

与第 4 章衔接引入凸包，DP 的单调思想加强。

## Translate:USACO/fc

---

### Fencing the Cows 圈奶牛

译 by Felicia Crazy

### 描述

---

农夫约翰想要建造一个围栏用来围住他的奶牛，可是他资金匮乏。他建造的围栏必须包括他的奶牛喜欢吃草的所有地点。对于给出的这些地点的坐标，计算最短的能够围住这些点的围栏的长度。

### 格式

---

**PROGRAM NAME:** fc

#### INPUT FORMAT

输入数据的第一行包括一个整数  $N$ 。 $N$  ( $0 \leq N \leq 10,000$ ) 表示农夫约翰想要围住的放牧点的数目。接下来  $N$  行，每行由两个实数组成， $X_i$  和  $Y_i$ ，对应平面上的放牧点坐标 ( $-1,000,000 \leq X_i, Y_i \leq 1,000,000$ )。数字用小数表示。

#### OUTPUT FORMAT

输出必须包括一个实数，表示必须的围栏的长度。答案保留两位小数。

### SAMPLE INPUT (file fc.in)

---

```
4
4 8
4 12
5 9.3
```

7 8

## SAMPLE OUTPUT (file fc.out)

12.00

分析:

Graham 方法.

## Translate:USACO/starry

**Starry Night** 夜空繁星

(IOI 98 )

### 描述

高高的星空，簇簇闪耀的群星形态万千。一个星座(**cluster**)是一群连通的星组成的非空集合，所谓连通是指水平，垂直或者对角相邻。一个星座不能是另一个更大星座的一部分星座可以相似(**similar**)。如果两个星座有相同的形状，而且包括相同数目的星体，那么不管其方向性如何，就算相似。一般而言，星座可能的方向有八个，如图 1 所示。

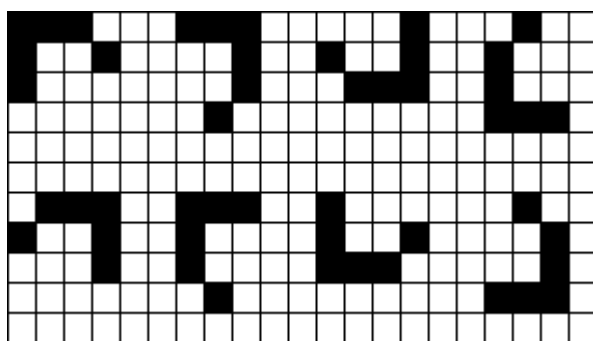


图 1 相似的八个星座

夜空可以表示为一份天体图(**sky map**)，它是一个由字符 **0** 和 **1** 组成的二维矩阵，字符 **1** 表示所在的位置有一颗星；字符 **0** 表示该位置上没有星.给定一份天体图，用同一个小写英文标识(**mark**)相似的所有星座。相似的星座必须用相同的字母标识，不同的星座表示为不同的字母。标识一个星座，就是将其各星体对应的字符 **1** 替换为相应的小写字母.

## INPUT FORMAT

---

文件的前两行分别记录了天体图的宽度 **W**、深度 **H**。而天体图则是由接下来的 **H** 行表示，每行包括 **W** 个字符.

## OUTPUT FORMAT

---

输出文件记录了天体图与文件 **STARRY.IN** 相似，不同之处在于，各个星座按照“任务”中的要求进行了标识(**mark**)。对于同一个输入文件，可能会有很多不同的标识，此时，输出字典序最小的标识。

## SAMPLE INPUT (file starry.in)

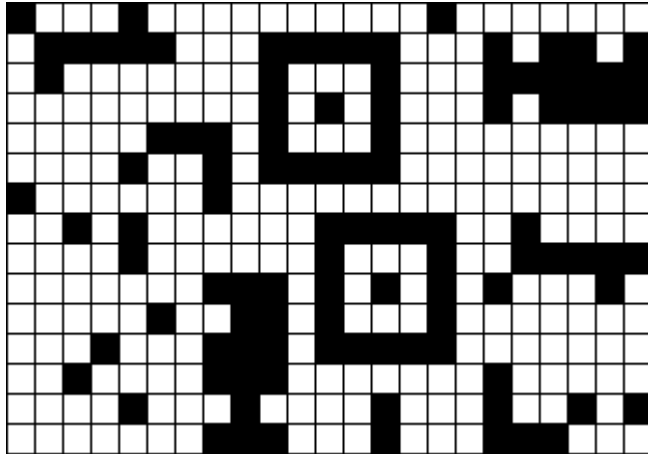
---

```
23
15
100010000000000010000000
01111100011111000101101
01000000010001000111111
00000000010101000101111
00000111010001000000000
00001001011111000000000
10000001000000000000000
00101000000111110010000
```

```

00001000000100010011111
00000001110101010100010
00000100110100010000000
00010001110111110000000
00100001110000000100000
00001000100001000100101
00000001110001000111000

