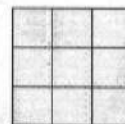


Bài 1: ĐẾM HÌNH VUÔNG (25 điểm)

Cho một bảng hình vuông kích thước $n \times n$ được chia thành lưới ô vuông đơn vị, các vị trí đỉnh của các ô vuông đơn vị được gọi là các mắt lưới. Người ta muốn đếm số lượng những hình vuông thỏa mãn hai điều kiện sau:



- Mỗi cạnh hình vuông phải song song với một trong hai cạnh bảng
- Cả 4 đỉnh của hình vuông phải nằm tại vị trí của các mắt lưới

Ví dụ với bảng kích thước 3×3 ta có thể đếm được 14 hình vuông thỏa mãn hai điều kiện trên.

Vì số lượng hình vuông đếm được có thể rất lớn nên em chỉ cần cho biết kết quả là số dư của phép chia: số lượng hình vuông đếm được chia cho 2013.

Em cần tạo file kết quả có tên là SQUARES.TXT gồm 10 dòng, mỗi dòng ghi một số nguyên duy nhất là kết quả tìm được (số dư khi chia số lượng hình vuông đếm được cho 2013) ứng một giá trị n dưới đây.

Dòng	n	SQUARES.TXT
1	2	14
2	3	
3	4	
4	5	
5	6	
6	666	
7	7777	
8	88888	
9	99999999	
10	123456789123456789	

Chú ý: Kết quả tương ứng với giá trị n nào cần ghi ĐÚNG trên dòng tương ứng với giá trị n đó

Bài 2. THÁP LŨY THỪA (30 điểm) - Tên file nộp TETRATION.TXT

Tháp lũy thừa (power tower) hay lũy thừa lặp (iterated power) là một phép toán thường được sử dụng để biểu diễn những giá trị rất lớn. Với hai số nguyên a và n ($a > 0, n \geq 0$), tháp lũy thừa bậc n của a (ký hiệu $a \uparrow\uparrow n$) định nghĩa như sau:

$$a \uparrow\uparrow n = \begin{cases} 1, & \text{nếu } n = 0 \\ a^{a \uparrow\uparrow (n-1)}, & \text{nếu } n > 0 \end{cases} = \underset{n \text{ cấp}}{a^{(a^a)}}$$

Ví dụ

$$2 \uparrow\uparrow 1 = 2$$

$$2 \uparrow\uparrow 2 = 2^2 = 4$$

$$2 \uparrow\uparrow 3 = 2^{(2^2)} = 2^4 = 16$$

$$2 \uparrow\uparrow 4 = 2^{(2^{(2^2)})} = 2^{16} = 65536$$

$$3 \uparrow\uparrow 3 = 3^{(3^3)} = 3^{27} = 7625597484987$$

Yêu cầu: Cho ba số nguyên dương a, n, m , hãy cho biết số dư trong phép chia $(a \uparrow n)$ cho m .

Em cần tạo file kết quả có tên là TETRATION.TXT gồm 15 dòng, mỗi dòng ghi một số nguyên duy nhất là kết quả tìm được ứng một bộ giá trị a, n, m dưới đây.

Dòng	a	n	m	TETRATION.TXT
1	2	4	100	36
2	11	2	100	
3	2222	2	1000	
4	1000000	2	111222	
5	7	3	1000000	
6	8	3	1000000	
7	9	3	456789	
8	6	100	108	
9	8	8	9	
10	11	100	12	
11	3	4	1000000	
12	15	1000000	3087	
13	1000000	1000000	123456	
14	24680	999999	999983	
15	123456	456789	142857	

Chú ý: Kết quả tương ứng bộ dữ liệu nào cần ghi ĐÚNG trên dòng tương ứng với bộ dữ liệu đó

Bài 3. PHÁO ĐÀI CỔ (45 điểm)

Trong những ghi chép cổ xưa có những thông tin về một pháo đài cổ ở thành Vinh. Trên bản đồ phẳng với hệ tọa độ Descartes vuông góc Oxy. Nền pháo đài có dạng hình chữ nhật với các cạnh song song với một trong hai trục tọa độ. Những khảo sát gần đây tại vị trí (x_0, y_0) đã khẳng định chắc chắn rằng vị trí (x_0, y_0) nằm trong pháo đài cổ xưa.

