

# CS381: Numerical Computation & Softwares

First Julia program

Jakramate Bootkrajang

Department of Computer Science

Chiang Mai University

Rev.2 – 2019

# Today's task

- วันนี้เราจะมาลองเขียนโปรแกรม Julia เพื่อประมาณค่า pi
- โดยใช้สมการดังต่อไปนี้ซึ่งคิดค้นโดยนักคณิตศาสตร์ชาวอินเดีย Srinivasa Ramanujan

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}} \quad (1)$$

# The steps

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

- ต้องคำนวณพจน์ที่เป็นอนุกรม
- ต้องหาค่า factorial (ใน Julia มีฟังก์ชันนี้ให้)

# Computing the series

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

- เป็นลักษณะทำซ้ำที่มีจำนวนรอบ (จำกัดหรือไม่จำกัด ?)
- นึกถึง for loop

# Computing the series

- ในทางปฏิบัติเราไม่สามารถคำนวณอนุกรมถึงพจน์อนันต์ได้
- เราอาจกำหนดค่า  $k$  เป็นค่าที่เหมาะสมไม่มากหรือน้อยเกินไปเช่น 3

```
for k in 1:3
    sum += factorial(4k) * ...
end
```

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

- เมื่อได้ค่าของอนุกรมแล้ว
- เราจะนำมาคูณกับค่าคงที่ด้านหน้าจะได้ส่วนกลับของค่า pi
- แล้วค่า pi มีค่าเท่าไร ?

# What should $k$ be ?

- 3 ?
- 5 ?
- 10 ?
- 100 ?
- เราจะตอบคำถามนี้ได้เราต้องลองเทียบค่า  $\pi$  ซึ่งเป็นที่ยอมรับและถูกนิยามไว้ใน Julia
- (HINT) เราสามารถพิมพ์สัญลักษณ์กรีกที่ใช้ในคณิตศาสตร์บ่อยๆได้โดยพิมพ์ backslash ตามด้วยชื่อสัญลักษณ์แล้วกด tab

## Computing the series

- คำนวณค่าประมาณ pi สำหรับพจน์ที่ 3,5,10,100

```
for k in 1:3
    sum += factorial(4k) * ...
end
...
for k in 1:100
    sum += factorial(4k) * ...
end
```



## How about defining a function

- นิยามฟังก์ชันเพื่อคำนวณค่า pi ที่พจน์ต่างๆกัน

```
function approx_pi(n)
    for k in 1:n
        sum += factorial(4*k) ....
    end
    return 1/sum
end
```

# Storing results

- คำนวณแล้วเก็บผลลัพธ์ไว้ใน array

NOTE: Index ของ Julia เริ่มที่ 1 ไม่ใช่ 0

```
result = zeros(4) #array definition
for k in 1:10
    result[k] = approx_pi(k) # pi for each k
end
```

## Comparing with real pi

```
diff = pi - result
```

- เอาค่า pi จริงๆ มาลบออกจากค่าประมาณจะได้ความคลาดเคลื่อน
- pi ถึงแม้ไม่ใช่ array แต่หากกระทำกับ array จะเป็นการนำ pi ไปลบกับสมาชิกแต่ละตัวของ array นั้น

# Plotting the diff

- เราสามารถนำค่าผลต่างที่ได้มาพล็อตกราฟโดยใช้ Plots package
- Try it out!