

# 南京大学第 16 届算法设计大赛（公开组）

## 试题册

比赛时间：2019 年 4 月 21 日 13:00-17:00



主办单位：南京大学 ACM-ICPC 集训队

## 试题列表

- A Coffee and Chicken
- B Coffee or Chicken
- C ICPC
- D Match Maker
- E Mission Probable
- F Neural Network
- G NJU Emulator
- H Road Construction
- I Welcome Party
- J Wood Processing

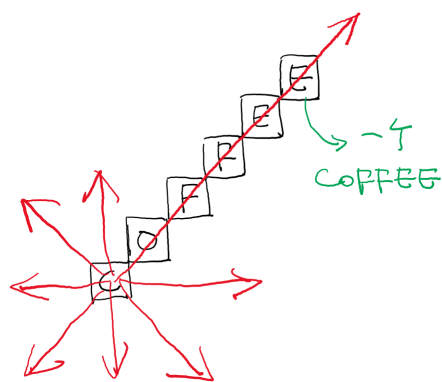
请不要在比赛开始前打开试题册！



# A. Coffee and Chicken

时间限制： 1 秒

据说，“Coffee Chicken”在组队的时候，ICPC 集训队刚好购置了福利咖啡机，所以起了这个名字。我们不禁感叹：ICPC 集训队的福利真实好！我们都要做 Coffee Chicken 的粉丝！这一次，咖啡鸡把自己藏在了一个  $n \times m$  的字符矩阵里 (字符矩阵由大写字母构成)，并且希望我们找出隐藏的“咖啡鸡”——从矩阵的某个格子出发，向如下图所示的 8 个方向，如果能读出 COFFEE 或 CHICKEN，则认为找到了一个单词。



你的任务是统计单词 COFFEE 和 CHICKEN 出现的次数。

## 输入格式

输入数据第一行包含两个整数  $n, m$  ( $1 \leq n, m \leq 100$ )。接下来  $n$  行，每行  $m$  个用空格分隔开的大写字母 A-Z，描述一个字符矩阵。

## 输出格式

- 输出两行：
1. 第一行格式：COFFEE x  $x$
  2. 第二行格式：CHICKEN x  $y$
- 其中  $x, y$  分别是字符矩阵中 COFFEE 和 CHICKEN 出现的次数。

### 样例输入 1

```
7 6
C O F F E E
H O B C D E
I A F G H I
C J K F L M
K N O P E Q
E R S T U E
N V W X Y Z
```

### 样例输出 1

```
COFFEE x 2
CHICKEN x 1
```

样例输入 2

2 3  
Z Z Z  
Z Z Z

样例输出 2

COFFEE x 0  
CHICKEN x 0

# B. Coffee or Chicken

时间限制： 2 秒

JYY 创造了一种“咖啡鸡字符串”，类似于“斐波那契例汤”——第  $n$  天的汤是由第  $n - 2$  天和第  $n - 1$  天的汤混合而成的：

- $S(1) = \text{"COFFEE"};$
- $S(2) = \text{"CHICKEN"};$
- $S(n) = S(n - 2) :: S(n - 1)$ ，其中“ $::$ ”是字符串的拼接运算符。

随着  $n$  的增长， $S(n)$  的长度很快就失控了，JYY 的小霸王学习机没有足够的内存存储  $S(n)$ 。JYY 希望你帮助已经被降智的他求出  $S(n)$  从第  $k$  个位置开始的连续 10 个字符。

## 输入格式

输入数据第一行包含两个整数  $n, k$  ( $1 \leq n \leq 30, 1 \leq k \leq |S(n)|$ )，其中  $|S|$  表示字符串  $S$  的长度。

## 输出格式

输出一行，代表  $S(n)$  从第  $k$  个位置起的 10 个字符。如果第  $k$  个位置起字符数不足 10 个，则输出到  $S(n)$  末尾的所有字符。

