



中国计算机学会  
China Computer Federation

# 从零开始学动态规划算法



# 数字三角形

Time Limit: 1000MS Memory Limit: 128MB

- 给定一个数字三角形
  - 从第一行的数开始, 每次可以向左下或右下走一格, 直到最后一行
  - 要求一路上所有数的和最大





中国计算机学会

China Computer Federation

## Input

第一行一个整数 $N$  ( $N \leq 1000$ ), 表示三角形总共有几行  
第二至第 $N+1$ 行, 给出这个数字三角形

## Output

一个整数, 表示一路上所有数的最大和, 结果不会超过 $2^{63} - 1$

## Sample Input

```
4
1
3 2
4 10 1
4 3 2 20
```

## Sample Output

```
24
```



中国计算机学会

China Computer Federation

- $a[i][j]$ 表示第*i*行第*j*列上的数值
- 从上往下走
  - $f[i][j]$ 表示从第一行数到第*i*行第*j*列上数的最大路径之和
  - $(i, j)$ 位置只能从 $(i-1, j)$ 或 $(i-1, j-1)$ 位置到达
  - 故 $f[i][j] = a[i][j] + \max\{f[i-1][j], f[i-1][j-1]\}$
  - $\text{ans} = \max\{f[n][i] \mid 1 \leq i \leq n\}$
- 从下往上走
  - $f[i][j]$ 表示第*i*行第*j*列上数到达底端的最大路径之和
  - $(i, j)$ 位置只能去往 $(i+1, j)$ 或 $(i+1, j+1)$ 位置
  - 故 $f[i][j] = a[i][j] + \max\{f[i+1][j], f[i+1][j+1]\}$
  - $\text{ans} = f[1][1]$



# 最长上升子序列

Time Limit: 1000MS Memory Limit: 128MB

- 给定一个长度为N的整数序列A
- 找到一组最长的整数序列x满足:
  - $1 \leq x_1 < x_2 < \dots < x_k \leq N$
  - $A[x_1] < A[x_2] < \dots < A[x_k]$

1	6	2	5	4	7
---	---	---	---	---	---

- 即寻找A的一个最长子序列, 满足:
  - 该子序列中每个元素**递增**



中国计算机学会  
China Computer Federation

## Input

第一行 一个整数 $N \leq 1000$  表示长度

第二行  $N$ 个数  $A[i]$ 表示序列里面的数 每个数小于 $\text{maxlongint}$

## Output

一行 表示最长递增子序列的长度

## Sample Input

6

1 6 2 5 4 7

## Sample Output

4



中国计算机学会

China Computer Federation

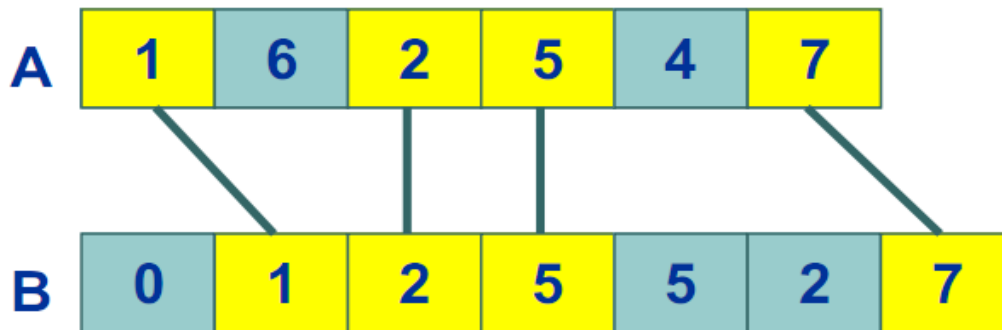
- $a[i]$ 表示第 $i$ 个数
- $f[i]$ 表示以 $a[i]$ 这个数为结尾的最长上升子序列的长度
- $f[i] = \max\{f[j] + 1 \mid a[i] > a[j] \text{ \&\& } 1 \leq j < i\}$
- $\text{ans} = \max\{f[i] \mid 1 \leq i \leq n\}$



# 最长公共子序列

Time Limit: 1000MS Memory Limit: 128MB

- 给定两个整数序列A和B
- 寻找一个最长的整数序列X, 满足:
  - X是A的子序列
  - X是B的子序列







中国计算机学会  
China Computer Federation

## Input

第一行：序列A的长度  $\leq 1000$

第二行：给出序列A

第三行：序列B的长度  $\leq 1000$

第四行：给出序列B

## Output

只有一行：表示最长的公共子序列的长度

## Sample Input

6

1 6 2 5 4 7

7

0 1 2 5 5 2 7

## Sample Output

4



中国计算机学会

China Computer Federation

- $a[i]$ 表示序列A上第 $i$ 个数
- $b[i]$ 表示序列B上第 $i$ 个数
- $f[i][j]$ 表示序列A前 $i$ 个数、序列B前 $j$ 个数的最长公共子序列

$$f[i][j] = \begin{cases} 0 & i == 0 || j == 0 \\ f[i-1][j-1] + 1 & a[i] == b[j] \\ \max(f[i][j-1], f[i-1][j]) & a[i] \neq b[j] \end{cases}$$



中国计算机学会  
China Computer Federation

# 装箱问题

Time Limit: 1000MS Memory Limit: 128MB

## Description

有一个箱子容量为 $V$  (正整数,  $0 \leq V \leq 20000$ ), 同时有 $n$ 个物品 ( $0 < n \leq 30$ ), 每个物品有一个体积 (正整数)。要求从 $n$ 个物品中, 任取若干个装入箱内, 使箱子的剩余空间为最小。

## Input

输入有若干行。第一行: 一个整数, 表示箱子容量 $V$ ; 第二行: 一个整数, 表示物品个数 $n$ ; 接下来 $n$ 行, 分别表示这 $n$ 个物品的各自体积。

## Output

输出只有一行数据, 该行只有一个数, 表示最小的箱子剩余空间。



中国计算机学会  
China Computer Federation

Sample Input

24

6

8

3

12

7

9

7

Sample Output

0



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个物品的体积
- (1)
  - $f[i][j]$ 表示前 $i$ 个物品空间为 $j$ 的箱子能装入最大体积
  - $$f[i][j] = \begin{cases} f[i-1][j] & a[i] > j \\ f[i-1][j-a[i]] + a[i] & a[i] \leq j \end{cases}$$
  - $ans = v - f[n][v]$
- (2)降维
  - $f[i]$ 表示空间为 $i$ 的箱子能否装满
  - $f[i] = f[i] \mid f[i-a[j]]$
  - $ans = v - \max\{ f[i]==true \mid 1 \leq i \leq n \}$



中国计算机学会  
China Computer Federation

# 砝码称重

Time Limit: 1000MS Memory Limit: 128MB

## Description

设有 $n$ 种砝码，第 $k$ 种砝码有 $C_k$ 个，每个重量均为 $W_k$ ，求：用这些砝码能秤出的不同重量的个数，但不包括一个砝码也不用的情况。

## Input

输入有若干行。第一行：一个整数，表示箱子容量 $V$ ；第二行：一个整数，表示物品个数 $n$ ；接下来 $n$ 行，分别表示这 $n$ 个物品的各自体积。

## Output

输出只有一行数据，该行只有一个数，表示最小的箱子剩余空间。



中国计算机学会

China Computer Federation

## Sample Input

24

6

8

3

12

7

9

7

## Sample Output

0

## Constraint

对于100%的数据,砵码的种类 $n$ 满足: $1 \leq n \leq 100$ ;

对于30%的数据,砵码的总数量 $C$ 满足: $1 \leq C \leq 20$ ;

对于100%的数据,砵码的总数量 $C$ 满足: $1 \leq C \leq 100$ ;

对于所有的数据,砵码的总重量 $W$ 满足: $1 \leq W \leq 400000$ ;



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个砝码的重量
- (1)
  - $f[i][j]$ 表示前 $i$ 个砝码能否装到重量 $j$
  - $$f[i][j] = \begin{cases} f[i-1][j] & a[i] > j \\ f[i-1][j-a[i]] + a[i] & a[i] \leq j \end{cases}$$
- (2)降维
  - $f[i]=\text{true}$ 表示能用砝码装到重量 $i$ , 否则不能
  - $f[i] = f[i] \mid f[i-a[j]]$





