

เลขส่วนคอมพิวเตอร์ปี 2547

ข้อ 1 พิจารณาส่วนของโปรแกรม

```
Module question (m,n : integer);  
  if n < 0 then  
    a := -n // ไม่ทำเพราะ n เท่ากับ 6  
  else // ทำกรณีนี้แทน  
    a := n // ดังนั้นได้ a = 6  
  k := 0  
  x := 0  
  ...
```

จบการทำงานส่วนนี้ ได้ $a=6, k=0, x=0, m=0, n=6$

```
  ...  
  while k < a  
  begin  
    x := x + m  
    k := k + 1  
  end  
  ...
```

ดังนั้นโปรแกรมทำงานในวงวนนี้ ที่ค่า $k=0,1,2,3,4,5$

ดังนั้นคำสั่ง $x := x + m$ จะทำทั้งหมด 6 ครั้ง

ได้ $a=6, k=6, x=0, m=0, n=6$

```
  ...  
  if n < 0 then  
    question := -x  
  else  
    question := x
```

ดังนั้น $question = 0$

ตอบข้อ ก.

ข้อ 2 ในทำนองเดียวกัน ก่อนเข้าวงวน while เราได้ค่าต่างๆดังนี้

$a=6, k=0, x=0, m=6, n=6$

```
  ...  
  while k < a  
  begin  
    x := x + m  
    k := k + 1  
  end  
  ...
```

คำสั่ง $x := x + m$ จะทำงานทั้งหมด 6 ครั้ง

ดังนั้น $question = 6 * 6 = 36$

ตอบข้อ ค.

ข้อ 3 ฟังก์ชันการทำงานในส่วนที่เป็นวงวน while ทำงานเหมือนกับการคูณ $|n|$ และ m แล้วนำผลเก็บไว้ในตัวแปร x และมีการพิจารณาเครื่องหมายของคำตอบจากเงื่อนไขต่อจากวงวน while ดังนั้นฟังก์ชันนี้ จึงเป็นการหาผลคูณ m และ n

ตอบข้อ ก.

ข้อ 5 ให้ $a = -91$ และ $b = 287$

```
Module question2 (a, b : integer);
    x := a
    y := b
    while y <> 0
    begin
        r := x mod y
        x := y
    y := r
    end
    question2 = x
```

พิจารณาค่าของตัวแปรแต่ละตัวในระหว่างการทำงานดังต่อไปนี้

รอบการทำงาน	x	y	r
ก่อนเข้าวงวน	-91	287	-
รอบที่ 1	287	196	196
รอบที่ 2	196	91	91
รอบที่ 3	91	14	14
รอบที่ 4	14	7	7
รอบที่ 5	7	0	0

โปรแกรมจบการทำงานหลังทำงานรอบที่ 5 เนื่องจาก y มีค่าเท่ากับ 0 ซึ่งทำให้เงื่อนไขในการทำงานในวงวนไม่เป็นจริง และได้ $question2 = x = 7$

หากผู้อ่านเคยศึกษาเรื่องการหาตัวหารร่วมมาก (ห.ร.ม.) ด้วยวิธีการของยุคลิด ผู้อ่านจะเห็นได้ชัดว่า ฟังก์ชันนี้เป็นการหา ห.ร.ม. ด้วยวิธีดังกล่าว แต่หากผู้อ่านไม่เคยศึกษาเรื่องนี้มาก่อน ก็อาจลองแทนค่า a และ b ต่างๆกัน และสังเกตผลจากการทำงาน ก็จะทราบได้เช่นเดียวกันว่า ผลจากการทำงานของฟังก์ชันนี้ เป็นการหา ห.ร.ม

ตอบข้อ ค.

ข้อ 6 เนื่องจากเราทราบแล้วว่า ฟังก์ชันข้างต้นเป็นการหา ห.ร.ม. ของ a และ b

ดังนั้นคำตอบข้อนี้จึงเป็น ห.ร.ม. ของ 252 และ $198 = 18$

ข้อ 7 ดังที่ได้อธิบายไว้แล้ว โปรแกรมนี้เป็นการคำนวณหาตัวหารร่วมมาก ด้วยวิธีการของยุคลิด

ข้อ 8 พิจารณาข้อความความยาว n

กรณีที่ 1 ข้อความขึ้นต้นด้วย 1 หรือ 2

ดังนั้น จำนวนข้อความความยาว n ที่มี 0 ติดกันสองตัว $= 2 * \text{จำนวนข้อความความยาว } n-1 \text{ ที่มี 0 ติดกันสองตัวในกรณีนี้}$ (เพราะเลือกตัวเริ่มต้นได้ 2 แบบ คือ 1 หรือ 2)

กรณีที่ 2 ข้อความขึ้นต้นด้วย 0

ดังนั้น จำนวนข้อความความยาว n ที่มี 0 ติดกันสองตัว แยกได้อีกเป็นสองกรณีคือ

กรณีที่ 2.1 ขึ้นต้นด้วย 00 มีทั้งหมด $3n-2$

กรณีที่ 2.2 ขึ้นต้นด้วย 01 หรือ 02 มีทั้ง $2 * \text{จำนวนข้อความความยาว } n-2 \text{ ที่มี 0 ติดกันสอง}$

