电脑被病毒感染，其感染网络内所有的电脑需要最少需要多长时间。如果最后有电脑不会感染，则返回-1
给定一个数组 times 表示一个电脑把相邻电脑感染所用的时间。
如图：path[i]= {i,j, t} 表示电脑 i->j 电脑 i 上的病毒感染 j，需要时间 t。
输入描述：4
3
2 1 1
2 3 1
3 4 1
2
输出描述：2
补充说明：第一个参数:局域网内电脑个数 N 1<=N<=200;
第二个参数：总共多少条网络连接
第三个 1 2 1 表示 1->2 时间为 1
第七行：表示病毒最开始所在的电脑号 1

```
示例

示例1
输入：4
      3
      2 1 1
      2 3 1
      3 4 1
      2
输出：2
说明：
```

1、ACM 编程 精准核酸检测

题目描述：为了达到新冠疫情精准防控的需要，为了避免全员核酸检测带来的浪费，需要精准圈定可能被感染的人群。

现在根据传染病流调以及大数据分析，得到了每个人之间在时间、空间上是否存在轨迹的交叉。

现在给定一组确诊人员编号（X1, X2, X3, ... Xn）,在所有人当中，找出哪些人需要进行核酸检测，输出需要进行核酸检测的人数。（注意：确诊病例自身不需要
再做核酸检测）

需要进行核酸检测的人，是病毒传播链条上的所有人员，即有可能通过确诊病例所能传播到的所有人。

例如：A 是确诊病例，A 和 B 有接触、B 和 C 有接触、C 和 D 有接触、D 和 E 有接触，那么 B\C\D\E 都是需要进行核酸检测的人。

输入描述：第一行为总人数 N

第二行为确诊病例人员编号（确诊病例人员数量<N），用逗号分割

第三行开始，为一个 N*N 的矩阵，表示每个人员之间是否有接触，0 表示没有接触，1 表示有接触。

输出描述：整数：需要做核酸检测的人数

补充说明：人员编号从 0 开始

0<N<100

```
示例

示例1
输入：5
        1,2
        1,1,0,1,0
        1,1,0,0,0
        0,0,1,0,1
        1,0,0,1,0
        0,0,1,0,1
输出：3
说明：编号为1、2号的人员，为确诊病例。
        1号与0号有接触，0号与3号有接触。
        2号与4号有接触。
        所以，需要做核酸检测的人是0号、3号、4号，总计3人需要进行核酸检测。
```

2、ACM 编程 提取字符串中的最长数学表达式并计算

题目描述：提取字符串中的最长合法简单数学表达式，字符串长度最长的，并计算表达式的值。如果没有，则返回 0

简单数学表达式只能包含以下内容

0-9 数字，符号 +-*

说明：

1. 所有数字，计算结果都不超过 long
2. 如果有多个长度一样的，请返回第一个表达式的结果
3. 数学表达式，必须是最长的，合法的
4. 操作符不能连续出现，如 +--+1 是不合法的

输入描述：字符串

输出描述：表达式值

```
示例

示例1
输入：1-2abcd
输出：-1
说明：
```

3、完善核心代码编程 攀登者 2

攀登者喜欢寻找各种地图，并且尝试攀登到最高的山峰。

地图表示为一维数组，数组的索引代表水平位置，数组的高度代表相对海拔高度。其中数组元素 0 代表地面。

例如[0,1,4,3,1,0,0,1,2,3,1,2,1,0]，代表如下图所示的地图，地图中有两个山脉位置分别为 1,2,3,4,5 和 8,9,10,11,12,13，最高峰高度分别为 4,3。最高峰位置分别为 3,10。

一个山脉可能有多座山峰(高度大于两边，或者在地图边界)。

```
        4


       +---+
       | |
       | |3              3
       | |
       | +---+           -----
       |   |           | |
      2|   |          2| |   2
       |   |           | |
     +---+   |         ----+ |  +---+
     |     |         |   | | |
    1|     |1       1|   |1| |1
     |     |         |   | | |
   +---+     +---+   +---+   +---+  +---+
   |         |   |           |
  0|          |0  0|           |0
   |         |   |           |
+---+          +-------+            +---+

 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
```

2. 登山时会消耗登山者的体力(整数)，上山时，消耗相邻高度差两倍的体力，下坡时消耗相邻高度差一倍的体力，平地不消耗体力，登山者体力消耗到零时会
有生命危险。

例如，上图所示的山峰，从索引 0，走到索引 1，高度差为 1，需要消耗 2X1 = 2 的体力，从索引 2 高度 2 走到高度 4 索引 3 需要消耗 2X2 = 4 的体力。如果是从索
引 3 走到索引 4 则消耗 1X1 的体力。