中国计算机学会  
China Computer Federation

# 石子归并

Time Limit: 1000MS Memory Limit: 128MB

## Description

有一堆石头质量分别为 $W_1, W_2, \dots, W_n$  ( $W_i \leq 10000$ ), 将石头合并为两堆, 使两堆质量的差最小。

## Input

输入第一行只有一个整数 $n$  ( $1 \leq n \leq 50$ ), 表示有 $n$ 堆石子。  
接下去的 $n$ 行, 为每堆石子质量。

## Output

输出只有一行数据, 该行只有一个数, 表示最小的质量差。



中国计算机学会  
China Computer Federation

Sample Input

5

5

8

13

27

14

Sample Output

3



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个石子的重量
- (1)
  - $f[i][j]$ 表示前 $i$ 颗石子能否合并成重量 $j$  ( $1 \leq j \leq (\sum_{i=1}^n a[i]) \setminus 2$ )
  - $f[i][j] = \begin{cases} f[i-1][j] & a[i] > j \\ f[i-1][j-a[i]] & a[i] \leq j \end{cases}$
  - $ans = \sum_{i=1}^n a[i] - 2 * \max\{j \mid f[n][j] == true\}$
- (2)降维
  - $f[i]=true$ 表示用石子合并为重量 $i$ , 否则不能
  - $f[i] = f[i] \mid f[i-a[j]]$



## Description

有四个人，每人身上的衣服分别有 $s_1, s_2, s_3$ 和 $s_4$ 处破损，而且每处破损程度不同，破损程度用需修好它用的时间表示( $A_1 \dots A_{s_1}, B_1 \dots B_{s_2}, C_1 \dots C_{s_3}, D_1 \dots D_{s_4}$ )。不过你可以同时修补2处破损。但是这2处破损，只能是同一件衣服上的。就是说你只能同时修补一件衣服，修好了，才能修补下一件。

## Input

本题包含5行数据：第1行，为 $s_1, s_2, s_3, s_4$  ( $1 \leq s_1, s_2, s_3, s_4 \leq 20$ )。

第2行，为 $A_1 \dots A_{s_1}$  共 $s_1$ 个数，表示第一件衣服上每个破损修好它所需的时间。

第3行，为 $B_1 \dots B_{s_2}$  共 $s_2$ 个数，表示第二件衣服上每个破损修好它所需的时间。

第4行，为 $C_1 \dots C_{s_3}$  共 $s_3$ 个数，表示第三件衣服上每个破损修好它所需的时间。

第5行，为 $D_1 \dots D_{s_4}$  共 $s_4$ 个数，表示第四件衣服上每个破损修好它所需的时间 ( $1 \leq A_1 \dots A_{s_1}, B_1 \dots B_{s_2}, C_1 \dots C_{s_3}, D_1 \dots D_{s_4} \leq 60$ )。

## Output

输出一行，为修好四件衣服所要的最短时间。



中国计算机学会  
China Computer Federation

## Sample Input

1 2 1 3

5

4 3

6

2 4 3

## Sample Output

20





中国计算机学会

China Computer Federation

- $a[i]$ 表示某件衣服上第 $j$ 个破洞需要修补的时间
- $f[i][j]$ 表示补前 $i$ 个破洞时间为 $j$ 能消耗的最大时间 ( $1 \leq j \leq (\sum_{i=1}^n a[i]) \setminus 2$ )
- $f[i][j] = \begin{cases} f[i-1][j] & a[i] > j \\ f[i-1][j-a[i]] + a[i] & a[i] \leq j \end{cases}$
- 一件衣服最少耗时  $ans = \sum_{i=1}^n a[i] - f[n][ (\sum_{i=1}^n a[i]) \setminus 2 ]$



中国计算机学会  
China Computer Federation

# 光光的作业

Time Limit:1000MS Memory Limit: 128MB

## Description

光光上了高中，科目增多了。在长假里，光光的老师们都非常严厉，都给他布置了一定量的作业。假期里，光光一共有时间是 $k$ 小时。

在长假前，老师们一共给光光布置了 $n$ 份作业，第 $i$ 份作业需要的时间是 $t_i$ 小时。但是由于老师们互相不商量，因此光光有可能不能完成老师的作业。当不能完成老师的作业时，光光就事后去向老师说明，然后被老师批评一顿了事。对于一件作业，只有2种情况：完成或者不完成（快要完成也算不完成）。如果没完成，受到批评是天经地义的。但是，不同的作业对于光光来说，批评的力度是不同的。第 $i$ 件作业如果没完成，就要受到 $p_i$ 个单位的批评。多次这样之后，光光想要在长假前就知道他至少会受到多少个单位的批评。你能帮助他吗？



中国计算机学会

China Computer Federation

## Input

输入包含以下内容：第一行只有一个数字 $k$ ,表示光光一共有时间数;第二行只有一个数字 $n$ ,表示作业数;接下来 $n$ 行, 每行两个数字, 分别是 $t_i$ 和 $p_i$ , 两个数字之间用一个空格分开。

## Output

输出只包含一行, 该行只有一个数字, 代表了光光最少受到的批评。

## Sample Input

```
5
3
2 6
1 3
4 7
```

## Sample Output

```
6
```





中国计算机学会

China Computer Federation

- $t[i]$ 表示第 $i$ 份作业需要时间,  $p[i]$ 表示第 $i$ 份作业未完成受到批评
- 若一份作业都未完成, 将受到 $sum = \sum p[i]$ 的批评; 完成作业 $i$ , 即可免受 $p[i]$ 批评
- (1)
  - $f[i][j]$ 表示前 $i$ 份作业用时 $j$ 免受最大批评量
  - $$f[i][j] = \begin{cases} f[i-1][j] & t[i] > j \\ f[i-1][j-t[i]] + p[i] & t[i] \leq j \end{cases}$$
  - $ans = sum - f[n][k]$
- (2)降维
  - $f[i]$ 表示用时 $j$ 免受最大批评量
  - $f[i] = f[i-t[j]] + p[j]$



中国计算机学会  
China Computer Federation

# 挤牛奶

Time Limit: 1000MS Memory Limit: 128MB

## Description

小卡卡终于帮农夫John找到了走丢的那一头奶牛，John为了感谢小卡卡，不仅告诉了他在Pascal山脉上可能存在Pascal圣地最大的宝藏，还说要请小卡卡喝牛奶。可是农夫John发现他家里所储藏的牛奶已经喝光了，所以要临时给奶牛挤奶。小卡卡实在是太好心了，说要帮农夫John一起挤牛奶。John答应了，他把他的 $n$ 头奶牛中的 $n/2$ 头（ $n$ 是个偶数）分给小卡卡挤，他自己挤剩下的 $n/2$ 头奶牛。但是每一头奶牛都有不同的产奶量，农夫John为了让小卡卡减轻负担，他希望他们两个所挤的牛奶的总量之差最小。小卡卡得知了每头奶牛的产奶情况之后很快就解决了这个问题。



中国计算机学会  
China Computer Federation

## Input

测试数据第一行一个整数 $n$ ， $n$ 为偶数且小于100。表示有 $n$ 头奶牛。第二行 $n$ 个数分别给出每头奶牛的产奶量（产奶量的总和不超过2000）。

## Output

输出小卡卡和农夫所挤的牛奶量的最小差值。

## Sample Input

```
6
7 9 2 6 4 2
```

## Sample Output

```
0
```



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个头奶牛的产奶量
- $f[i][j]$ 表示以用 $i$ 头奶牛能否产奶量和为 $j$
- $f[i][j] = f[i][j] \vee f[i-1][j-a[k]]$  如果 $j \geq a[k]$



## Description

你现在拿到了许多的礼物，你要把这些礼物放进袋子里。你只有一个最多装下 $V$  体积物品的袋子，你不能全部放进去。你也拿不动那么重的东西。你估计你能拿的最大重量为 $G$ 。现在你了解了每一个物品的完美值、重量和体积，你当然想让袋子中装的物品的完美值总和最大，你又得计划一下了。