ตัว

กำหนดให้ $f(x)$ แทน จำนวนข้อความที่มี 0 สองตัวติดกันสำหรับ Ternary String ความยาว x

เขียนเป็นความสัมพันธ์เวียนเกิด ได้ $f(n) = f(n-1) + f(n-2) + 3^{n-2}$

$f(1) = 0$ เพราะไม่มีข้อความใดๆเลยที่มี 0 สองตัวติดกัน

$f(2) = 1$ เพราะมีกรณีเดียว คือ 00

$f(3) = 2*f(2) + 2*f(1) + 3^1 = 0 + 2 + 3 = 5$

$f(4) = 2*f(3) + 2*f(2) + 3^2 = 10 + 2 + 9 = 21$

ดังนั้น มีทั้งหมด 21 แบบ

ตอบข้อ ง.

ข้อ 9 พิจารณาค่าของ x หลังแต่ละรอบการทำงานในวงวน for

รอบการทำงาน	x	i
ก่อนทำงาน	1	-
รอบที่ 1	1	1
รอบที่ 2	$1*2$	2
รอบที่ 3	$1*2*3$	3
รอบ 4	$1*2*3*4$	4
...
รอบ n	$1*2*3*4*...*n$	n

เป็นการทำงานเพื่อหาค่า $n!$

ข้อ 10 พิจารณาแยกแต่ละกรณีตามนี้

```
I)
Fact = n * Fact(n-1)
```

กรณีนี้ จะเห็นได้ว่า การหาค่าของ $\text{Fact}(n)$ จะมีการเรียกใช้ $\text{Fact}(n-1)$ และ $\text{Fact}(n-1)$ ก็จะเรียกใช้ $\text{Fact}(n-2)$ ไปเรื่อยๆ ไม่สิ้นสุด ดังนั้น โปรแกรมทำงานไม่ได้ในกรณีนี้

```

II)
if n <= 2 then
    Fact = 2
else
    Fact = n * Fact(n-1)

```

กรณีนี้สามารถทำงานได้ถูกต้องทุก $n \geq 2$ พิจารณาได้ดังนี้

$$\text{Fact}(2) = 2 = 2!$$

$$\text{Fact}(3) = 3 * \text{Fact}(2) = 3 * 2 = 3!$$

$$\text{Fact}(4) = 4 * \text{Fact}(3) = 4 * 3! = 4!$$

...

$$\text{Fact}(n) = n * \text{Fact}(n-1) = n * (n-1)! = n!$$

ดังนั้น กรณี สามารถทำงานได้ถูกต้องทุก $n \geq 2$

```

III)
if n = 1 then
    Fact = 1
else
    Fact = Fact(n+1) / (n+1)

```

จะเห็นได้ว่า $\text{Fact}(n)$ จะเรียก $\text{Fact}(n+1)$ และ $\text{Fact}(n+1)$ ก็จะเรียก $\text{Fact}(n+2)$ ไปเรื่อยๆ และทำให้ฟังก์ชันนี้ทำงานไม่สิ้นสุด

ดังนั้น ฟังก์ชันที่ถูกต้องที่สุดควรเป็นฟังก์ชันที่ 2

ตอบข้อ ข.

ข้อ 11 โจทย์กำหนดให้

```

T(0) = 1
T(1) = 3
T(N) = 2 * T(N-1) - T(N-2) เมื่อ N > 1

```

ให้ $G(N)$ แทน จำนวนครั้งที่ $T(2)$ ถูกเรียกในการเรียก $T(N)$

$$\text{จากความสัมพันธ์ } T(N) = 2 * T(N-1) - T(N-2)$$

เราจะได้ความสัมพันธ์ $G(N) = G(N-1) + G(N-2)$ ด้วย เพราะการเรียกใช้ $T(N-1)$ จะเรียก $T(2)$ เป็นจำนวน $G(N-1)$ ครั้ง และการเรียก $T(N-2)$ จะเรียกใช้ $T(2)$ เป็นจำนวน $G(N-2)$ ครั้ง

$$\text{ดังนั้น } G(N) = G(N-1) + G(N-2)$$

$$\text{พบว่า } T(3) = 2 * T(2) - T(1) \text{ ดังนั้น } G(3) = 1$$

$$\text{และ } T(4) = 2 * T(3) - T(2) \text{ ดังนั้น } G(4) = 1 + 1 = 2$$

$$G(5) = G(4) + G(3) = 2 + 1 = 3$$

$$G(6) = G(5) + G(4) = 3 + 2 = 5$$

ดังนั้น ในการหา $T(6)$ จะมีการเรียกใช้ $T(2)$ ทั้งหมด 5 ครั้ง

ตอบข้อ ข.