3. 登山者体力上限为 999。

4. 登山时的起点和终点可以是地图中任何高度为 0 的地面例如上图中的 0,6,7,14 都可以作为登山的起点和终点

攀登者想要评估一张地图内有多少座山峰可以进行攀登，且可以安全返回到地面，且无生命危险。

例如上图中的数组，有 3 个不同的山峰，登上位置在 3 的山可以从位置 0 或者位置 6 开始，从位置 0 登到山顶需要消耗体力 1X2+1X2+2X2 = 8，从山顶返回到地面 0
需要消耗体力 2X1 + 1X1 + 1X1 = 4 的体力，按照登山路线 0->3->0 需要消耗体力 12。攀登者至少需要 12 以上的体力（大于 12）才能安全返回。

```
示例

示例1
输入：[0,1,4,3,1,0,0,1,2,3,1,2,1,0],13
输出：3
说明：登山者只能够登上位置10和12的山峰，7->10->7, 14->12>14

示例2
输入：[1,4,3],999

输出：0
说明：没有合适的起点和终点
```

1、ACM 编程 分披萨

题目描述：“吃货”和“馋嘴”两人到披萨店点了一份铁盘（圆形）披萨，并嘱咐店员将披萨按放射状切成大小相同的偶数扇形小块。但是粗心服务员将披萨切成了每块大
小都完全不同奇数块，且肉眼能分辨出大小。
由于两人都想吃到最多的披萨，他们商量了一个他们认为公平的分法：从“吃货”开始，轮流取披萨。除了第一块披萨可以任意选取以外，其他都必须从缺口开
始选。
他俩选披萨的思路不同。“馋嘴”每次都会选最大块的披萨，而且“吃货”知道“馋嘴”的想法。
已知披萨小块的数量以及每块的大小，求“吃货”能分得的最大的披萨大小的总和。
输入描述：第 1 行为一个正整数奇数 N，表示披萨小块数量。3 <= N < 500。
接下来的第 2 行到第 N+1 行（共 N 行），每行为一个正整数，表示第 i 块披萨的大小。1 <= i <= N。披萨小块从某一块开始，按照一个方向依次顺序编号为
1~N。每块披萨的大小范围为[1, 2147483647]。
输出描述：“吃货”能分得的最大的披萨大小的总和。

```
示例

示例1
输入：5
      8
      2
      10
      5
      7
输出：19
说明：此例子中，有5块披萨。每块大小依次为8、2、10、5、7。按照如下顺序拿披萨，可以使“吃货”拿到最多披萨：
      1、“吃货”拿大小为10的披萨
      2、“馋嘴”拿大小为5的披萨
      3、“吃货”拿大小为7的披萨
      4、“馋嘴”拿大小为8的披萨
      5、“吃货”拿大小为2的披萨
      至此，披萨瓜分完毕，“吃货”拿到的披萨总大小为10+7+2=19。
      可能存在多种拿法，以上只是其中一种。
```

2、ACM 编程 API 集群负载统计

题目描述：某个产品的 RESTful API 集合部署在服务器集群的多个节点上，近期对客户端访问日志进行了采集，需要统计各个 API 的访问频次，根据热点信息在服务器节点之
间做负载均衡，现在需要实现热点信息统计查询功能。

RESTful API 的由多个层级构成，层级之间使用 / 连接，如 /A/B/C/D 这个地址，A 属于第一级，B 属于第二级，C 属于第三级，D 属于第四级。
现在负载均衡模块需要知道给定层级上某个名字出现的频次，未出现过用 0 次表示，实现这个功能。
输入描述：第一行为 N，表示访问历史日志的条数，0 < N≤100。
接下来 N 行，每一行为一个 RESTful API 的 URL 地址，约束地址中仅包含英文字母和连接符 /，最大层级为 10，每层级字符串最大长度为 10。
最后一行为层级 L 和要查询的关键字。
输出描述：输出给定层级上，关键字出现的频次，使用完全匹配方式（大小写敏感）。

```
示例

示例1
输入：5
        /huawei/computing/no/one
        /huawei/computing
        /huawei
        /huawei/cloud/no/one
        /huawei/wireless/no/one
        2 computing
输出：2
说明：在第二层级上，computing出现了2次，因此输出2.

示例2
输入：5
        /huawei/computing/no/one
        /huawei/computing
        /huawei
        /huawei/cloud/no/one
        /huawei/wireless/no/one
        4 two
输出：0
说明：存在第四层级的URL上，没有出现two，因此频次是0
```

3、完善核心代码编程 会议室占用时间

题目描述：现有若干个会议，所有会议共享一个会议室，用数组表示每个会议的开始时间和结束时间，格式：[[会议 1 开始时间，会议 1 结束时间], [会议 2 开始时间，会议 2 结束时间]]。
请计算会议室占用时间段。
补充说明：会议室个数范围：[1, 100]
会议室时间段：[1, 24]

```
示例

示例1
输入：[[1,4],[2,5],[7,9],[14,18]]
输出：[[1,5],[7,9],[14,18]]
说明：时间段 [1,4] 和 [2,5] 重叠, 合并为 [1,5]

示例2
输入：[[1,4],[4,5]]
输出：[[1,5]]
说明：时间段 [1,4] 和 [4,5] 连续
```

1、ACM 编程 寻找身高相近的小朋友

题目描述：小明今年升学到小学一年级，来到新班级后发现其他小朋友们身高参差不齐，然后就想基于各小朋友和自己的身高差对他们进行排序，请帮他实现排序。
输入描述：第一行为正整数 H 和 N，0<H<200，为小明的身高，0<N<50，为新班级其他小朋友个数。
第二行为 N 个正整数 H1-HN，分别是其他小朋友的身高，取值范围 0<Hi<200（1<=i<=N），

