

## Recurrences

จงเขียนฟังก์ชันตามนิยามที่กำหนดให้ต่อไปนี้

Padovan sequence	<b>def p(n) :</b>	$p_n = \begin{cases} 1 & \text{if } 0 \leq n \leq 2 \\ p_{n-2} + p_{n-3} & \text{if } n \geq 3 \end{cases}$
Mallow's sequence	<b>def m(n) :</b>	$m(n) = m(m(n-2)) + m(n - m(n-2))$ with $m(1) = m(2) = 1$
Segner's recurrence	<b>def e(n) :</b>	$e_n = \sum_{k=1}^{n-2} e_{k+1} e_{n-k} \text{ if } n > 3, \quad e_2 = e_3 = 1$
Sylvester's sequence	<b>def s(n,k) :</b>	$s(n,k) = \begin{cases} 2 & \text{if } n = 0 \\ (s(n-1,k)^2 - s(n-1,k) + 1) \bmod k & \text{if } n > 0 \end{cases}$

จงเขียนฟังก์ชันเพื่อคำนวณค่าต่าง ๆ ที่นิยามในตารางข้างบนนี้ ด้วยโครงของโปรแกรมข้างล่างนี้

<pre>def p(n) :     ???  def m(n) :     ???  def e(n) :     ???  def s(n,k) :     ???  exec(input().strip())    # do not remove this line!!</pre>	<p>ข้อควรระวัง : กรณีทดสอบของ Grader สำหรับฟังก์ชัน <b>m</b> และ <b>s</b> อาจทำให้ผลการตรวจเป็น T (คือใช้เวลามากเกินไป) จึงควรหลีกเลี่ยงการเรียกฟังก์ชันตัวเองซ้ำมากเกินไป</p>
---	--

## ข้อมูลนำเข้า

คำสั่งในการทดสอบฟังก์ชันที่เขียน

## ข้อมูลส่งออก

ผลลัพธ์ที่ได้จากคำสั่งที่ป้อนเป็นข้อมูลนำเข้า

## ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
<code>print(p(3))</code>	2
<code>print(p(20))</code>	200
<code>print(m(5))</code>	3
<code>print(m(50))</code>	35
<code>print(e(7))</code>	42
<code>print(e(10))</code>	1430
<code>print(s(4,10))</code>	7
<code>print(s(100,23))</code>	19