

1. 编程：断开链条

给定一个包含N个整数的数组A，假设这个数组表示的是一条链条，其中每个元素表示链条上的一环。现在我们把这条链条断成3小部分。我们需要做的就是断开不相连的两个环，更确切的说，我们需要断开环P、Q ($0 < P < Q < N-1$, $Q-P > 1$)，得到三个小链条 $[0, P-1]$, $[P+1, Q-1]$, $[Q+1, N-1]$ 。上述操作的成本是 $A[P]+A[Q]$ 。

例如：数组A为

$A[0] = 5$

$A[1] = 2$

$A[2] = 4$

$A[3] = 6$

$A[4] = 3$

$A[5] = 7$

我们可以选择如下方式断开链条

- (1, 3): 成本是 $2+6 = 8$
- (1, 4): 成本是 $2+3 = 5$
- (2, 4): 成本是 $4+3 = 7$

写一个函数

```
int breakchain(int A[], int N);
```

对任一个给定的链条，返回断开链条的最低成本，比如上面的例子中，需要返回最低成本5

假设：

- N是整数，范围 $[5, 100000]$

A的元素是范围在 $[1, 1000000000]$ 的整数

2. 编程：M个最大的数

编程实现从N个无序数中选取M个最大的数 ($0 < M < N$)

3. 编程：最大方形

给定一个元素是布尔型大小为 $N \times M$ 的矩阵A。大小为L的方形可以放置在A中的 (X, Y) 处，如果满足：

$$0 < L \leq \min(N, M)$$

$$0 \leq X \leq N-L$$

$$0 \leq Y \leq M-L$$

$$A[X+i][Y+j] = \text{true}, 0 \leq i < L, 0 \leq j < L$$

如果L能被放置在 (X, Y) 以及 $(X+1, Y)$ 或者 $(X, Y+1)$ ，我们说它可以被从 (X, Y) 移动到 $(X+1, Y)$ 或者 $(X, Y+1)$ 。

我们想找出满足以下条件的最大的L：

L可以被放置在 $(0, 0)$

它可以被经过一系列的移动，从 $(0, 0)$ 到达 $(N-L, M-L)$

换句话说，我们想找到最大的方形，可以从左上移动到右下，移动方式是向下或者向右移动。一个极端情况是如果矩阵A中的元素全部是true，那么 $L = \min(N, M)$ 。

写一个函数：`int move_square(int **A, int N, int M);`

返回值为L。

比如：

```
A[0][0]=true  A[0][1]=true  A[0][2]=true  A[0][3]=false
A[1][0]=true  A[1][1]=true  A[1][2]=true  A[1][3]=false
A[2][0]=true  A[2][1]=true  A[2][2]=true  A[2][3]=false
A[3][0]=true  A[3][1]=true  A[3][2]=true  A[3][3]=true
A[4][0]=false A[4][1]=true  A[4][2]=true  A[4][3]=true
A[5][0]=true  A[5][1]=false A[5][2]=true  A[5][3]=true
返回 2
```

```
A[0][0]=true  A[0][1]=true  A[0][2]=false A[0][3]=false
A[1][0]=true  A[1][1]=false A[1][2]=false A[1][3]=false
A[2][0]=false A[2][1]=true  A[2][2]=false A[2][3]=true
返回 0
```

```
A[0][0]=true
返回 1
```

假设：

N和M是范围为[1, 200]的整数

4. 编程：顺时针环绕列印矩阵元素

给定一个整数元素组成的矩阵，写一个函数，返回一个数组，数组中的元素是按照顺时针遍历矩阵中的元素而组成。例如如下的3x4矩阵：

2, 3, 4, 8

5, 7, 9, 12

1, 0, 6, 10

得到的数组的元素按照顺序是“2, 3, 4, 8, 12, 10, 6, 0, 1, 5, 7, 9”。

5. 问答题（只要给出答案和解释就行，是否通过编程得到答案自己决定）
假设你英姿飒爽的站在120层摩天大楼的天台，手握两个iphone（就是这么豪!），并且可轻易到达任一楼层的阳台。请用最少的尝试次数，确定能够让iphone自由下落而不会摔坏的最高层数。比如20层摔不坏，21层就摔坏了。在尝试中你可以摔坏这两个iphone，只要能得到答案。请说明最坏情况需要多少次尝试以及你的思路。