且 N 个正整数各不相同。
输出描述：输出排序结果，各正整数以空格分割。和小明身高差绝对值最小的小朋友排在前面，和小明身高差绝对值最大的小朋友排在最后，如果两个小朋友和小明身高差
一样，则个子较小的小朋友排在前面。

示例

示例1
输入：100 10
95 96 97 98 99 101 102 103 104 105
输出：99 101 98 102 97 103 96 104 95 105
说明：小明身高100，班级学生10个，身高分别为95 96 97 98 99 101 102 103 104 105，按身高差排序后结果为：99 101 98 102 97 103 96 104 95 105。

2、ACM 编程 提取字符串中的最长数学表达式并计算

题目描述：提取字符串中的最长合法简单数学表达式，字符串长度最长的，并计算表达式的值。如果没有，则返回 0
简单数学表达式只能包含以下内容
0-9 数字，符号 +-*
说明：
1. 所有数字，计算结果都不超过 long
2. 如果有多个长度一样的，请返回第一个表达式的结果
3. 数学表达式，必须是最长的，合法的
4. 操作符不能连续出现，如 +--+1 是不合法的
输入描述：字符串
输出描述：表达式值

示例

示例1
输入：1-2abcd
输出：-1
说明：

3、ACM 编程 结队编程

题目描述：某部门计划通过结队编程来进行项目开发，已知该部门有 N 名员工，每个员工有独一无二的职级，每三个员工形成一个小组进行结队编程，结队分组规则如下：
从部门中选出序号分别为 i、j、k 的 3 名员工，他们的职级分别为 level[i]、level [j]、level [k]
结队小组需满足： level [i] < level [j] < level [k] 或者 level [i] > level [j] > level [k] ，其中 0 <= i < j < k < n
请你按上述条件计算可能组合的小组数量。同一员工可以参加多个小组。
输入描述：第一行输入：员工总数 n
第二行输入：按序号依次排列的员工的职级 level，中间用空格隔开
限制：
1 <= n <= 6000
1 <= level[i] <= 10^5
输出描述：可能组合的小组数量

示例

示例1
输入：4
1 2 3 4
输出：4
说明：可能结队成的组合 (1,2,3)、(1,2,4)、(1,3,4)、(2,3,4)

示例2
输入：3
5 4 7
输出：0
说明：根据结队条件，我们无法为该部门组建小组

1、ACM 编程 小华最多能得到多少克黄金

题目描述：小华按照地图去寻宝，地图上被划分成 m 行和 n 列的方格，横纵坐标范围分别是 [0, n−1]和[0, m-1]。在横坐标和纵坐标的数位之和不大于 k 的方格中存在黄金（每个方格中仅存在一克黄金），但横坐标和纵坐标之和大于 k 的方格存在危险不可进入。小华从入口(0,0)进入，任何时候只能向左，右，上，下四个方向移动一格。请问小华最多能获得多少克黄金？

输入描述：坐标取值范围如下：

0<=m<=50

0<=n<=50

k 的取值范围如下：

0<=k<=100

输入中包含 3 个字数，分别是 m，n，k

输出描述：最多能获得多少克黄金

示例1
输入：40 40 18
输出：1484
说明：
当k为18时，小华能够进入方格（10,10），因为1+0+1+0 = 2。 但是，他不能进入方格（36,38），因为3+6+3+8 = 20

示例2
输入：4 5 7
输出：20
说明：如图每个单元格中的数位之和均不大于7，都是符合要求的，所以可以最多可获得20克黄金

	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	5
2	2	3	4	5	6
3	3	4	5	6	7

2、ACM 编程 找座位

题目描述：在一个大型体育场内举办了一场大型活动，由于疫情防控的需要，要求每位观众的必须间隔至少一个空位才允许落座。现在给出一排观众座位分布图，座位中存在已落座的观众，请计算出，在不移动现有观众座位的情况下，最多还能坐下多少名观众。

输入描述：一个数组，用来标识某一排座位中，每个座位是否已经坐人。0 表示该座位没有坐人，1 表示该座位已经坐人。

输出描述：整数，在不移动现有观众座位的情况下，最多还能坐下多少名观众。

补充说明：1<=数组长度<=10000

```
示例

示例1
输入：10001
输出：1
说明：


示例2
输入：0101
输出：0
说明：
```

3、ACM 编程 信道分配

题目描述：算法工程师小明面对着这样一个问题，需要将通信用的信道分配给尽量多的用户：信道的条件及分配规则如下：

1) 所有信道都有属性："阶"。阶为 r 的信道的容量为 2^r 比特；
2) 所有用户需要传输的数据量都一样：D 比特；
3) 一个用户可以分配多个信道，但每个信道只能分配给一个用户；
4) 只有当分配给一个用户的所有信道的容量和>=D，用户才能传输数据；

给出一组信道资源，最多可以为多少用户传输数据？

输入描述：第一行，一个数字 R。R 为最大阶数。

0<=R<20

第二行，R+1 个数字，用空格隔开。

代表每种信道的数量 Ni。按照阶的值从小到大排列。

0<=i<=R, 0<=Ni<1000.