样例输入 1

|     |            |
|-----|------------|
| 3 4 | FEECHICKEN |
|-----|------------|

样例输出 1

样例输入 2

|     |   |
|-----|---|
| 2 7 | N |
|-----|---|

样例输出 2

本页无正文

## C. ICPC

时间限制：3 秒

ICPC 国际大学生程序设计竞赛是全球最具影响力的大学生程序设计竞赛。比赛以团队的形式进行，需要在 5 个小时内解决至多 13 道题目。程序完成之后提交裁判运行，运行的结果会判定为正确或错误并及时通知参赛队。有趣的是，每队在正确解答一题后，组织者将在其位置上升起一只代表该题颜色的气球，每道题目第一支解决掉它的队还会额外获得一个“FIRST PROBLEM SOLVED”的气球，也就是所谓的“一血气球”。



Figure 1: ICPC 2019 World Finals

ICPC 最后的获胜者为正确解答题目最多且总用时最少的队伍。每道试题用时将从竞赛开始到试题解答被判定为正确为止，其间每一次提交运行结果被判错误的话将被加罚 20 分钟时间，未正确解答的试题不计时。

JYY 在参加完一场 ICPC 竞赛后，观察了最终的榜单，他想知道自己在比赛中的策略是否可以改进。确切的说，JYY 想知道，如果在比赛中他改变了做题的顺序，最多能够拿到多少个“一血气球”，以及拿到的最好名次是多少。

在本题中，为了简化问题，我们不考虑罚时，并且假定：

- JYY 所在队伍完成第一道题所需花费的时间即为通过该题时刻，此后完成每道题所需花费的时间为通过该题的时刻减去通过上一题的时刻；
- JYY 改变做题顺序不会影响解答每道题所花费的时间；
- JYY 所在队伍在比赛中没有做出来的题，改变做题顺序后仍然做不出来；
- 若存在多支队伍在同一个最小时刻解决某道问题，则这些队伍都能获得“一血气球”。

最终所有队伍先按照正确解答的题目数量从大到小排名，正确解题数相同的队伍，按照总用时从小到大排名。总用时是指通过每道题的时刻之和。若存在多支队伍过题数，罚时均相同，则这些队伍获得相同的、较优的排名。例如，第一支队通过了 7 题，总用时 100 分钟，第二支队通过了 7 题，总用时 100 分钟，第三支队通过了 6 题，总用时 70 分钟，则在这三支队中，第一、第二支队并列第一名，第三支队是第三名。

### 输入格式

输入的第一行包含两个整数  $n, m$  ( $2 \leq n \leq 1000, 1 \leq m \leq 10$ )，表示参加比赛的团队数量和这场比赛中题目的数量。

接下来  $n$  行，每行包含  $m$  个整数，表示一支队伍的通过每道题的时刻（单位：分钟）。具体来说，第  $i$  行的第  $j$  个整数  $a_{ij}$  ( $0 < a_{ij} < 300$ ) 表示第  $i$  只队伍通过第  $j$  道题的时间。如果第  $i$  支队没有通过第  $j$  题， $a_{ij}$  将会以 -1 代替。保证对于每个队，在同一分钟内最多通过一题。

JYY 所在的队伍是第一支队。

输出格式

输出一行包括两个整数，即最多能够拿到多少个“一血气球”和最高能取得的名次。为实现这两个目标，你可以分别采取两种不同的做题顺序。

样例输入 1

```
3 3
80 90 100
10 20 100
- 5 15
```

样例输出 1

```
1 1
```

样例输入 2

```
5 4
30 50 - 115
80 15 75 -
30 40 - 120
70 80 180 105
- - 1 -
```

样例输出 2

```
2 3
```



# D. Match Maker

时间限制： 3 秒

NJU ACM/ICPC 集训队的机房内堆放了一些碎木片。这些碎木片都是加工木板时剩下的边角料。由于 zyb 很心疼他的钱，zyb 决定将这些碎木片加工成火柴。为了实现这一目标，zyb 首先将这些木片切割成长短不一的细木棍。