## Input

第一行： $G$  和  $V$  表示最大重量和体积。 第二行： $N$  表示拿到  $N$  件礼物。 第三到 $N+2$ 行：每行3个数  $T_i$   $G_i$   $V_i$  表示各礼物的完美值、重量和体积。

## Output

输出共一个数，表示可能获得的最大完美值。



中国计算机学会  
China Computer Federation

## Sample Input

```
6 5
4
10 2 2
20 3 2
40 4 3
30 3 3
```

## Sample Output

```
50
```

## Constraint

对于20%的数据  $N, V, G, T_i, V_i, G_i \leq 10$

对于50%的数据  $N, V, G, T_i, V_i, G_i \leq 100$

对于80%的数据  $N, V, G, T_i, V_i, G_i \leq 300$

80%到100%的数据是  $N, V, G, T_i, V_i, G_i \leq 380$  的离散随机数据。



中国计算机学会

China Computer Federation

- $t[i]$ 、 $g[i]$ 、 $v[i]$ 表示第 $i$ 个礼物的完美值、重量和体积
- $f[i][j]$ 表示重量为 $i$ ，体积为 $j$ 的最大完美值
- $f[i][j] = \max\{f[i - g[k]][j - v[k]] + t[k]\}$



中国计算机学会

China Computer Federation

# 一般性的最少硬币组成问题

## Description

Time Limit: 1000MS Memory Limit: 128MB

从n种币值为 $a[1..n]$ 的硬币中,任选几个硬币组成价值为V的一堆货币,问最少需要几个硬币?其中每种硬币的数量没有限制。

$1 \leq n \leq 100, 1 \leq v \leq 100000, 1 \leq a[i] \leq 100000$

## Input

输入中有两行:第一行有两个数v和n;第二行有n个以空格分隔的数,表示n个币值。

## Output

输出只有一行,该行只有一个数,表示所需的最少硬币数,如果无论如何选取硬币,均不能得到币值v,则输出0。

## Sample Input

10 2

3 5

## Sample Output

2





中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 种硬币的价值
- (1)
  - $f[i][j]$ 表示前 $i$ 种硬币形成价值为 $j$ 使用的最少硬币数量(值为-1表示不能形成价值 $j$ )
  - $$f[i][j] = \begin{cases} f[i-1][j] & t[i] > j \vee f[i-1][j-a[i]] == -1 \\ f[i-1][j-a[i]] + 1 & t[i] \leq j \& f[i-1][j-a[i]] \neq -1 \end{cases}$$
  - $ans = f[v]$
- (2)降维
  - $f[j]$ 表示达到价值 $j$ 使用的最少硬币数量
  - $f[i] = f[i-a[j]] + 1$  如果 $f[i-a[j]] \neq -1$



# 多个公司间的机器分配问题

## Description

Time Limit:1000MS Memory Limit: 128MB

已知第 $j$ 个公司使用 $k$ 台机器时,能得到的利润为 $a[j,k]$ ,问如何将 $m$ 台机器在 $n$ 个公司中分配,才能获得最大利润?要求输出能获得的最大利润及方案.将3台机器分配给2个公司能获得的盈利情况如下:

公司号\机器数	1	2	3
1	2	3	4
2	1	4	5

最大盈利为6,方案为公司2使用2台,公司1使用1台.

## Input

第1行 $n, m$  分别表示公司数和机器数, 第2至第 $n+1$ 行分别表示第 $i$ 个公司分别使用每台机器的盈利情况, 可结合题目描述进行理解。

## Output

输出共一个数, 最大的盈利值为多少。



中国计算机学会  
China Computer Federation

## Sample Input

2 3

2 3 4

1 4 5

## Sample Output

6

## Constraint

数据范围：机器数 $m \leq 300$ ,

公司数 $n \leq 300$ ;

最大总利润不超过longint范围



中国计算机学会

China Computer Federation

- $a[i][j]$ 表示第 $i$ 个公司使用第 $j$ 台机器的利润
- $f[i][j]$ 表示前 $i$ 个公司使用前 $j$ 台机器所得的最大利润
- $f[i][j] = \begin{cases} f[i-1][j] \\ f[i-1][j-k] + a[i][k] \end{cases}$
- 可降维



## Description

小XC发现垒城堡的时候,如果下面的积木比上面的积木大,那么城堡便不易倒。所以他在垒城堡的时候总是遵循这样的规则。小XC想把自己垒的城堡送给幼儿园里同学们,这样可以增加他的好感度。为了公平起见,他决定把送给每个同学一样高的城堡,这样可以避免同学们为了获得更漂亮的城堡而引起争执。可是他发现自己在垒城堡的时候并没有预先考虑到这一点。所以他现在要改造城堡。由于他没有多余的积木了,他灵机一动,想出了一个巧妙的改造方案。他决定从每一个城堡中挪去一些积木,使得最终每座城堡都一样高。为了使他的城堡更雄伟,他觉得应该使最后的城堡都尽可能的高。

任务: 请你帮助小XC编一个程序, 根据他垒的所有城堡的信息, 决定应该移去哪些积木才能获得最佳的效果。



中国计算机学会

China Computer Federation

## Input

第一行是一个整数 $N$  ( $N \leq 100$ ), 表示一共有几座城堡。以下 $N$ 行每行是一系列非负整数, 用一个空格分隔, 按从下往上的顺序依次给出一座城堡中所有积木的棱长。用-1结束。一座城堡中的积木不超过100块, 每块积木的棱长不超过100。

## Output

一个整数, 表示最后城堡的最大可能的高度。如果找不到合适的方案, 则输出0。

## Sample Input

```
2
2 1 -1
3 2 1 -1
```

## Sample Output

```
3
```



中国计算机学会

China Computer Federation

- $a[i][j]$ 表示第 $i$ 个城堡的第 $j$ 块积木高度
- $f[i][j]=\text{true}$ 表示第 $i$ 个城堡能垒成高度为 $j$ 的城堡，否则为不能
- $\text{sum}[i]$ 表示第 $i$ 个城堡能垒成的最高高度，即 $\text{sum}[i] = \sum_{j=1}^{j=\text{num}[i]} a[i][j]$
- $f[i][j] = f[i][j] \parallel f[i][j - a[i][k]]$
- 对于每个 $i$ ，都有 $f[i][j]=\text{true}$ ，最大的 $j$ 即为答案



中国计算机学会  
China Computer Federation

# 生物基元问题

Time Limit: 1000MS Memory Limit: 128MB

## Description

一个生物体的结构可以用“基元”的序列表示，一个“基元”用一些英文字符串表示。对于一个基元集合 $P$ ，可以将字符串 $S$ 看作由 $n$ 个基元 $P_1, P_2, \dots, P_n$ 依次连接而成的。问题是给定一个字符串 $S$ 和一个基元集合 $P$ ，使 $S$ 的前缀可由 $P$ 中的基元组成。求这个前缀的最大长度。基元的个数最大为20，长度最大为20，字符中的长度最大为500000。

例如基元集合为 $\{A, AB, BBC, CA\}$ ，字符串为 $ABACABBBCAACCB$ ，则最大长度为10，其具体组成为 $ABACABBCCA$  2214433311





中国计算机学会

China Computer Federation

## Input

第一行为字符串S，第二行为一个整数m，表示基元数 以下m行，每行一个基元。

## Output

只有一个整数，表示字符串S中能得到的最大前缀长度。

## Sample Input

ABACABBCCAACCB

4

A

AB

BBC

CA

## Sample Output

10



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个基元,  $s$ 表示字符串
- $f[i]=\text{true}$ 表示前 $1 \sim i$ 个字符能生成前缀, 否则为不能
- $\text{sum}[i]$ 表示第 $i$ 个城堡能垒成的最高高度, 即 $\text{sum}[i] = \sum_{j=1}^{j=\text{num}[i]} a[i][j]$
- $f[i] = f[i] || f[i - \text{strlen}(a[j])]$  (如果 $a[j]$ 基元能与 $s$ 串中 $i - \text{strlen}(a[j]) + 1 \sim i$ 位置匹配)
- 如果有 $f[i]=\text{true}$ , 最大的 $i$ 即为答案