第三行，一个数字 D。

D 为单个用户需要传输的数据量。

0<D<1000000

输出描述：一个数字，代表最多可以供多少用户传输数据。

```
示例

示例1
输入：5
        10 5 0 1 3 2
        30
输出：4
说明：最大阶数为5.
        信道阶数：0      1   2   3   4   5
        信道容量：1      2   4   8   16  32
        信道个数：10     5   0   1   3   2
        单个用户需要传输的数据量为30

        可能存在很多分配方式，举例说明：
        分配方式1：
        1)    32*1 = 32
        2)    32*1 = 32
        3)    16*2 = 32
        4)    16*1 + 8*1 + 2*3 = 30
        剩下2*2 + 1*10=14不足以再分一个用户了。
```

```
分配方式2：
1)    16*1 + 8*1 + 2*3 = 30
2)    16*1 + 2*2 + 1*10 = 30
3)    32*1 = 32
4)    32*1 = 32
剩下16*1=16不足以再分一个用户了。

分配方式3：
1)    16*1 + 8*1 + 2*3 = 30
2)    16*1 + 2*2 + 1*10 = 30
3)    16*1 + 32*1 = 48
4)    32*1 = 32
恰好用完。
```

虽然每种分配方式剩下的容量不同，但服务的用户数量是一致的。因为这个问题中我们只关心服务的用户数，所以我们认为这些分配方式等效。

1、ACM 编程 求字符串中所有整数的最小和

题目描述：输入字符串 s，输出 s 中包含所有整数的最小和

说明

1. 字符串 s，只包含 a-z A-Z +- ；
2. 合法的整数包括
   1） 正整数 一个或者多个 0-9 组成，如 0 2 3 002 102
   2）负整数 负号 - 开头，数字部分由一个或者多个 0-9 组成，如 -0 -012 -23 -00023

输入描述：包含数字的字符串

输出描述：所有整数的最小和

```
示例

示例1
输入：bb1234aa
输出：10
说明：

示例2
输入：bb12-34aa
输出：-31
说明：1+2+（-34）= 31
```

2、ACM 编程 停车场车辆统计

题目描述：特定大小的停车场，数组 cars[]表示，其中 1 表示有车，0 表示没车。车辆大小不一，小车占一个车位（长度 1），货车占两个车位（长度 2），卡车占三个车位（长度 3），统计停车场最少可以停多少辆车，返回具体的数目。

输入描述：整型字符串数组 cars[]，其中 1 表示有车，0 表示没车，数组长度小于 1000。

输出描述：整型数字字符串，表示最少停车数目。

示例

示例1
输入：1,0,1
输出：2
说明：1个小车占第1个车位
第二个车位空
1个小车占第3个车位
最少有两辆车

示例2
输入：1,1,0,0,1,1,1,0,1
输出：3
说明：1个货车占第1、2个车位
第3、4个车位空
1个卡车占第5、6、7个车位
第8个车位空
1个小车占第9个车位
最少3辆车

3、ACM 编程 根据 IP 查找城市

题目描述：某业务需要根据终端的 IP 地址获取该终端归属的城市，可以根据公开的 IP 地址池信息查询归属城市。地址池格式如下城市名=起始 IP,结束 IP，起始和结束地址按照英文逗号分隔，多个地址段采用英文分号分隔。

比如 City1=1.1.1.1,1.1.1.2;City1=1.1.1.11,1.1.1.16;City2=3.3.3.3,4.4.4.4;City3=2.2.2.2,6.6.6.6。

一个城市可以有多个 IP 段，比如 City1 有 2 个 IP 段。城市间也可能存在包含关系，如 City3 的 IP 段范围包含 City2 的 IP 段范围。现在要根据输入的 IP 列表，返回最佳匹配的城市列表。

注：最佳匹配即包含待查询 IP 且长度最小的 IP 段，比如例子中 3.4.4.4 最佳匹配是 City2=3.3.3.3,4.4.4.4，5.5.5.5 的最佳匹配是 City3=2.2.2.2,6.6.6.6。

输入描述：输入共 2 行。

第一行为城市的 IP 段列表，多个 IP 段采用英文分号';'分隔，IP 段列表最大不超过 500000。城市名称只包含英文字母、数字和下划线，最多不超过 100000 个。IP 段包含关系可能有多层，但不超过 100 层。

第二行为查询的 IP 列表，多个 IP 采用英文逗号','分隔，最多不超过 10000 条。

输出描述：最佳匹配的城市名列表，采用英文逗号','分隔，城市列表长度应该跟查询的 IP 列表长度一致。

补充说明：1）无论是否查到匹配正常都要输出分隔符。举例：假如输入 IP 列表为 IPa,IPb 两个 IP 均未有匹配城市，此时输出为“,”，即只有一个逗号分隔符，两个城市均为空；

2）可以假定用例中的所有输入均合法，IP 地址均为合法的 ipv4 地址，满足（1~255）.（0~255）.（0~255）.（0~255）的格式，且可以假定用例中不会出现组播和广播地址；