现在问题来了。zyb 如何进一步将这些木棍加工成火柴呢？zyb 希望加工的过程满足以下条件：

- 首先，选出至少一半（大于等于  $\lceil \frac{n}{2} \rceil$  根，其中  $\lceil x \rceil$  表示大于等于  $x$  的最小整数）木棍，并将其他未选择的木棍丢弃；
- zyb 需要将这些木棍进一步切割成等长的火柴棍（当然，如果木棍的长度等于火柴棍的长度，则不需要切割这根木棍），切割的过程不允许产生废弃的短木棍，也不允许将木棍拼接起来；
- 为防止火柴燃烧时间过短，火柴的长度必须大于 1，但不必是整数。

你需要判断是否存在满足这些条件的火柴制作方案。

例如，如果有三根长度分别为 3,5,6 的木棍，zyb 可以选出其中长度为 3 和 6 的木棍，将它们切割成长度为 1.5 的火柴棍共 6 根。

## 输入格式

第一行包含一个整数  $n$  ( $1 \leq n \leq 2 \times 10^5$ )，表示细木棍的数量。  
第二行包含  $n$  个整数  $x_1, x_2, \dots, x_n$  ( $2 \leq x_i \leq 10^{12}$ )，表示每根细木棍的长度。

## 输出格式

如果存在满足题目中所述条件的火柴制作方案，输出 YES；否则输出 NO。

| 样例输入 1            | 样例输出 1 |
|-------------------|--------|
| 3<br>3 5 6        | YES    |
| 样例输入 2            | 样例输出 2 |
| 5<br>2 4 6 3 5    | YES    |
| 样例输入 3            | 样例输出 3 |
| 6<br>2 4 5 7 9 11 | NO     |

样例输入 4

样例输出 4

|              |     |
|--------------|-----|
| 4<br>2 2 3 3 | YES |
|--------------|-----|

## E. Mission Probable

时间限制：1 秒

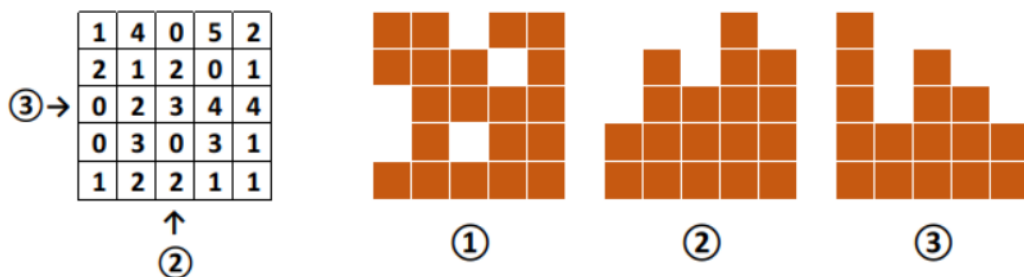
国际犯罪预防公司（International Crime Prevention Company，简称 ICPC 公司）最近为超级计算机制造公司（Advanced Computer Manufacturers，简称 ACM 公司）设计了一套产品仓库防盗窃系统。

ACM 公司的产品仓库可以划分为  $n$  行  $m$  列共  $n \times m$  个格子，而 ACM 公司的产品——装在立方体纸箱内的成品计算机，则整齐地垒放在这些格子内。

ICPC 公司设计的防盗窃系统很简单：在仓库的上方、前方、左侧各安装一个朝下、朝后、朝右的摄像头，摄像头的编号分别为 1,2,3。当摄像头拍摄的画面和仓库管理系统内的记录不一致时，发出警报。

现在，ICPC 公司已经完成了硬件设备的制作。作为 ICPC 的职工，你需要编写这套系统的配套软件：输入仓库管理系统内记录的产品堆放情况，输出三个摄像头应当拍摄到的画面。