## Description

2001年9月11日，一场突发的灾难将纽约世界贸易中心大厦夷为平地，Mr. F曾亲眼目睹了这次灾难。为了纪念“911”事件，Mr. F决定自己用水晶来搭建一座双塔。Mr. F有N块水晶，每块水晶有一个高度，他想用这N块水晶搭建两座有同样高度的塔，使他们成为一座双塔，Mr. F可以从这N块水晶中任取M ( $1 \leq M \leq N$ ) 块来搭建。但是他不知道能否使两座塔有同样的高度，也不知道如果能搭建成一座双塔，这座双塔的最大高度是多少。所以他来请你帮忙。

给定水晶的数量N ( $1 \leq N \leq 100$ ) 和每块水晶的高度 $H_i$  (N块水晶高度的总和不超过2000)，你的任务是判断Mr. F能否用这些水晶搭建成一座双塔（两座塔有同样的高度），如果能，则输出所能搭建的双塔的最大高度，否则输出“Impossible”。



中国计算机学会

China Computer Federation

## Input

输入的第一行为一个数 $N$ ，表示水晶的数量。

第二行为 $N$ 个数，第 $i$ 个数表示第 $i$ 个水晶的高度。

## Output

输出仅包含一行，如果能搭成一座双塔，则输出双塔的最大高度，否则输出一个字符串 "Impossible"。

## Sample Input

5

1 3 4 5 2

## Sample Output

7



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个水晶的高度
- (1)
  - $f[i][j][k]=\text{true}$ 表示前 $i$ 个水晶能左边搭成高度 $j$ ，右边搭成高度 $k$ ；否则为不能。
  - $f[i][j][k] = f[i-1][j][k] \parallel f[i][j-a[i]][k] \parallel f[i][j][k-a[i]]$
- (2)
  - $f[i][j]$ 表示前 $i$ 个水晶搭成双塔高度差为 $j$ ，第一座塔的最大高度
  - $f[i][j] = \max\{f[i-1][j], f[i-1][j-a[i]], f[i-1][j+a[i]] + a[i]\}$



# 加分二叉树

Time Limit: 1000MS Memory Limit: 128MB

## Description

设一个 $n$ 个节点的二叉树 $tree$ 的中序遍历为 $(1, 2, 3, \dots, n)$ , 其中数字 $1, 2, 3, \dots, n$ 为节点编号。每个节点都有一个分数（均为正整数），记第 $j$ 个节点的分数为 $d_j$ ， $tree$ 及它的每个子树都有一个加分，任一棵子树 $subtree$ （也包含 $tree$ 本身）的加分计算方法如下：

$subtree$ 的左子树的加分  $\times$   $subtree$ 的右子树的加分  $+$   $subtree$ 的根节点的分数

若某个子树为空, 规定其加分为1, 叶子的加分就是叶节点本身的分数。不考虑它的空子树。

试求一棵符合中序遍历为 $(1, 2, 3, \dots, n)$ 且加分最高的二叉树 $tree$ 。要求输出：（1）

$tree$ 的最高加分 （2） $tree$ 的前序遍历。



中国计算机学会

China Computer Federation

## Input

第1行：一个整数 $n$  ( $n < 30$ ) 为节点个数。

第2行： $n$ 个用空格隔开的整数, 为每个节点的分数 (分数  $< 100$ )。

## Output

第1行：一个整数，为最高加分（结果不会超过4,000,000,000）。

第2行： $n$ 个用空格隔开的整数，为该树的前序遍历。

## Sample Input

5

5 7 1 2 10

## Sample Output

145

3 1 2 4 5



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个节点的分数
- $f[i][j]$ 表示 $i$ 到 $j$ 节点生成的二叉树能得到的最大分数

$$f[i][j] = \begin{cases} a[i] & i == j \\ a[i] + a[j] & i + 1 == j \\ \max\{f[i][k-1] * f[k+1][j] + a[k]\} & j - i > 1 \& i < k < j \end{cases}$$





## Description

问题描述： 今年是中国计算机学会确定的“2000——世界数学年”，又恰逢我国著名数学家华罗庚先生 诞辰90周年。

在华罗庚先生的家乡江苏金坛,组织了一场别开生面的数学智力竞赛的活动,你的一个好朋友XZ也有幸得以参加。活动中,主持人给所有参加活动的选手出了这样一道题目：设有一个长度为N的数字串，要求选手使用K个乘号将它分成K+1个部分，找出一种分法，使得这K+1个部分的乘积能够为最大。

同时，为了帮助选手能够正确理解题意，主持人还举了如下的一个例子：有一个数字串：312，当N=3，K=1时会有以下两种分法：

1)  $3*12=36$

2)  $31*2=62$

这时，符合题目要求的结果是： $31*2=62$ 。

现在，请你帮助你的好朋友XZ设计一个程序，求得正确的答案。



中国计算机学会

China Computer Federation

## Input

第1行：共有2个自然数 $N, K$  ( $6 \leq N \leq 40, 1 \leq K \leq 6$ )

第二行是一个长度为 $N$ 的数字串。

## Output

输出所求得的最大乘积（一个自然数）。

## Sample Input

4 2

1231

## Sample Output

62



中国计算机学会

China Computer Federation

- $s$ 表示该字符串,  $num[i][j]$ 表示字符串 $s[i]$ 至 $s[j]$ 形成的数
- $f[i][j]$ 表示前 $i$ 个数中放 $j$ 个乘号得到的最大值
- $f[i][j] = \begin{cases} num[1][i] & j = 0 \\ \max\{f[k][j-1] * num[k+1][i]\} & j-1 < k < i \end{cases}$



# 数字游戏

Time Limit:1000MS Memory Limit: 128MB

## Description

丁丁最近沉迷于一个数字游戏之中。这个游戏看似简单,但丁丁在研究了许多天之后却发现原来在简单的规则下想要赢得这个游戏并不容易。

游戏是这样的,在你面前有一圈整数(一共 $n$ 个),你要按顺序将其分为 $m$ 个部分,各部分内的数字相加,相加所得的 $m$ 个结果对10取模后再相乘,最终得到一个数 $k$ 。游戏的要求是使你所得的 $k$ 最大或者最小。

例如,对于下面这圈数字 ( $n=4$ ,  $m=2$ )  $[4\ 2\ -1\ 3]$ : 当要求最小值时, $((2-1) \bmod 10) \times ((4+3) \bmod 10) = 1 \times 7 = 7$ ,要求最大值时,为  $((2+4+3) \bmod 10) \times (-1 \bmod 10) = 9 \times 9 = 81$ 。特别值得注意的是,无论是负数还是正数,对10取模的结果均为非负值。丁丁请你编写程序帮他赢得这个游戏。



中国计算机学会

China Computer Federation

## Input

输入第一行有两个整数,  $n$  ( $1 \leq n \leq 50$ ) 和  $m$  ( $1 \leq m \leq 9$ )。

以下  $n$  行每行有个整数, 其绝对值不大于  $10^4$ , 按顺序给出圈中的数字, 首尾相接。

## Output

输出有两行, 各包含一个非负整数。

第一行是你程序得到的最小值, 第二行是最大值。

## Sample Input

4 2

4

3

-1

2

## Sample Output

7

81



中国计算机学会

China Computer Federation

- $s[i][j]$ 表示*i*位置数字至*j*位置数字和对10求余的值
- $f[i][j][k]$ 表示从*i*至*j*分成*k*个部分得到的最大乘积
- $f[i][j][k] = \max\{f[i][h][k-1] * s[h+1][j]\}$  其中  $i+k-1 \leq h < j$
- 同理可定义最小乘积



中国计算机学会  
China Computer Federation

# 多边形的三角划分

Time Limit: 1000MS Memory Limit: 128MB

## Description

N个顶点的凸多边形[顶点顺序为1->N]，各顶点权值已知，要求划分成N-2个三角形，使各三角形顶点权值乘积之和为最小。

当n=4，各顶点的权值分别为10,5,7,6时，所求最小值为 $10*5*6+5*6*7=510$ 。

## Input

第一行：一个整数n。

第二行：n个整数,依次表示各顶点的权值。

## Output

一个整数表示最小的乘积之和。



中国计算机学会  
China Computer Federation

## Sample Input

4

10 5 7 6

## Sample Output

510

## Constraint

$n \leq 200$ , 所有输入数据均  $\leq 1000$ , 所求得的最小值小于10的9次方。





中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个顶点的权值
- $f[i][j]$ 表示从 $i$ 至 $j$ 个顶点最小乘积之和

$$f[i][j] = \begin{cases} 0 & i+1 \geq j \\ \max\{f[i][k] + f[k][j] + a[i] * a[j] * a[k]\} & i < k < j \end{cases}$$



中国计算机学会  
China Computer Federation

# 二叉树数

Time Limit: 1000MS    Memory Limit: 128MB

## Description

求由 $n$ 个结点构成的不同的二叉树数 ( $n \leq 100$ )。  
每个节点均认为是等价的!