```



## SAMPLE OUTPUT (file starry.out)

```

a000a0000000000b0000000
0aaaaa000cccc000d0dd0d
0a0000000c000c000ddddd
000000000c0b0c000d0ddd
00000eee0c000c000000000
0000e00e0cccc000000000
b000000e000000000000000
00b0f000000cccc00a0000
0000f000000c000c00aaaa
0000000ddd0c0b0c0a000a0
00000b00dd0c000c0000000
000g000ddd0cccc0000000
00g0000ddd0000000e00000
0000b000d0000f000e00e0b
0000000ddd000f000eee000

```

这是上述输入实例的一个可能的结果。请注意，该输出文件对应于下面的天空景象。

a				a									b						
	a	a	a	a	a			c	c	c	c	c			d		d	d	d
	a							c				c			d	d	d	d	d
								c	b		c				d		d	d	d
					e	e	e	c				c							
				e			e	c	c	c	c	c							
b							e												
		b		f					c	c	c	c	c		a				
				f					c				c		a	a	a	a	a
							d	d	d		c	b		c	a				a
					b		d	d	d		c			c					
							d	d	d		c	c	c	c	c				
			g				d	d	d										
		g					d	d	d						e				
				b			d				f				e		e		b
							d	d	d			f			e	e	e		

## Constraints

0 <= W (天体图的宽度) <= 100

0 <= H (天体图的深度) <= 100

0 <= 星座的数目 <= 500

0 <= 不相似的星座数目 <= 26 (a..z)

1 <= 各星座包含的星体数目 <= 160

分析：

Floodfill+Hash。先 Floodfill 分离出各个连通块，然后构造 8 个方向的 Hash,每个都要自左上向右下顺序扫描赋上键值，判断时还应当注意方块的长宽是否一样。加上这些判断就严谨了。速度也在 0.1s 以内。

## Translate:USACO/theme

**Musical Themes** 乐曲主题

Brian Dean

译 by Maigo Akisame

描述

我们用  $N(1 \leq N \leq 5000)$  个音符的序列来表示一首乐曲, 每个音符都是 1..88 范围内的整数, 每个数表示钢琴上的一个键。很不幸这种表示旋律的方法忽略了音符的时值, 但这项编程任务是关于音高的, 与时值无关。

许多作曲家围绕一个重复出现的“主题”来构建乐曲。在我们的乐曲表示法中, “主题”是整个音符序列的一个子串, 它需要满足如下条件:

1. 长度至少为 5 个音符
2. 在乐曲中重复出现(可能经过转调, 见下)
3. 重复出现的同一主题不能有公共部分。

“转调”的意思是主题序列中每个音符都被加上或减去了同一个整数值。给定一段乐曲, 计算其中最长主题的长度(即音符数)。

本题时限为 1 秒钟!

## 格式

---

**PROGRAM NAME:** theme

### INPUT FORMAT

输入文件的第一行包含整数  $N$ 。下面的每一行(最后一行可能除外)包含 20 个整数, 表示音符序列。最后一行可能少于 20 个音符。

### OUTPUT FORMAT

输出文件应只含一个整数, 即最长主题的长度。如果乐曲中没有主题, 那么输出 0。

## SAMPLE INPUT (file theme.in)

---

```
30
25 27 30 34 39 45 52 60 69 79 69 60 52 45 39 34 30 26 22 18
82 78 74 70 66 67 64 60 65 80
```

## SAMPLE OUTPUT (file theme.out)

```
5
```

(这个长度为 5 的主题是输入文件中第一行的最后 5 个音符和第二行开头 5 个音符)

分析：

本题目使用动态规划解决。用  $dp[i][j]$  表示以  $i, j$  开头的可以匹配的最大串长。考虑如果  $num[i]$  与  $num[j]$  匹配那么，以这两个数字开头上的可以匹配的最长串就是  $dp[i][j]=dp[i+1][j+1]+1$ ；如果不匹配则  $dp[i][j]=1$ ；

优化成一维的可以从后向前逆推， $dp[j]$  表示最长串长度。枚举方向分别为：模式串由后向前，匹配串由前向后，模式串由后向前是因为枚举顺序决定的。匹配串是因为  $dp$  的无后效性，避免覆盖与背包的一维实现类似。