下图显示了样例数据 1 的仓库堆放情况和三个摄像头应当拍摄到的画面。



### 输入格式

第一行包含两个整数  $n, m$  ( $1 \leq n, m \leq 100$ ), 表示仓库划分成的行数和列数。

接下来  $n$  行，每行  $m$  个非负整数，表示该位置堆放的产品个数。每个位置堆放的产品不超过 100。保证仓库中至少有一个格子内有产品。

### 输出格式

请依次输出上方、前方、左侧摄像头应当拍摄到的画面，用 # 表示箱子，用 . 表示空白。画面之间请用空行分隔。

对于前方和左侧摄像头的画面，请保证输出的行数最小（即没有多余的全为空的行）。

样例输入 1

```
5 5
1 4 0 5 2
2 1 2 0 1
0 2 3 4 4
0 3 0 3 1
1 2 2 1 1
```

样例输出 1

```
##.##
###.#
.####
.#.##
#####

...#.
.#.##
.####
#####
#####

#....
#.#..
#.#.#
#####
#####
```

样例输入 2

```
2 3
0 0 0
0 1 0
```

样例输出 2

```
...
.#.

.#.

.#
```

## F. Neural Network

时间限制：8 秒

终于，人类的神经网络模型被神秘博士彻底破译。神秘博士发现，人类的本质只是一个简单的复读机，由三种类型的神经元（节点）组成：

- 输入节点 (I)
- 输出节点 (O)
- 隐藏节点 (H)

其中，节点和节点之间可能存在神经连接，每个神经连接都有其强度  $w$ ，神经连接构成一个有向图。

JYY 受到 JitBrains 公司的赞助 (JitBrains 同样赞助了神秘博士的研究)，得到了一台最新的脑电波干扰器。干扰器有一个旋钮，可以设置发射功率为  $k$ 。一旦干扰器启动，对于强度为  $w$  的神经连接  $u \rightarrow v$ ，如果它满足  $w \leq k$ ，则该神经连接将被切断。切断神经网络连接的这种骚操作可以有效降低一定范围内人类的智商。

NJUPC 终于要到了。为了防止咖啡鸡屠榜，JYY 打算悄悄把脑电波干扰器安装在咖啡鸡的电脑附近，以便在比赛开始后降低他们的智商。但因为干扰器有一定的作用范围，所以过大的发射功率可能会把无辜的同学降智从而暴露 JYY 的行为，因此善良的 JYY 希望你帮助他设置最佳的脑电波干扰器的发射功率  $k$ 。具体来说，你需要

1. 把咖啡鸡的智商降低到零，即删除图中强度  $w \leq k$  的神经连接后，对于任意一个输入节点和任意一个输出节点，它们之间都不存在神经连接直接或间接连通的路径；
2. 对其他选手的影响最小，即你需要找到满足条件 1 的  $k$  的最小值。

### 输入格式

输入数据描述了咖啡鸡的神经网络模型。

输入数据第一行  $1 \leq m \leq 10^6$  表示神经连接的数量。接下来  $m$  行，每行由空格分开的几部分组成，形如：  $u \rightarrow v (w)$ ，其中：

- $u$  和  $v$  是一个节点的描述，用  $I[x]$ ,  $O[x]$ ,  $H[x]$  分别表示输入、输出和隐藏节点 ( $1 \leq x \leq 10^5$ )；不同类型的神经元可能就有相同的编号；
- 整数  $w$  是神经连接的强度 ( $0 < w \leq 10^6$ )；
- 保证  $u$  不会是输入节点， $v$  不会是输出节点。

注意，同一对神经元之间可能有多个（方向相同或相反）的神经连接，但一个神经不会连接两个相同的神经元。输入数据保证咖啡鸡是有智商的：至少一对输入/输出节点是连通的。

### 输出格式

输出一个整数，表示答案。

#### 样例输入 1

```
2
I[10] -> O[10] (1)
I[10] -> O[10] (2)
```

#### 样例输出 1

```
2
```

样例输入 2

样例输出 2

|   |     |
|---|-----|
| 3<br>I[1] -> H[1] (200)<br>I[2] -> H[1] (300)<br>H[1] -> O[1] (100) | 100 |
|---|-----|