## Input

一个整数 $n$ 。

## Output

一个整数表示不同的二叉树数。

## Sample Input

3

## Sample Output

5



中国计算机学会

China Computer Federation

- $f[i]$ 表示节点数为 $n$ 的二叉树数量
- $f[i] = \sum_{j=0}^{i-1} f[j] * f[i-j-1]$





# 石子合并

## Description

Time Limit:1000MS Memory Limit: 128MB

有 $n$ 堆石子围成一个圆圈。现在需要把它们合并成一堆石子。每次合并时，只能合并相邻的两堆石子，所耗力气为两堆石子重量之和，合并得到的新堆的重量为原两堆重量之和。问最少需要耗费多少力气？

数据规模：  $1 \leq n \leq 200$ ，合并 $n$ 堆石子最少需要耗费的力气不超过 $2 \times 10^9$ 。

## Input

第一行一个整数 $n$ ，第二行 $n$ 个整数，表示顺序排列的每堆石子的重量。

## Output

只有一行，该行只有一个整数，表示合并这 $n$ 堆石子最少需要耗费的力气。

## Sample Input

```
3
1 3 5
```

## Sample Output

```
13
```



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 堆石子的重量
- $s[i]$ 表示前 $i$ 堆石子的重量和
- $f[i][j]$ 表示从 $i$ 至 $j$ 堆石子合并成一堆耗费的最少力气
- $f[i][j] = \min\{f[i][k] + f[k+1][j] + s[j] - s[i]\}$  其中 $i \leq k < j$



中国计算机学会  
China Computer Federation

# 1的最优操作序列

Time Limit:1000MS Memory Limit: 128MB

## Description

在黑板上写了 $N$  ( $n \leq 10000$ ) 个1, 进行如下操作: 每次擦去其中两个数 $a$ 、 $b$ , 并写上数 $a*b+1$ , 如此下去直至最后一个数 $A$ 。

请你编程序求出 $A$ 的最大值。

## Input

只有一个整数 $n$ 。

## Output

也只有一个整数,表示得到的最大值。

## Sample Input

5

## Sample Output

7



中国计算机学会

China Computer Federation

- $f[i]$ 表示 $i$ 个1能达到的最大值
- $f[i] = \max\{f[j] * f[i - j] + 1\}$  其中 $j < i$





中国计算机学会  
China Computer Federation

# 排序工作量之新任务

Time Limit: 1000MS Memory Limit: 128MB

## Description

假设我们将序列中第 $i$ 件物品的参数定义为 $A_i$ ，那么排序就是指将 $A_1, \dots, A_n$ 从小到大排序。若 $i < j$ 且 $A_i > A_j$ ，则 $\langle i, j \rangle$ 就为一个“逆序对”。

SORT公司是一个专门为用户提供排序服务的公司，他们的收费标准就是被要求排序物品的“逆序对”的个数，简称“逆序数”。Grant是这家公司的排序员，他想知道对于 $n$ 个参数都不同的物品组成的序列集合中，逆序对数为 $t$ 的物品序列有多少个，并试给出其中一个最小的物品序列。

所谓最小，即若有两个物品序列 $(A_1, A_2, \dots, A_n)$ ， $(B_1, B_2, \dots, B_n)$ ，存在 $1 \leq i \leq n$ ，使得 $(A_1, A_2, \dots, A_{i-1}) = (B_1, B_2, \dots, B_{i-1})$ 且 $A_i < B_i$ 。





中国计算机学会  
China Computer Federation

## Input

只有一行，该行只有两个整数 $n$ 和 $t$  ( $1 \leq n \leq 20$ ,  $0 \leq t \leq n*(n-1)/2$  )。

## Output

有二行，第一行只有一个数,表示 $n$ 个参数都不同的物品组成的序列集合中,逆序数为 $t$ 的序列个数；第二行是所求物品参数序列。假设 $n$ 个物品的参数分别为1到 $n$ 。每个数字后面有一个空格，包括最后一个数字。

## Sample Input

4 3

## Sample Output

6

1 4 3 2



中国计算机学会

China Computer Federation

- $f[i][j]$ 表示前 $i$ 个数中逆序对数有 $j$ 个的数量

- $f[i][j] = \sum_{k=0}^{k=\min(i-1,j)} f[i-1][j-k]$

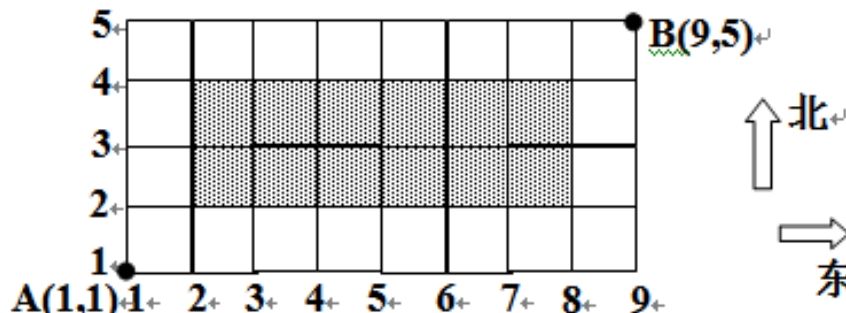


# 街道路径条数

## Description

Time Limit: 1000MS Memory Limit: 128MB

设有一个 $N \times M$  ( $1 \leq N \leq 50, 1 \leq M \leq 50$ ) 的街道(如下图):



规定行人从 $A(1,1)$ 出发,在街道上只能向东或向北方向行走。如图,从 $(1,1)$ 点出发,至 $(3,3)$ 点,共有6条不同的路径:  
 $(1,1)-(2,1)-(3,1)-(3,2)-(3,3)$ ;  $(1,1)-(2,1)-(2,2)-(3,2)-(3,3)$ ;  $(1,1)-(2,1)-(2,2)-(2,3)-(3,3)$ ;  $(1,1)-(1,2)-(2,2)-(3,2)-(3,3)$ ;  $(1,1)-(1,2)-(2,2)-(2,3)-(3,3)$ ;  $(1,1)-(1,2)-(1,3)-(2,3)-(3,3)$ 。

若在 $N \times M$ 的街道中,设置一个矩形障碍区域(包括围住该区域的街道)不让行人通行,如图中用阴影线表示的部分。此矩形障碍区域可以用2对顶点坐标给出,如上图中的障碍区域以2对顶点坐标 $(2,2)$ ,  $(8,4)$ 表示。此时,从 $A(1,1)$ 出发至 $B(9,5)$ ,只有两条路径:

路径一:  $(1,1)-(2,1)-(3,1)-(4,1)-(5,1)-(6,1)-(7,1)-(8,1)-(9,1)-(9,2)-(9,3)-(9,4)-(9,5)$

路径二:  $(1,1)-(1,2)-(1,3)-(1,4)-(1,5)-(2,5)-(3,5)-(4,5)-(5,5)-(6,5)-(7,5)-(8,5)-(9,5)$

程序要求: 给出 $N, M$ , 同时再给出此街道中的矩形障碍区域的2对顶点坐标 $(X1, Y1)$ ,  $(X2, Y2)$ , 然后求出此种情况下所有从 $(1,1)$ 出发到达 $(N, M)$ 的路径的条数。



中国计算机学会

China Computer Federation

## Input

第一行有两个数字,表示N和M; 第二行有两个数字,表示矩形障碍的左下角坐标;  
第三行有两个数字,表示矩形障碍的右上角坐标。

## Output

只有一个数字,表示求得的路径条数。

## Sample Input

9 5

2 2

8 4

## Sample Output

2



中国计算机学会

China Computer Federation

- $f[i][j]$ 表示到达 $(i, j)$ 点位置的路径数量
- $f[i][j] = f[i - 1][j] + f[i][j - 1]$  其中 $(i, j)$ 不在矩形障碍区域内

