

ซูโดโค้ด (Pseudocode) หรือ ในภาษาไทย เรียกว่า รหัสจำลอง หรือ รหัสเทียม คือ การจำลองการเขียนโปรแกรม โดยเขียนคำสั่งเป็นภาษาอะไรก็ได้ เช่น ไทย หรือ อังกฤษ ฯลฯ ที่ไม่ใช่ภาษาคอมพิวเตอร์ขั้นสูงอย่าง ภาษา C หรือ Java ซูโดโค้ดนี้จะไม่ขึ้นกับภาษาคอมพิวเตอร์ใดๆ โดยเฉพาะ ใช้อธิบายการทำงานของโปรแกรม เนื่องจากสื่อด้วยภาษาง่ายๆ อย่าง ไทย, อังกฤษ ทำให้ผู้อ่านเข้าใจได้ง่าย และตรงกัน โดยปกติแล้วมักใช้ภาษาอังกฤษ เนื่องจากสามารถแปลเป็นภาษาคอมพิวเตอร์ได้ง่าย

หัวใจหลักของการเขียน Pseudocode ==> สื่อให้ผู้อ่านเข้าใจได้ง่ายที่สุด

หลักในการเขียน Pseudocode

1. คำสั่งที่เขียนใช้ภาษาที่เข้าใจง่าย ไม่ต้องคำนึงถึงภาษาคอมพิวเตอร์
2. ในหนึ่งบรรทัด ให้มีเพียงหนึ่งคำสั่งเท่านั้น
3. ใช้ย่อหน้าในการแสดงกลุ่มของคำสั่ง ที่เป็นคำสั่งย่อยในคำสั่งพวกเงื่อนไข เช่น if-else, while, for ฯลฯ
4. ในการเขียนแต่ละคำสั่งให้เรียงการทำงานจากบนลงล่าง และมีทางออกหรือจุดสิ้นสุดเพียงจุดเดียว
5. กลุ่มคำสั่งอาจจะเขียนรวมกันเป็นโมดูล และเวลาเรียกใช้ก็เรียกใช้ผ่านชื่อโมดูล (คล้ายกับการเรียกใช้ฟังก์ชัน)

ต่อไปจะเป็นตัวอย่างโจทย์ปัญหา พร้อมกับการเขียน Pseudocode จากปัญหาเหล่านี้ครับ ...

ตัวอย่างที่ 1 จงเขียน Pseudocode จากโปรแกรมหาผลรวมของตัวเลข 2 ค่า

แล้วแสดงผลรวมออกมาทางหน้าจอ

1. read x , y
2. calculate sum = x + y
3. print sum

ตัวอย่างที่ 2 จงเขียน Pseudocode จากโปรแกรมรับตัวเลข 1 ค่าแล้วตรวจสอบว่าเป็นเลขคู่หรือเลขคี่

1. read x
2. if $x \% 2 = 0$ <- ตรวจสอบว่าเศษจากการหาร x ด้วย 2 เท่ากับ 0 หรือไม่
 1. then
 1. print "even"
 2. else
 1. print "odd"

ตัวอย่างที่ 3 จงเขียน Pseudocode จากโปรแกรมหาค่าเฉลี่ยของตัวเลข 10 ค่า

แล้วแสดงค่าเฉลี่ยออกมาทางหน้าจอ

1. for i = 1 to 10 <- เป็นคำสั่งวนรอบจำนวน 10 รอบ
 1. read num
 2. calculate sum = sum + num
2. calculate mean = sum / 10
3. print mean

ตัวอย่างที่ 4 จงเขียน Pseudocode จากโปรแกรมหาค่าเฉลี่ยของตัวเลข n ค่า

แล้วแสดงค่าเฉลี่ยออกมาทางหน้าจอ

1. read n
2. for i = 1 to n <- เป็นคำสั่งวนรอบจำนวน n รอบ
 1. read num
 2. calculate sum = sum + num
3. calculate mean = sum / n

4. print mean

ตัวอย่างที่ 5 จงเขียน Pseudocode จากโปรแกรมเปรียบเทียบตัวเลข 2 ค่า แล้วแสดงความสัมพันธ์ออกมา

1. read x , y

2. if x > y

1. then

1. print "x > y"

2. else if x < y

1. then

1. print "x < y"

2. else

1. print "x = y"

ตัวอย่าง จงเขียนรหัสจำลอง (pseudocode)

ของโปรแกรมที่ใช้หาระดับคะแนนจากคะแนนที่รับเข้ามาแล้วรายงานผลคะแนนและระดับคะแนน

โดยมีข้อความสื่อสารในกรณีที่คะแนนมากกว่า 100 หรือต่ำกว่า 0 ด้วย

จากโจทย์ อาจเขียน รหัสจำลองแสดงขั้นตอนการทำงานของโปรแกรม ได้ทำนองนี้

Input score

If score > 100

Output "impossible"

Else if score >= 80

Output score , grade = 4

Else if score >= 75

Output score , grade = 3.5

Else if score >= 70

Output score , grade = 3

Else if score >= 65

Output score , grade = 2.5

Else if score >= 60

Output score , grade = 2

Else if score >= 55

Output score , grade = 1.5

Else if score >= 50

Output score , grade = 1

Else if score >= 0

Output score , grade = 0

Else

Output too low

ตัวอย่าง จงเขียนรหัสจำลอง (pseudocode)

ของโปรแกรมที่เมื่อโปรแกรมทำงานจะมีการให้ป้อนจำนวนแถวและสดมภ์ของข้อมูลที่ต้องการ
และให้ป้อนข้อมูลของแต่ละสดมภ์และแถวเมื่อป้อนเรียบร้อยแล้วมีการแสดงข้อมูลลักษณะดังนี้

25	12	36	99
48	77	102	26
53	64	22	31

pseudocode อาจเป็นทำนองนี้

```
input num_row
input num_col
for (i ← 1; i ≤ num_row ; i++)
    for(j←1; j≤ num_col ; j++)
        input num
    end for
end for
for (i ← 1 ; i≤ num_row ; i++)
    for(j ← 1; j ≤ num_col ; j++)
        output num
    end for
end for
```

ตัวอย่าง จงเขียนรหัสจำลอง (pseudocode)

ของโปรแกรมที่เมื่อโปรแกรมทำงานจะมีการเล่นเกมทายเลขที่ผู้เล่นคนแรกกำหนดจากเลข 0 ถึง 9 ให้ผู้เล่นคนที่ 2 ทาย

โดยถ้าทายผิดจะมีข้อความขึ้นมามากกว่าเลขที่ทายน้อยกว่าหรือมากกว่าคำตอบที่กำหนดไว้

ถ้าทายถูกเกมจะยุติและขึ้นข้อความบอกว่าถูกต้องในการทายครั้งที่เท่าใด โดยให้ทายได้ไม่เกิน 4 ครั้ง ถ้าทายครบ 4 ครั้งแล้วยังทายไม่ถูกต้องให้ขึ้นข้อความมาตำหนิ

pseudocode อาจเป็นทำนองนี้

```
input assign_num
```

```
predict_no ← 1
```

```
while (predict_no <=4) do
```

```
    input predict_num
```

```
    if ( predict_num == assign_num)
```

```
        output good answer in number of prediction  predict_no
```

```
        exit while
```

```
    end if
```

```
    if (predict_num < assign_num)
```

```
        output bad answer  predictct_num < assign_num
```

```
    else
```

```
        output bad answer  predictct_num > assign_num
```

```
    end if
```

```
    if (predict_no == 4)
```

```
        output stupid
```

```
    end if
```

```
    predict_no ← predict_no+1
```

```
end while
```