# G. NJU Emulator

时间限制：1 秒

NJU Emulator（简称 NEMU）是 JYY 博士最新研发的 s86 架构模拟器。s86 是一种栈式计算机体系结构，其机器指令只能对栈顶元素进行操作。s86 的计算模型包括一个栈和一段有限长的程序。栈中的每个元素均为 64 位无符号整数类型；程序则由下表中的指令构成，且必须以 end 指令结束。s86 机器运行时，首先会将栈初始化为空，随后将会依次执行程序中的每条指令，直到执行完最后一条 end 指令后，机器会输出栈顶元素并停机。

| 助记符   | 功能              | 限制                    |
|-------|-----------------|-----------------------|
| pl    | 将常数 1 压入栈中      | 无                     |
| dup   | 将栈顶元素复制一份后压入栈中  | 当前栈必须非空               |
| pop   | 弹出栈顶元素          | 当前栈必须非空               |
| swap  | 交换栈顶两个元素        | 当前栈的大小至少是 2           |
| add x | 栈顶元素加上栈中第 x 个元素 | x 为非负整数，当前栈的大小必须大于 x  |
| sub x | 栈顶元素减去栈中第 x 个元素 | x 为非负整数，当前栈的大小必须大于 x  |
| mul x | 栈顶元素乘以栈中第 x 个元素 | x 为非负整数，当前栈的大小必须大于 x  |
| end   | 输出栈顶元素并停止程序执行   | 当前栈必须非空，且必须是程序的最后一条指令 |

Table 1: s86 架构的指令表

其中，栈中第 x 个元素表示栈顶向下第 x 个元素。栈顶元素本身是第 0 个元素。

需要注意的是，s86 中所有算术运算指令（add, sub, mul）的结果都需要对  $2^{64}$  取模，即当算术运算的结果为  $X$  时，s86 指令的运算结果为  $X'$  ( $0 \leq X' < 2^{64}$ )，并且  $X - X'$  是  $2^{64}$  的倍数。可以证明，对于任意整数  $X$ ，这样的  $X'$  都是唯一存在的。

现在，JYY 已经完成了 NEMU 的开发。为了测试 NEMU 的正确性，JYY 想请你编写一段 s86 程序，使得机器执行结束后，能够输出给定整数  $N$ 。

以样例输出 2 为例，s86 的运行过程如下所示：

| 指令    | 栈          |
|-------|------------|
| pl    | [ 1 )      |
| dup   | [ 1 1 )    |
| add 1 | [ 1 2 )    |
| add 1 | [ 1 3 )    |
| dup   | [ 1 3 3 )  |
| add 2 | [ 1 3 4 )  |
| add 2 | [ 1 3 5 )  |
| mul 1 | [ 1 3 15 ) |
| swap  | [ 1 15 3 ) |
| pop   | [ 1 15 )   |
| end   | (输出 15)    |

Table 2: s86 程序运行示例

## 输入格式

输入一个整数  $N$  ( $0 \leq N < 2^{64}$ )。

## 输出格式

输出一段程序，程序执行结束后，机器能够输出  $N$ 。你的程序最多只能包含 **200** 条指令。注意，如果你的程序在执行过程中发生违反指令表中相关限制的行为，你的答案将会被判定为错误。