示例

示例1
输入：City1=1.1.1.1,1.1.1.2;City1=1.1.1.11,1.1.1.16;City2=3.3.3.3,4.4.4.4;City3=2.2.2.2,6.6.6.6
1.1.1.15,3.3.3.5,2.2.2.3
输出：City1,City2,City3
说明：1）City1有2个IP段，City3的IP段包含City2的IP段；

2）1.1.1.15仅匹配City1=1.1.1.11,1.1.1.16，所以City1就是最佳匹配；2.2.2.3仅匹配City3=2.2.2.2,6.6.6.6，所以City3是最佳匹配；3.3.3.5同时匹配为City2=3.3.3.3,4.4.4.4和City3=2.2.2.2,6.6.6.6，但是City3=2.2.2.2,6.6.6.6的IP段范围更小，所以City3为最佳匹配；

1、ACM 编程 查找众数及中位数

题目描述：1.众数是指一组数据中出现次数量多的那个数，众数可以是多个
2.中位数是指把一组数据从小到大排列，最中间的那个数，如果这组数据的个数是奇数，那最中间那个就是中位数，如果这组数据的个数为偶数，那就把中间的
两个数之和除以 2，所得的结果就是中位数
3.查找整型数组中元素的众数并组成一个新的数组，求新数组的中位数
输入描述：输入一个一维整型数组，数组大小取值范围 0<N<1000，数组中每个元素取值范围 0<E<1000
输出描述：输出众数组成的新数组的中位数

```
示例1
输入：10 11 21 19 21 17 21 16 21 18 15
输出：21
说明：

示例2
输入：2 1 5 4 3 3 9 2 7 4 6 2 15 4 2 4
输出：3
说明：

示例3
输入：5 1 5 3 5 2 5 5 7 6 7 3 7 11 7 55 7 9 98 9 17 9 15 9 9 1 39
输出：7
说明：
```

2、ACM 编程 最长子字符串的长度（一）

题目描述：给你一个字符串 s，字符串 s 首尾相连成一个环形 ，请你在环中找出 'o' 字符出现了偶数次最长子字符串的长度。
输入描述：输入是一串小写字母组成的字符串
输出描述：输出是一个整数
补充说明：1 <= s.length <= 5 x 10^5
s 只包含小写英文字母。

```
示例

示例1
输入：alolobo
输出：6
说明：最长子字符串之一是 "alolob"，它包含'o' 2个。

示例2
输入：looxdolx
输出：7
说明：最长子字符串是 "oxdolxl"，由于是首尾连接在一起的，所以最后一个 'x' 和开头的 'l'是连接在一起的，此字符串包含 2 个'o' 。

示例3
输入：bcbcbc
输出：6
说明：这个示例中，字符串 "bcbcbc" 本身就是最长的，因为 'o' 都出现了 0 次。
```

3、ACM 编程 找城市

题目描述：一张地图上有 n 个城市，城市和城市之间有且只有一条道路相连：要么直接相连，要么通过其它城市中转相连（可中转一次或多次）。城市与城市之间的道路都
不会成环。
当切断通往某个城市 i 的所有道路后，地图上将分为多个连通的城市群，设该城市 i 的聚集度为 DP （Degree of Polymerization），DP = max(城市群 1 的城市
个数， 城市群 2 的城市个数, ... 城市群 m 的城市个数)。
请找出地图上 DP 值最小的城市（即找到城市 j，使得 DP = min(DP , DP ... DP))

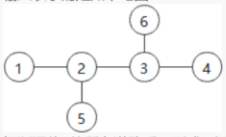
提示：如果有多个城市都满足条件，这些城市都要找出来（可能存在多个解）
提示：DP 的计算，可以理解为已知一棵树，删除某个节点后，生成的多个子树，求解多个子树节点数的问题。
输入描述：每个样例：第一行有一个整数 N，表示有 N 个节点。1<=N<=1000
接下来的 N-1 行每行有两个整数 x,y，表示城市 x 与城市 y 连接。1<=x, y<=N
输出描述：输出城市的编号。如果有多个，按照编号升序输出。

示例 展开

示例1
输入：5
1 2
2 3
3 4
4 5
输出：3
说明：输入表示的是如下地图：

对于城市3，切断通往3的所有道路后，形成2个城市群[（1,2），（4,5）]，其聚集度分别都是2。$DP_3 = 2$。 对于城市4，切断通往城市4的所有道路后， 形成2个城市群[(1,2,3), (5)]，$DP_4 = \max(3, 1) = 3$。依次类推，切断其它城市的所有道路后，得到的DP都会大于2，因为城市3就是满足条件的城市，输出是3。

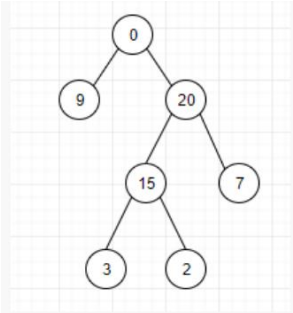
示例2
输入：6
1 2
2 3
2 5
3 4
3 6
输出：2 3
说明：输入表示的是如下地图：



