

## Đi trên vòng tròn

Trên một vòng có đánh dấu  $N$  vị trí, các vị trí được đánh số từ 0 tới  $N - 1$  theo chiều kim đồng hồ.

Bạn hiện đang ở một trong  $N$  vị trí (bạn không biết vị trí cụ thể của mình) và nhiệm vụ của bạn là đi đến được vị trí 0. Bạn chỉ được phép đi theo chiều kim đồng hồ, sau một bước đi thì từ vị trí  $i$  bạn sẽ đi đến vị trí  $(i + 1) \bmod N$ . Bạn có một chiếc la bàn ma thuật, mỗi khi kích hoạt, nó sẽ cho bạn biết liệu bạn có thể đi đến vị trí 0 bằng cách đi theo chiều kim đồng hồ không quá  $N / 2$  bước nữa hay không.

Chiếc la bàn sắp sửa hết pin và bạn chỉ có thể kích hoạt nó tối đa 35 lần nữa thôi.

### Cài đặt:

Hãy viết một chương trình tương tác với chương trình chấm trong đó cài đặt hàm:

`void escape(int N)`

- $N$  ( $2 \leq N \leq 10^9$ ) là số lượng vị trí trên vòng tròn.
- Sau khi hàm này kết thúc, bạn phải đang ở vị trí 0.
- Chương trình chấm sẽ gọi hàm này đúng một lần.

Hàm `escape` có thể gọi hàm sau:

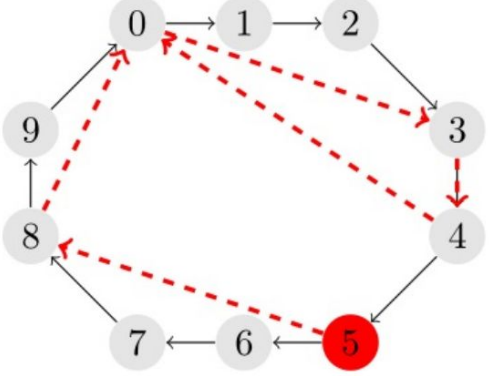
`bool jump(int x)`

- $x$  ( $0 \leq x \leq N$ ) là số lượng bước bạn sẽ đi trước khi kích hoạt la bàn ma thuật.
- Hàm này sẽ trả về `true` nếu sau khi đi  $x$  bước theo chiều kim đồng hồ, bạn còn cách vị trí 0 không quá  $N / 2$  bước nữa. Ngược lại, hàm này sẽ trả về `false`.
- Hàm này được phép gọi không quá 35 lần.

Để gọi được hàm `jump` trên, chương trình của bạn phải khai báo thư viện `cycle.h`

`#include "cycle.h"`

Ví dụ  $N = 10$  và ban đầu bạn đang ở vị trí 5, chương trình chấm sẽ gọi `escape(10)`. Bạn có thể tương tác như sau:

	Giá trị trả về	Vị trí hiện tại	
<code>escape(10)</code>		5	
<code>jump(3)</code>	true	8	
<code>jump(2)</code>	true	0	
<code>jump(3)</code>	false	3	
<code>jump(1)</code>	false	4	
<code>jump(6)</code>	true	0	

### Chấm điểm:

Với mỗi test, bạn sẽ nhận được điểm của test đó nếu các lời gọi hàm là hợp lệ và sau khi kết thúc bạn phải đang ở vị trí 0.

**Lưu ý: Chương trình của bạn không được viết hàm main()**

**Chương trình chấm mẫu:**

Để tiện cho việc kiểm thử, bạn được cung cấp chương trình chấm mẫu *grader\_sample.cpp*, file *cycle.h* và một file bài làm mẫu *cycle\_sample.cpp*. Để thuận tiện cho việc biên dịch, nên đặt 3 file này cùng một thư mục.

Chương trình chấm mẫu *grader\_sample.cpp* đọc dữ liệu đầu vào theo định dạng:

- Dòng đầu:  $N$   $p$  ( $p$  là vị trí ban đầu,  $1 \leq p < N$ )

Nếu chương trình của bạn đúng, chương trình chấm mẫu sẽ in ra OK và số lượng lần gọi hàm *jump*. Nếu chương trình của bạn trả lời không chính xác, trình chấm mẫu sẽ in ra lỗi của bạn.