如果存在多个满足要求的程序，输出任意一个即可。保证对于所有输入数据都存在至少一个满足要求的程序。

样例输入 1

2

样例输出 1

```
p1
add 0
end
```

样例输入 2

15

样例输出 2

```
p1
dup
add 1
add 1
dup
add 2
add 2
mul 1
swap
pop
end
```

样例输入 3

0

样例输出 3

```
p1
dup
sub 1
end
```

样例输入 4

18446744073709551615

样例输出 4

```
p1
dup
add 1
swap
sub 1
end
```

## 提示

在 C++ 中，可以使用 `std::uint64_t`（在 `cstdint` 头文件内）类型实现模  $2^{64}$  下的算术运算。



## H. Road Construction

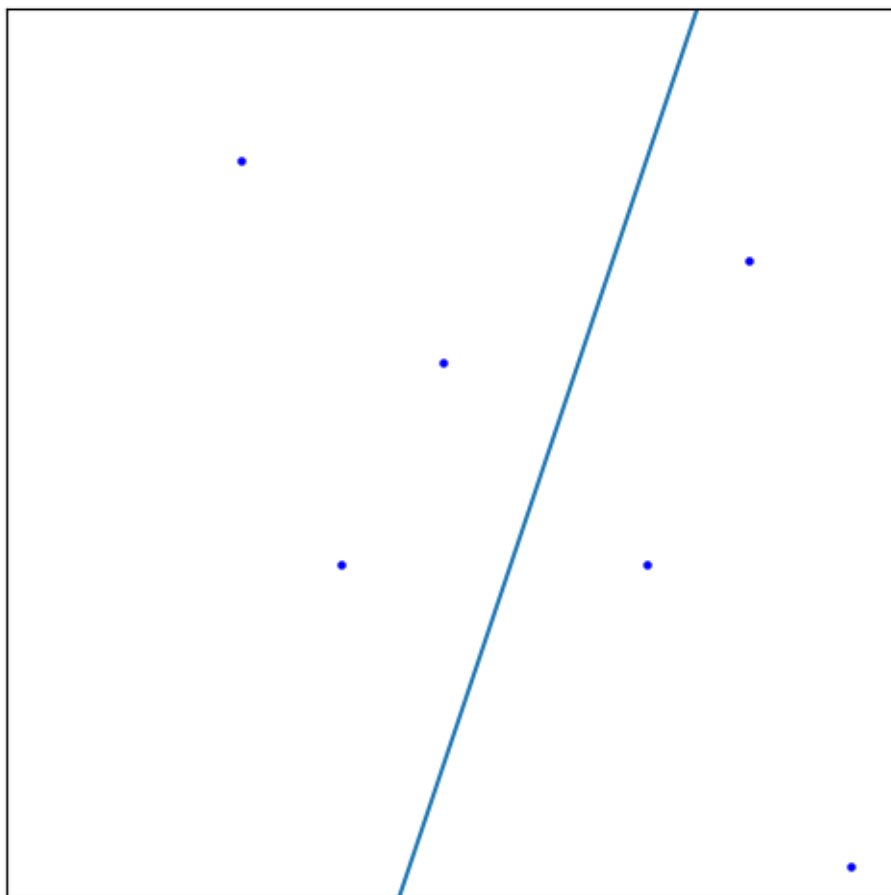
时间限制：5 秒

X 村内有若干住户。为促进经济发展，方便居民出行，X 村准备新修建一条穿城而过的直线形公路。修建的公路必须满足以下条件：

- 公路不能经过任何住户；
- 为公平起见，公路两侧的住户数量必须相等；
- 在满足上述条件的前提下，需要尽量减少交通噪声对居民生活的影响；即，所有住户到公路距离的最小值必须尽可能大。

请你帮助 X 村规划新建的道路。你只需要计算所有住户到公路距离最小值最大能达到多少。

下图是测试数据 1 的最优道路修建方案。



输入格式

第一行包含一个偶数  $n$  ( $2 \leq n \leq 300$ ), 表示住户的个数。  
接下来  $n$  行, 每行两个整数  $x,y$  ( $|x|, |y| \leq 10^6$ ), 表示一个住户的坐标。  
保证没有坐标相同的两个住户。

输出格式

输出一个实数, 表示答案。你的答案的绝对或相对误差不应超过  $10^{-6}$ 。

| 样例输入 1  | 样例输出 1         |
|---|----------------|
| 6<br>0 0<br>3 0<br>1 2<br>4 3<br>-1 4<br>5 -3 | 1.264911064067 |

| 样例输入 2          | 样例输出 2         |
|-----------------|----------------|
| 2<br>1 3<br>2 5 | 1.118033988750 |

# I. Welcome Party

时间限制：5 秒

计算机科学与技术系一年一度的迎新晚会马上就要开始了！今年，报名参加迎新晚会的同学格外的多，而每位报名的同学又可以选择表演一个歌舞类节目，或是表演一个语言类节目。这可愁坏了迎新晚会的总导演：该怎样安排节目，既可以使得每位报名的同学都能得到上台表演的机会，又能使观众尽量满意？