切断通往2的所有道路后，形成3个城市群[（1），（5），（3,4,6）]，其聚集度分别都是1、1、3，因此$DP_2 = 3$。
切断通往3的所有道路后，形成3个城市群[（1，2,5），（4），（,6）]，其聚集度分别都是3、1、1，因此$DP_3 = 3$。
切断其它城市的所有道路后，得到的DP都会大于3，因为城市2、3就是满足条件的城市，升序排列输出是2 3

1、ACM 编程 传递悄悄话

题目描述：给定一个二叉树，每个节点上站着一个人，节点数字表示父节点到该节点传递悄悄话需要花费的时间。
初始时，根节点所在位置的人有一个悄悄话想要传递给其他人，求二叉树所有节点上的人都接收到悄悄话花费的时间。
输入描述：给定二叉树
0 9 20 -1 -1 15 7 -1 -1 -1 -1 3 2
注：-1 表示空节点
输出描述：返回所有节点都接收到悄悄话花费的时间 38



示例

示例1
输入：0 9 20 -1 -1 15 7 -1 -1 -1 -1 3 2
输出：38
说明：

2、ACM 编程 最大坐标值

题目描述：小明在玩一个游戏，游戏规则如下：
在游戏开始前，小明站在坐标轴原点处（坐标值为 0）。
给定一组指令和一个幸运数，每个指令都是一个整数，小明按照指定的要求前进或者后退指定的步数。前进代表朝坐标轴的正方向走，后退代表朝坐标轴的负方
向走。
幸运数为一个整数，如果某个指令正好和幸运数相等，则小明行进步数加 1。

例如：
幸运数为 3，指令为[2,3,0,-5]
指令为 2，表示前进 2 步；
指令为 3，正好和幸运数相等，前进 3+1=4 步；
指令为 0，表示原地不动，既不前进，也不后退。
指令为-5，表示后退 5 步；
请你计算小明在整个游戏过程中，小明所处的最大坐标值。
输入描述：第一行输入 1 个数字，代表指令的总个数 n（1<=n<=100）。
第二行输入 1 个数字，代表幸运数 m（-100<=m<=100）
第三行输入 n 个指令，每个指令值的取值范围为：-100<=指令值<=100。
输出描述：输出在整个游戏过程中，小明所处的最大坐标值。异常情况下输出：12345

```
示例

示例1
输入：2
      1
      -5 1
输出：0
说明：总共2个指令，幸运数为1，依照指令行进，依次如下：
      游戏开始前，站在坐标轴原点，此时坐标值为0；
      指令为-5，后退5步，此时坐标值为-5；
      指令为1，正好等于幸运数，前进1+1=2步，此时坐标值为-3；
      整个游戏过程中，小明所处的坐标值依次为[0,-5,-3]，最大坐标值为0。

示例2
输入：5
      -5
      -5 1 6 0 -7
输出：1
说明：总共5个指令，幸运数为-5，依照指令行进，依次如下：
      游戏开始前，站在坐标轴原点，此时坐标值为0；
      指令为-5，正好等于幸运数，后退5+1=6步，此时坐标值为-6；
      指令为1，前进1步，此时坐标值为-5；
      指令为6，前进6步，此时坐标值为1；
      指令为0，既不前进，也不后退，此时坐标值为1；
      指令为-7，后退7步，此时坐标值为-6；
      整个游戏过程中，小明所处的坐标值依次为[0,-6,-5,1,1,-6]，最大坐标值为1。
```

3、ACM 编程 推荐多样性

题目描述：推荐多样性需要从多个列表中选择元素，一次性要返回 N 屏数据（窗口数量），每屏展示 K 个元素（窗口大小），选择策略：

1. 各个列表元素需要做穿插处理，即先从第一个列表中为每屏选择一个元素，再从第二个列表中为每屏选择一个元素，依次类推
2. 每个列表的元素尽量均分为 N 份，如果不够 N 个，也要全部分配完，参考样例图：

（1）从第一个列表中选择 4 条 0 1 2 3，分别放到 4 个窗口中
（2）从第二个列表中选择 4 条 10 11 12 13，分别放到 4 个窗口中
（3）从第三个列表中选择 4 条 20 21 22 23，分别放到 4 个窗口中
（4）再从第一个列表中选择 4 条 4 5 6 7，分别放到 4 个窗口中
...
（5）再从第一个列表中选择，由于数量不足 4 条，取剩下的 2 条，放到窗口 1 和窗口 2
（6）再从第二个列表中选择，由于数量不足 4 条并且总的元素数达到窗口要求，取 18 19 放到窗口 3 和窗口 4

输入

N：4
K：7

recallA	0	1	2	3	4	5	6	7	8	9
recallB	10	11	12	13	14	15	16	17	18	19
recallC	20	21	22	23	24	25	26	27	28	29

处理

window1	window2	window3	window4
0	1	2	3
10	11	12	13
20	21	22	23
4	5	6	7
14	15	16	17
24	25	26	27
8	9	18	19