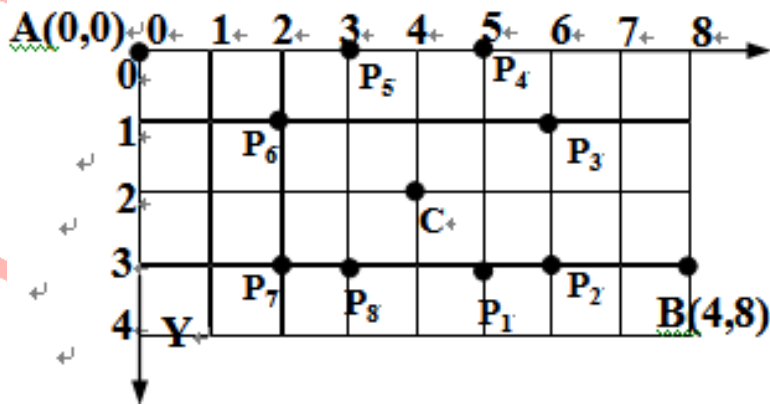


## Description

如下图,A 点有一个过河卒,需要走到目标 B 点。卒行走规则:可以 向下、或者向右。同时在棋盘上的某一点有一个对方的马(如上图的C 点),该马所在的点和所有跳跃一步可达的点称为马的控制点。例如下图C点上的马可以控制9个点(图中的P1,P2,...,P8和C)。

卒不能通过对方马的控制点。

棋盘用坐标表示,A 点坐标为(0,0)、B 点坐标为(n,m) (n,m为不超过 20 的整数),同样马的位置坐标是需要给出的(约定点:C≠点A,同时点C≠点B)。现在要求你计算出卒从A 点到达 B 点的路径的条数。





中国计算机学会

China Computer Federation

## Input

只有一行数据,该行中有4个以空格分隔的数,表示B点的坐标和马的坐标

## Output

只有一行数据,该行只有一个数,表示求得的路径条数。

## Sample Input

6 6 3 2

## Sample Output

17



中国计算机学会

China Computer Federation

- $f[i][j]$ 表示到达 $(i, j)$ 点位置的路径数量
- $g[i][j]$ 表示 $(i, j)$ 是否为马的控制点, 1为是, 0为不是
- $f[i][j] = \begin{cases} 0 & g[i][j] == 1 \\ f[i-1][j] + f[i][j-1] & g[i][j] == 0 \end{cases}$





中国计算机学会  
China Computer Federation

# 统计正方形和长方形个数

Time Limit: 1000MS Memory Limit: 128MB

## Description

设有一个 $n*m$ 方格的棋盘( $1 \leq m, n \leq 1000$ ), 求出该棋盘中包含多少个正方形、多少个长方形(不包括正方形)。例如: 当 $n=2, m=3$ 时, 正方形的个数有8个; 长方形的个数有10个。

## Input

只有一行数据, 包含2个以逗号分隔的数, 分别代表 $n$ 和 $m$

## Output

只有一行数据, 包含2个以逗号分隔的数, 分别代表正方形的个数和长方形的个数。

## Sample Input

2,3

## Sample Output

8,10



中国计算机学会

China Computer Federation

- 长方形:  $\frac{n(n+1)}{2} * \frac{m(m+1)}{2}$
- $f[i]$ 表示边长为 $i$ 的正方形个数
- $f[i] = (n - i + 1) * (m - i + 1)$
- 正方形: 所有 $f[i]$ 的总和



# 骑士游历

Time Limit: 1000MS Memory Limit: 128MB

## Description

设有一个 $n*m$ 的棋盘( $2 \leq n \leq 50, 2 \leq m \leq 50$ ), 在棋盘上左下角 $(1,1)$ 处有一个中国象棋马。马走的规则为: (1) 马走日字; (2) 马只能向右走。如图1所示。

当 $n, m$ 给出之后, 同时给出马起点的位置和终点的位置, 试找出从起点到终点的所有路径的数目。

如图3所示, 给出马的起点坐标为 $(1,8)$ , 终点坐标为 $(3,8)$ , 则有2条路径。

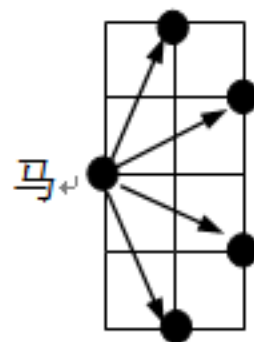
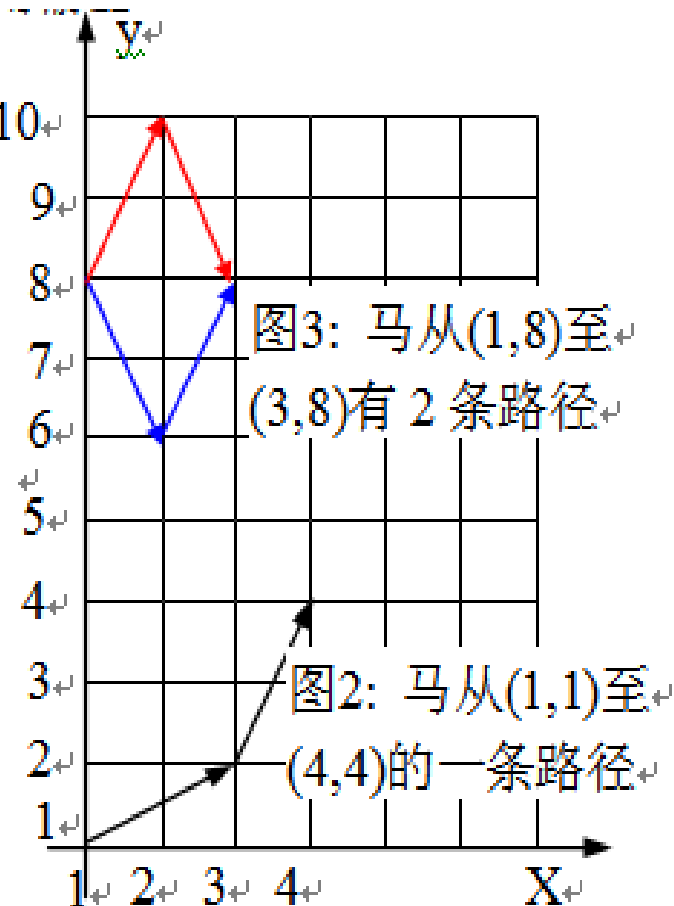


图1 马的4种走法





中国计算机学会

China Computer Federation

## Input

第1行有两个数,表示右上角坐标( $n,m$ )

第2行有两个数,表示起点坐标( $x_1,y_1$ )

第3行有两个数,表示终点坐标( $x_2,y_2$ )

## Output

输出一个数,表示路径数。

## Sample Input

10 10

1 8

3 8

## Sample Output

2



中国计算机学会

China Computer Federation

- $f[i][j]$ 表示到达 $(i, j)$ 点位置的路径数量
- 每个点可由四个点跳至
- $f[i][j] = f[i-2][j+1] + f[i-2][j-1] + f[i-1][j+2] + f[i-1][j-2]$



# 方格取数

Time Limit: 1000MS Memory Limit: 128MB

## Description

设有 $N \times N$ 的方格图( $N \leq 30$ ),我们将其中的某些方格中填入正整数,而其他的方格中则放入数字0。

某人从图的左上角的A点出发,可以向下行走,也可以向右走,直到到达右下角的B点。在走过的路上,他可以取走方格中的数(取走后的方格中将变为数字0)。此人从A点到B点共走两次,试找出2条这样的路径,使得取得的数之和为最大。