Figure 2: 计算机科学与技术系 2018 迎新晚会

为了解决这一难题，总导演提出了以下模型：每位报名同学都有两个属性  $x, y$ ，分别是他所表演的歌舞类节目和语言类节目的精彩程度。观众对歌舞类节目的满意度为所有歌舞类节目精彩程度的最大值；同样，观众对语言类节目的满意度为所有语言类节目精彩程度的最大值。而观众对歌舞类节目、语言类节目的满意度之差的绝对值越小，观众对整场迎新晚会的满意度越高。也就是说，这两类节目的满意度越接近，观众对迎新晚会就越满意。

数学不好的总导演并不会求最优的节目安排方案，使得观众对整场迎新晚会最为满意。具体来说，他希望你找到一种节目安排方案，满足以下条件：

- 每位报名同学要么表演一个歌舞类节目，要么表演一个语言类节目；
- 至少有一位同学表演歌舞类节目，也至少有一位同学表演语言类节目；
- 在满足上述前提的条件下，观众对两类节目满意度之差的绝对值要尽可能小。

你需要求出观众对两类节目满意度之差的绝对值的最小值。

## 输入格式

第一行包含一个整数  $n$  ( $2 \leq n \leq 10^4$ )，表示报名参加迎新晚会的同学数量。

接下来  $n$  行，每行包含两个数字  $x, y$  ( $0 \leq x, y \leq 10^9$ )，表示一位同学所表演的歌舞类节目和语言类节目的精彩程度。

输出格式

输出一个整数，表示答案。

样例输入 1

5  
27 46  
89 13  
55 8  
71 86  
22 35

样例输出 1

3

样例输入 2

3  
3 5  
4 7  
6 2

样例输出 2

1

# J. Wood Processing

时间限制：3 秒

出于某些奇怪的原因（比如说，练功），NJU ACM/ICPC 集训队购置了一些木板。不过由于负责人 **zyb** 的疏忽，购置的木板数量比预期的要多，所以 **zyb** 决定通过裁剪一部分木板对木板加工，使得木板数量符合要求。由于 **zyb** 很心疼他的钱，所以他希望被裁剪掉部分的面积尽量小。

**zyb** 有  $n$  块木板，其中第  $i$  块木板的高度是  $h_i$ ，宽度是  $w_i$ ，高度和宽度都是正整数。木板不能旋转。**zyb** 可以横向裁剪每块木板，即将其高度  $h_i$  变为  $h'_i$  ( $0 < h'_i \leq h_i$ )， $h'_i$  也必须是正整数。随后，**zyb** 可以将高度相同的一块或多块木板拼接成一块大木板。

**zyb** 希望知道在拼成  $k$  块大木板的前提下，被裁剪掉的区域的总面积的最小值。**zyb** 并不想做这件工作，但是算不出来的话会受到 **Acesrc** 的严厉批评。他希望你能帮助他计算出这个答案，如果你能成功通过这道题的话，**zyb** 会决定请你喝一杯奶茶。

例如，对于测试样例 1，可以将宽度为 2、高度为 5 的木板裁剪为高度为 3，再和宽度为 3、高度为 3 的木板拼成一块大木板。宽度为 4、高度为 7 的木板自身构成一块大木板。

## 输入格式

第一行包含两个整数  $n, k$  ( $k \leq n, 1 \leq n \leq 700, 1 \leq k \leq 300$ )，表示木板的个数和需要拼成大木板的个数。

接下来  $n$  行，每行包含两个数字  $w_i, h_i$  ( $1 \leq w_i, h_i \leq 10^7$ )，表示一块木板的宽度和高度。

## 输出格式

输出一个整数，表示答案。

样例输入 1

|                          |   |
|--------------------------|---|
| 3 2<br>3 3<br>4 7<br>2 5 | 4 |
|--------------------------|---|

样例输出 1

样例输入 2

|   |   |
|---|---|
| 6 3<br>1 1<br>1 2<br>1 3<br>1 4<br>1 5<br>1 6 | 3 |
|---|---|

样例输出 2

本页无正文