输出

0 10 20 4 14 24 8 1 11 21 5 15 25 9 2 12 22 6 16 26 18 3 13 23 7 17 27 19

window1 window2 window3 window4

输入描述：第一行输入为 N，表示需要输出的窗口数量，取值范围 [1, 10]
第二行输入为 K，表示每个窗口需要的元素数量，取值范围 [1, 100]
之后的行数不定（行数取值范围[1,10]），表示每个列表输出的元素列表。元素之间以空格分隔，已经过排序处理，每个列表输出的元素数量取值范围[1,100]
输出描述：输出元素列表，元素数量=窗口数量*窗口大小，元素之间以空格分隔，多个窗口合并为一个列表输出，参考样例：
先输出窗口 1 的元素列表，再输出窗口 2 的元素列表，再输出窗口 3 的元素列表，最后输出窗口 4 的元素列表
补充说明：1. 每个列表会保证元素数量满足窗口要求，不需要考虑元素不足情况
2. 每个列表的元素已去重，不需要考虑元素重复情况
3. 每个列表的元素列表均不为空，不需要考虑列表为空情况
4. 每个列表的元素列表已经过排序处理，输出结果要保证不改变同一个列表的元素顺序
5. 每个列表的元素数量可能是不同的

```
示例

示例1
输入：4
      7
      0 1 2 3 4 5 6 7 8 9
      10 11 12 13 14 15 16 17 18 19
      20 21 22 23 24 25 26 27 28 29
输出：0 10 20 4 14 24 8 1 11 21 5 15 25 9 2 12 22 6 16 26 18 3 13 23 7 17 27 19
说明：
```

1、ACM 编程 来自异国的客人

题目描述：有位客人来自异国，在该国使用 m 进制计数。该客人有个幸运数字 n(n<m)，每次购物时，其总是喜欢计算本次支付的花费(折算为异国的价格后)中存在多少幸运数字。问：当其购买一个在我国价值 k 的产品时，其中包含多少幸运数字？
输入描述：第一行输入为 k, n, m。
其中：
k 表示 该客人购买的物品价值（以十进制计算的价格）
n 表示 该客人的幸运数字
m 表示 该客人所在国度的采用的进制
输出描述：输出幸运数字的个数，行末无空格。
补充说明：当输入非法内容时，输出 0

示例

示例1
输入：10 2 4
输出：2
说明：10用4进制表示时为22，同时，异国客人的幸运数字是2，故而此处输出为2，表示有2个幸运数字。

示例2
输入：10 4 4
输出：0
说明：此时客人的幸运数字为4，但是由于该国最大为4进制，故而在该国的进制下不可能出现幸运数字，故而返回0

2、ACM 编程 小朋友来自多少小区

题目描述：幼儿园组织活动，老师布置了一个任务：每个小朋友去了解与自己同一个小区的小朋友还有几个。我们将这些数量汇总到数组 garden 中。

请根据这些小朋友给出的信息，计算班级小朋友至少来自几个小区？

输入描述：输入：garden[] = {2, 2, 3}

说明：

garden 数组长度最大为 999

每个小区的小朋友数量最多 1000 人，也就是 garden[i]的范围为[0,999]

输出描述：输出：7

示例

示例1
输入：2 2 3
输出：7
说明：解释：
第一个小朋友反馈有两个小朋友和自己同一小区，即此小区有3个小朋友
第二个小朋友反馈有两个小朋友和自己同一小区，即此小区有3个小朋友。
这两个小朋友，可能是同一小区的，且此小区的小朋友只有3个人。
第三个小区反馈还有3个小朋友与自己同一小区，则这些小朋友只能是另外一个小区的。这个小区有4个小朋友。

3、ACM 编程 加密算法

题目描述：有一种特殊的加密算法，明文为一段数字串，经过密码本查找转换，生成另一段密文数字串。规则如下：

1. 明文为一段数字串由 0~9 组成
2. 密码本为数字 0~9 组成的二维数组
3. 需要按明文串的数字顺序在密码本里找到同样的数字串，密码本里的数字串是由相邻的单元格数字组成，上下和左右是相邻的，注意：对角线不相邻，同一
个单元格的数字不能重复使用。
4. 每一位明文对应密文即为密码本中找到的单元格所在的行和列序号（序号从 0 开始）组成的两个数字。如明文 第 i 位 Data[i] 对应密码本单元格为 Book[x][y]，
则明文第 i 位对应的密文为 X Y，X 和 Y 之间用空格隔开。如果有多条密文，返回字符序最小的密文。如果密码本无法匹配，返回"error"。

请你设计这个加密程序。

示例 1：
密码本：
{0, 0, 2},
{1, 3, 4},
{6, 6, 4}
明文"3",密文" 1 1"
示例 2：
密码本：
{0, 0, 2},
{1, 3, 4},
{6, 6, 4}
明文"0 3",密文" 0 1 1 1"
示例 3：
密码本：
{0, 0, 2, 4}, {0, 0, 2, 4},
{1, 3, 4, 6}, {1, 3, 4, 6},
{3, 4, 1, 5}, {3, 4, 1, 5},
{6, 6, 6, 5}, {6, 6, 6, 5},
输入：明文"0 0 2 4", 密文" 0 0 0 1 0 2 0 3" 和" 0 0 0 1 0 2 1 2"，返回字符序小的" 0 0 0 1 0 2 0 3"
输入：明文"8 2 2 3", 密文" error"，密码本中无法匹配。