Em được yêu cầu xác định vị trí ban đầu của nền pháo đài cổ bằng một robot hoạt động theo quy trình sau. Tại mỗi bước, em yêu cầu robot đào sâu xuống khảo sát tại một vị trí do em lựa chọn. Robot sẽ báo cáo có tìm thấy dấu vết của pháo đài cổ tại vị trí đó hay không? Vì chi phí của mỗi lần khảo sát là khá lớn nên em cần yêu cầu robot khảo sát tại càng ít vị trí càng tốt.

Nhiệm vụ của em là viết một chương trình Pascal có tên CITADEL.PAS, sử dụng các hàm được cung cấp trong thư viện ROBOT.PP để thực hiện việc khảo sát các vị trí, nhận kết quả khảo sát và đưa ra đánh giá về vị trí chính xác của khu pháo đài cổ. Chương trình CITADEL.PAS của em phải khai báo sử dụng thư viện ROBOT.PP bằng cú pháp:

uses robot;

Các hàm và thủ tục được cung cấp trong thư viện ROBOT.PP:

function getX0: LongInt;

hàm trả về giá trị x_0

function getY0: LongInt;

hàm trả về giá trị y_0

function find(const x, y: LongInt): LongInt;

Em chỉ được phép dùng hàm này tối đa 1 triệu lần. Hàm trả về giá trị 0 hoặc 1:

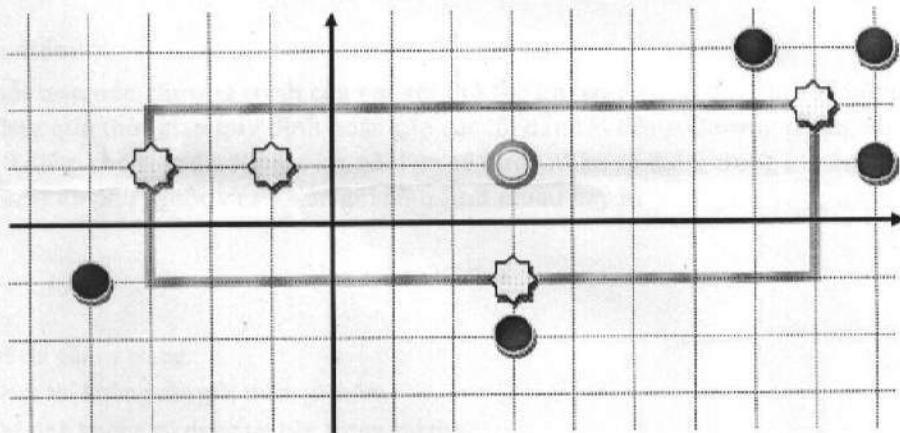
- 0 tương ứng với **không** tìm thấy dấu vết tại vị trí có tọa độ (x, y) . Nói cách khác là vị trí (x, y) nằm ở phía ngoài nền pháo đài cổ.
- 1 tương ứng với **có** tìm thấy dấu vết tại vị trí có tọa độ (x, y) . Nói cách khác là vị trí (x, y) nằm ở phía trong hoặc trên biên nền của pháo đài cổ.
- Nếu hàm này được gọi quá 1 triệu lần, thư viện sẽ tự động ngắt chương trình và ghi nhận em bị 0 điểm với test đang chạy.

procedure answer(const x1, y1, x2, y2: LongInt);

Thủ tục answer được gọi đúng một lần trước khi kết thúc chương trình để cho biết về kết quả mà em xác định được; trong đó (x_1, y_1) là tọa độ của góc trái dưới của nền pháo đài, (x_2, y_2) là tọa độ của góc phải trên của nền pháo đài.

Chương trình bắt buộc phải gọi thủ tục answer một lần duy nhất, nếu không sẽ bị 0 điểm. Thủ tục answer của ban giám khảo khi chấm bài sẽ tự động thoát chương trình.

Ví dụ:



Các thủ tục được gọi	Giá trị thư viện ROBOT.PP trả về
getX0	3
getY0	1
find(3,-1)	1
find(3,-2)	0
find(-4,-1)	0
find(-1,1)	1
find(-3,1)	1
find(8,2)	1
find(9,3)	0
find(9,1)	0
find(7,3)	0
answer(-3,-1,8,2)	Ghi nhận kết quả và kết thúc chương trình