## Input

输入第一行为一个整数 $N$ (表示 $N \times N$ 的方格图),接下来的每行有三个整数,前两个表示位置,第三个数为该位置上所放的数。

一行单独的0表示输入结束。

## Output

输出只有一行数据,该行只有一个数字,表示2条路径上取得的最大的和。



中国计算机学会  
China Computer Federation

## Sample Input

```
8
2 3 13
2 6 6
3 5 7
4 4 14
5 2 21
5 6 4
6 3 15
7 2 14
0 0 0
```

## Sample Output

67





中国计算机学会

China Computer Federation

- $a[i][j]$ 表示 $(i,j)$ 位置的数
- $f[x1][y1][x2][y2]$ 表示两条路线走至 $(x1,y1)$ 和 $(x2,y2)$ 时得到的最大和
- $$f[x1][y1][x2][y2] = \max \begin{cases} f[x1-1][y1][x2-1][y2] \\ f[x1][y1-1][x2-1][y2] \\ f[x1-1][y1][x2][y2-1] \\ f[x1][y1-1][x2][y2-1] \end{cases} + \begin{cases} a[x1][y1] + a[x2][y2] \\ a[x1][y1] \end{cases} \quad \begin{matrix} \text{其他情况} \\ x1 == x2 \& \& y1 == y2 \end{matrix}$$
- 两条路线走的步数相同, 可知 $x1+y1==x2+y2$ , 从而可以降维为 $f[k][x1][x2]$ , 其中 $k=x1+y1$





## Description

栈是计算机中经典的数据结构,简单的说,栈就是限制在一端进行插入删除操作的线性表。栈有两种最重要的操作,即pop (从栈顶弹出一个元素) 和push (将一个元素进栈)。栈的重要性不言而喻,任何一门数据结构的课程都会介绍栈。宁宁在复习栈的基本概念时,想到了一个书上没有讲过的问题,而他自己无法给出答案,所以需要你的帮忙。

宁宁考虑的是这样一个问题: 一个操作数序列, 从1, 2, ...一直到n, 栈A的深度大于n。现在可以进行两种操作, 1. 将一个数, 从操作数序列的头端移到栈的头端 (对应数据结构栈的push操作) 2. 将一个数, 从栈的头端移到输出序列的尾端 (对应数据结构栈的pop操作) 使用这两种操作, 由一个操作数序列就可以得到一系列的输出序列。

你的程序将对给定的n, 计算并输出由操作数序列1, 2, ..., n经过操作可能得到的输出序列的总数。



中国计算机学会

China Computer Federation

## Input

只含一个整数 $n$  ( $1 \leq n \leq 500$ )

## Output

只有一行，即可能输出序列的总数目

## Sample Input

3

## Sample Output

5

## Constraint

【思考】如果 $1 \leq n \leq 3000$ ，如何做？



中国计算机学会

China Computer Federation

- $f[i]$ 表示前 $i$ 个数的序列总数
- $f[i] = \sum_{j=0}^{i-1} f[j] * f[i-j-1]$





# 花店橱窗布置问题

Time Limit:1000MS Memory Limit: 128MB

## Description

假设想以最美观的方式布置花店的橱窗,有F束花,每束花的品种都不一样,同时,至少有同样数量的花瓶被按顺序摆成一行,花瓶的位置是固定的,并按从左到右,从1到V顺序编号,V是花瓶的数目,编号为1的花瓶在最左边,编号为V的花瓶在最右边,花束可以移动,并且每束花用1到F的整数唯一标志.标志花束的整数决定了花束在花瓶中排列的顺序,即如果 $i < j$ ,则花束i必须放在花束j左边的花瓶中.

例如,假设杜鹃花的标志数为1,秋海棠的标志数为2,康乃馨的标志数为3,所有的花束在放入花瓶时必须保持其标志数的顺序,即:杜鹃花必须放在秋海棠左边的花瓶中,秋海棠必须放在康乃馨左边的花瓶中.如果花瓶的数目大于花束的数目,则多余的花瓶必须为空,即每个花瓶中只能放一束花.每一个花瓶的形状和颜色也不相同,因此,当各个花瓶中放入不同的花束时,会产生不同的美学值(一个整数)来表示,空置花瓶的美学值为0.在上述例子中,花瓶与花束的不同搭配所具有的美学值,可以用如上所示的表格表示:根据上表,杜鹃花放在花瓶2中,会显得很好看,但若放在花瓶4中则显得很难看.

为取得最佳美学效果,必须在保持花束顺序的前提下,使花的摆放取得最大的美学值.题中数据满足下面条件: $1 \leq F \leq 1000$ ,  $F \leq V \leq 1000$ ,  $-50 \leq A_{ij} \leq 50$ ,其中 $A_{ij}$ 是花束i摆放在花瓶j中的美学值.

	花瓶 1	花瓶 2	花瓶 3	花瓶 4	花瓶 5
杜鹃花	7	23	-5	-24	16
秋海棠	5	21	-4	10	23
康乃馨	-21	5	-4	-20	20



中国计算机学会

China Computer Federation

## Input

输入整数 $F$ ， $V$ 和矩阵 $A$ 。

## Output

输出最大美学值和每束花摆放在各个花瓶中的花瓶编号。如果最佳摆放方案不止一个，则输出按照第1束花的花瓶编号较小的方案，如果第1束花的花瓶编号相同，则输出第2束花的花瓶编号较小者，依次类推。

相邻数字用空格隔开，最后一个数字后面没有多余的空格！

## Sample Input

```
3 5
-3 10 12 49 50
5 21 -4 10 23
-21 -4 50 -20 20
```

## Sample Output

```
68
1 2 3
```



中国计算机学会

China Computer Federation

- $a[i][j]$ 表示第 $i$ 种花放在第 $j$ 个花瓶中的美学值
- $f[i][j]$ 表示前 $i$ 种花用 $j$ 个花瓶的最大美学值
- $f[i][j] = \max\{f[i-1][k] + a[i][j]\}$  其中 $i-1 \leq k \leq j$





## Description

Time Limit:1000MS Memory Limit: 128MB

在一处原始森林中，发现了N个藏有宝石的山洞，这N个山洞以1至N编号。某些山洞间可能会有通道相连，山洞间的通道是单向的，编号较小的山洞可以通过通道走至编号较大的山洞，编号较大的山洞不可以通过通道走至编号较小的山洞。你可以选择任意一个山洞进入，选择一条通道，进入下一个较大编号的山洞，.....，最后从某个山洞出来，并获得所有经过的山洞中的宝石。从外面只能进入一次，然后从一个山洞至另一个山洞，直至走出山洞。现在告诉你每个山洞中的宝石数，以及山洞之间的通道情况，求最多能取得多少宝石？

## Input

第一行有两个整数n和m，分别表示山洞数和通道数，两数间以一个空格分隔。

第二行有n个整数，第一个数表示第一个山洞的宝石数,第二个数表示第二个山洞的宝石数,...,第n个数表示第n个山洞的宝石数。这n个数间互相以一个空格分隔。

以下m行，每行描述两个山洞间有一条从编号较小山洞通向编号较大山洞的单向通道。每一行的两个整数a和b,可能表示山洞a至山洞b间的单向通道，也可能表示山洞b至山洞a间的单向通道。a和b间有一个空格分隔。

## Output

只有一行，该行只有一个整数，表示最多能获得的宝石数。



中国计算机学会

China Computer Federation

## Sample Input

5 6  
10 8 4 7 6  
1 3  
1 2  
1 4  
3 4  
3 5  
4 5

## Sample Output

27

## Constraint

对于所有的数据, 获得的宝石总数不会超过 $2 \times 10^9$

30%的数据,  $1 \leq n \leq 1000$ ,  $0 \leq m \leq 10000$

100%的数据,  $1 \leq n \leq 20000$ ,  $0 \leq m \leq 100000$





中国计算机学会

China Computer Federation

- $a[i][j] = 1$  表示能从山洞  $i$  通向山洞  $j$
- $f[i]$  表示到达山洞  $i$  的最大宝石数量
- $f[i] = \max\{f[j]\} + a[i]$  如果  $a[j][i] == 1$



中国计算机学会  
China Computer Federation

# 导弹拦截

Time Limit: 1000MS Memory Limit: 128MB

## Description

某国为了防御敌国的导弹袭击，发展出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭。由于该系统还在试用阶段，所以只有一套系统，因此有可能不能拦截所有的导弹。

输入导弹依次飞来的高度（雷达给出的高度数据是不大于30000的正整数）：1) 计算这套系统最多能拦截多少导弹；2) 如果要拦截所有导弹最少要配备多少套这种导弹拦截系统。