输入描述：第一行输入 1 个正整数 N，代表明文的长度(1 <= N <= 200)
第二行输入 N 个明文数字组织的序列 Data[i]，(整数，0 <= Data[i] <= 9)
第三行输入 1 个正整数 M，(1 <= M <= 200)
接下来输入一个 M * M 的矩阵代表密码本 Book[i][j]，(整数，0<=Book[i][j] <= 9)
输出描述：如明文 第 i 位 Data[i]对应密码本单元格为 Book[x][y]，则明文第 i 位对应的密文为 X Y，X 和 Y 之间用空格隔开。如果有多条密文，返回字符序最小的密文。如果密码本无法匹配，返回"error"。

示例

示例1
输入：2
0 3
3
0 0 2
1 3 4
6 6 4
输出：0 1 1 1
说明：

示例2
输入：2
0 5
3
0 0 2
1 3 4
6 6 4
输出：error
说明：找不到0 5的序列，返回error

1、ACM 编程 英文输入法
题目描述：主管期望你来实现英文输入法单词联想功能。需求如下：
依据用户输入的单词前缀，从已输入的英文语句中联想出用户想输入的单词，按字典序输出联想到的单词序列，如果联想不到，请输出用户输入的单词前缀。
注意：
1. 英文单词联想时，区分大小写

2. 缩略形式如”don't”，判定为两个单词，”don”和”t”
3. 输出的单词序列，不能有重复单词，且只能是英文单词，不能有标点符号
输入描述：输入为两行。
首行输入一段由英文单词 word 和标点符号组成的语句 str；
接下来一行为一个英文单词前缀 pre。
0 < word.length() <= 20
0 < str.length <= 10000
0 < pre <= 20
输出描述：输出符合要求的单词序列或单词前缀，存在多个时，单词之间以单个空格分割

示例 展开

示例1
输入：I love you
He
输出：He
说明：从用户已输入英文语句”I love you”中提炼出“I”、“love”、“you”三个单词，接下来用户输入“He”，从已输入信息中无法联想到任何符合要求的单词，因此输出用户输入的单词前缀。

示例2
输入：The furthest distance in the world, Is not between life and death, But when I stand in front of you, Yet you don't know that I love you.
f
输出：front furthest
说明：从用户已输入英文语句”The furthestdistance in the world, Is not between life and death, But when I stand in frontof you, Yet you dont know that I love you.”中提炼出的单词，符合“f”作为前缀的，有“furthest”和“front”，按字典序排序并在单词间添加空格后输出，结果为“frontfurthest”。

2、ACM 编程 考勤信息
公司用一个字符串来表示员工的出勤信息：
absent：缺勤
late：迟到
leaveearly：早退
present：正常上班
现需根据员工出勤信息，判断本次是否能获得出勤奖，能获得出勤奖的条件如下：
缺勤不超过一次；没有连续的迟到/早退；任意连续 7 次考勤，缺勤/迟到/早退不超过 3 次
输入描述：用户的考勤数据字符串，记录条数 >= 1；输入字符串长度<10000；不存在非法输入
如：
2
present
present absent present present leaveearly present absent
输出描述：根据考勤数据字符串，如果能得到考勤奖，输出"true"；否则输出"false"，对于输入示例的结果应为：
true false

示例

示例1
输入：2
present
present present
输出：true true
说明：

示例2
输入：2
present
present absent present present leaveearly present absent
输出：true false
说明：

3、ACM 编程 购物

题目描述：商店里有 N 件唯一性商品，每件商品有一个价格，第 i 件商品的价格是 。一个购买方案可以是从 N 件商品中选择任意件进行购买（至少一件），花费即价格之和。现在你需要求出所有购买方案中花费前 K 小的方案，输出这些方案的花费。

当两个方案选择的商品集合内至少有一件不同，视为不同方案，因此可能存在两个方案花费相同。

输入描述：输入数据含三行

第一行包含两个整数 N，K，整数之间通过空格隔开。分别表示商品的个数，以及需要求得的花费个数。

$1 \le N \le 10000, 1 \le K \le \min(2^N - 1, 100000)$

第二行包含 N 个整数，整数之间通过空格隔开。表示 N 件商品的价格。

$1 \le a_1 \le a_2 \le \ldots \le a_n \le 10000$

输出描述：按花费从小到大的顺序依次输出 K 行，一行一个整数。表示花费前 K 小的购买方案的花费。

示例

示例1
输入：5 6
1 1 2 3 3
输出：1
1
2
2
3
3
说明：花费前10小的方案：

集合	花费
1	1
2	1
3	2
1,2	2
4	3
5	3
1,3	3
2,3	3
1,2,3	4
1,4	4

示例2
输入：4 15
1 2 3 4
输出：1
2
3
3
4
4
5
5
6
6
7
7
8
9
10
说明：

示例3
输入：3 7
1 10 100
输出：1
10
11
100
101
110
111
说明：