中国计算机学会

China Computer Federation

## Input

只有一行数据,包括若干以空格分隔的正整数,表示来袭的导弹的高度.  
总共不超过350个数字。

## Output

第一行只有一个正整数,表示最多能拦截的导弹数; 第二行也只有一个正整数,表示要拦截所有导弹最少要配备的系统数。

## Sample Input

389 207 155 300 299 170 158 65

## Sample Output

6

2



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个导弹的高度
- $f[i]$ 表示截取第 $i$ 个导弹时，能截取的最大导弹数量
- $f[i] = \max\{ f[j] + 1 \}$  其中 $a[i] \leq a[j]$
- 最少的系统数量，用二分答案方法



中国计算机学会  
China Computer Federation

# 合唱队形

Time Limit:1000MS Memory Limit: 128MB

## Description

N位同学站成一排，音乐老师要请其中的(N-K)位同学出列，使得剩下的K位同学排成合唱队形。合唱队形是指这样的一种队形：设K位同学从左到右依次编号为1, 2..., K, 他们的身高分别为 $T_1, T_2, \dots, T_K$ , 则他们的身高满足 $T_1 < \dots < T_{i+1} > \dots > T_K (1 \leq i \leq K)$ 。

你的任务是，已知所有N位同学的身高，计算最少需要几位同学出列，可以使得剩下的同学排成合唱队形。



中国计算机学会

China Computer Federation

## Input

有2行数据。第一行是一个整数 $n$  ( $2 \leq n \leq 100$ ), 表示同学的总数。

第二行有 $n$ 个整数, 用空格分隔, 第 $i$ 个整数 $T_i$  ( $130 \leq T_i \leq 230$ ) 是第 $i$ 位同学的身高(厘米)。

## Output

包括一行, 这一行只包含一个整数, 就是最少需要几位同学出列。

## Sample Input

8

186 186 150 200 160 130 197 220

## Sample Output

4

## Constraint

对于50%的数据, 保证有 $n \leq 20$ ;

对于全部的数据, 保证有 $n \leq 100$ 。



中国计算机学会

China Computer Federation

- $a[i]$ 表示第 $i$ 个同学的高度
- $f[i]$ 表示从第一位同学开始，在选择第 $i$ 位同学的情况下最长上升序列
- $g[i]$ 表示选择第 $i$ 位同学，至最后一位同学最长下降序列
- $f[i] = \max\{ f[j] + 1 \}$  其中 $a[i] > a[j]$ 且 $i > j$
- $g[i] = \max\{ g[j] + 1 \}$  其中 $a[i] > a[j]$ 且 $i < j$
- $ans = \min\{ n - (f[i] + g[i] - 1) \}$



## Description

iRabbit的国家被一条河流(河流是直的)分成南北两岸，南北两岸各有N个城市。北岸的每一个城市有一个唯一的友好城市在南岸，且他们的友好城市彼此不同。为了城市关系的发展，每对城市之间都想要开通轮渡。由于河面上常常有雾。iRabbit决定禁止船只航线相交。以避免发生安全事故。iRabbit希望能在保证安全的情况下，尽可能多地开通航线。

由于N非常大 ( $1 \leq N \leq 2004$ )，所以必须用程序解决。iRabbit因为备战竞赛，所以十分繁忙，没有时间来编写程序，所以交给手下的TCR和sceoy解决。可是他们两个想了很久都没有想出答案，所以想请你来帮助解决。





中国计算机学会

China Computer Federation

### Input

第1行有1个整数:  $N$  ( $1 \leq N \leq 2004$ )。

接下来 $N$ 行,每行两个数 $A, B$ 。表示这一对友好城市与河源头的距离(不过100000), 其中 $A$ 代表北岸城市、 $B$ 代表南岸城市。

### Output

只有一个整数。表示可以开通轮渡的最大线路数。

### Sample Input

```
7
22 4
2 6
10 3
15 12
9 8
17 17
4 2
```

### Sample Output

```
4
```



中国计算机学会

China Computer Federation

- 对所有路径，按照a城市的大小升序排序，题目转换为求b城市的最长上升子序列问题
- $f[i]$ 表示选择第i条路径时的最大线路数
- $f[i] = \max\{ f[j] + 1 \}$  其中  $b[i] > b[j]$  且  $i > j$



中国计算机学会  
China Computer Federation

# 最长公共子序列：勇闯黄金十二宫

Time Limit: 1000MS Memory Limit: 128MB

## Description

“已知艾尔里斯和弟弟艾尔里亚的基因基本相同，由于基因表达起来不方便，所以就用 $n$ 个数字来表示。(因为至今共发现100000种基因，所以每个数字都 $\leq 100000$ )兄弟之间的基因个数是相同的，就是说他们都有 $n$ 个数字。且对于每个人，这 $n$ 个数字互不相同。现在要求兄弟之间基因的最长公共部分。可以不连续。”



中国计算机学会

China Computer Federation

## Input

第1行, 为 $n(1 \leq n \leq 100000)$

下面2行, 每行 $n$ 个数字, 表示了一个人的所有基因。

## Output

一行, 为他们两人基因的最长公共部分的长度。

## Sample Input

7

1 2 3 4 5 6 7

7 6 5 4 1 2 3

## Sample Output

3



中国计算机学会

China Computer Federation

- 一般解法

- $f[i][j]$ 表示第一个人前 $i$ 个数, 第二个人前 $j$ 个数的最长公共子序列

- $$f[i][j] = \begin{cases} \max\{f[i-1][j], f[i][j-1]\} & a[i] \neq b[j] \\ f[i-1][j-1] + 1 & a[i] == b[j] \end{cases}$$

- 优化

- 由于每个基因不同, 可将第二个人离散化处理成第一个人的对应序列, 得到 $d[i]$ ; 问题转化成求 $d$ 数组的最长上升子序列
  - $f[i]$ 表示长度为 $i$ 的子序列最小的末尾值,  $len$ 表示当前最长上升子序列长度
  - 如果 $a[i] > f[len]$ ,  $f[len+1] = a[i]$ ; 否则, 二分查找第一个比 $a[i]$ 小的 $f[k]$ , 更新 $f[k+1] = a[i]$



## Description

Tom是一个非常有创业精神的人,由于大学学的是汽车制造专业,所以毕业后他用有限的资金开了一家汽车零件加工厂 专门为汽车制造商制造零件。由于资金有限他只能先购买一台加工机器。现在他却遇到了麻烦,多家汽车制造商需要他加工一些不同零件(由于厂家和零件不同,所以给的加工费也不同),而且不同厂家对于不同零件的加工时间要求不同(有些加工时间要求甚至是冲突的,但开始和结束时间相同不算冲突)。

Tom当然希望能把所有的零件都加工完,以得到更多的加工费,但当一些零件的加工时间要求有冲突时,在某个时间内他只能选择某种零件加工(因为他只有一台机器),为了赚得尽量多的加工费,Tom不知如何进行取舍。现在请你帮Tom设计一个程序,合理选择部分(或全部)零件进行加工,使得得到最大的加工费。



中国计算机学会

China Computer Federation

## Input

第一行是一个整数 $n$  ( $n \leq 30000$ ), 表示共有 $n$ 个零件须加工。

接下来的 $n$ 行中, 每行有3个整数, 分别表示每个零件加工的时间要求。第一个表示开始时间, 第二个表示该零件加工的结束时间, 第三个表示加工该零件可以得到的加工费。

注: 数据中的每个数值不会超过100000。

## Output

只有一个整数, 表示Tom可以得到的最大加工费。

## Sample Input

```
3
1 3 10
4 6 20
2 5 25
```

## Sample Output

```
30
```



中国计算机学会

China Computer Federation

- $a[i]$ 、 $b[i]$ 、 $c[i]$ 表示第 $i$ 个零件加工起始时间、结束时间和加工费
- 对 $n$ 个零件按照 $b$ 数组升序排序
- $f[i]$ 表示结束时间为 $i$ 时得到的最大加工费
- $$f[i] = \max \begin{cases} f[i-1] \\ f[a[j]] + c[j] & \text{如果 } b[j] == i \end{cases}$$