

9C

```
1 import java.math.BigInteger;
2 class Exercise9C {
3     public static void factorial(int n){
4         int fact = 1;
5         int i;
6         for (i = 1 ; i<=n ; i++){
7             fact = i*fact;
8             System.out.println(i + "! = " + fact);
9         }
10    }
11    public static void factorialBig(int n){
12        int fact = 1;
13        BigInteger fbig = BigInteger.ONE;
14        BigInteger ibig;
15        for (int i=1; i<=n ; i++){
16            ibig = BigInteger.valueOf(i);
17            fbig = fbig.multiply(ibig);
18            System.out.println(i + "! = " + fbig );
19        }
20    }
21    public static void main(String[] args) {
22        factorial(20);
23        System.out.println();
24        factorialBig(20);
25    }
26 } //Supawit Saengrattayanon 64050694
```

----- Java Compile -----
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding=UTF-8
Output completed (0 sec consumed) - Normal Termination

```
C:\WINDOWS\system32\cm...
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
13! = 1932053504
14! = 1278945280
15! = 2004310016
16! = 2004189184
17! = -288522240
18! = -898433024
19! = 109641728
20! = -2102132736
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
13! = 6227020800
14! = 87178291200
15! = 1307674368000
16! = 20922789888000
17! = 355687428096000
18! = 6402373705728000
19! = 121645100408832000
20! = 2432902008176640000
Press any key to continue . . .
```

3. (9C) จงเขียนคลาสจาวาชื่อ Exer9C (ตัวอย่าง 7.2.9 และแบบฝึกหัดบทที่ 7 ตอน 3 ข้อ 5)

เพื่อสร้างตารางแฟคตอเรียล สำหรับค่า n ตั้งแต่ 1 ถึง 20 มีเมทอดดังนี้

- ให้มีคำสั่ง import ดังนี้ `import java.math.BigInteger;`

- เมทอด factorial แบบ static void มีตัวแปรรับค่า int n

```
int fact = 1;
for (int i=1; i<= n; i++) {
    fact = i * fact;
    System.out.println(i + "! = " + fact);
}
```

- เมทอด factorialBig แบบ static void มีตัวแปรรับค่า int n

```
int fact = 1;
BigInteger fbig = BigInteger.ONE;
BigInteger ibig;
for (int i=1; i<= n; i++) {
    ibig = BigInteger.valueOf(i); // convert int to BigInteger
    fbig = fbig.multiply( ibig ); // multiply in BigInteger
    System.out.println(i + "! = " + fbig );
}
```

- เมทอด main ทำงานโดยเรียกใช้เมทอดทั้ง 2 ดังนี้

```
factorial(20);
factorialBig(20);
```

ให้เขียนผลลัพธ์ของเมทอดทั้งสอง

ให้สังเกตและอธิบายว่ามีอะไรที่ไม่สมเหตุสมผลอย่างไร

ข้อควรระวัง BigInteger คือค่าที่ถูกต้องมากกว่า ข้อควรระวัง
เมื่อค่าถึง 2,100 ล้าน หรือจนกระทั่งถึง -2,100 ล้าน ทด:รับ
กลับมาจากเครื่องคอมพิวเตอร์

ผลลัพธ์จาก factorial

```
10! = 3 628 800
11! = 3 991 680 0
12! = 4 79 001 600
13! = 1 932 053 504
14! = 1 274 452 80
15! = 200 491 001 6
16! = 200 491 001 6
17! = -2 155 292 40
18! = -944 493 024
19! = 1044 417 29
20! = -2102152736
```

ผลลัพธ์จาก factorialBig

```
10! = 3 628 800
11! = 3 991 680 0
12! = 4 79 001 600
13! = 6 22 70 20 800
14! = 8 71 78 291 200
15! = 130 7 674 3 68 00
16! = 209 22 7 84 88 8000
17! = 355 68 7 4 28 0 96 000
18! = 640 23 7 37 0 5 72 8000
19! = 121 64 5 100 4 0 9 92 000
20! = 243 2 0 20 0 8 1 7 6 64 000 0
```

หมายเหตุ คลาส BigInteger รองรับจำนวนเต็มที่มีมากๆ เป็นอ็อบเจกต์

จึงมีการดำเนินการด้วยเมทอดของคลาส ตัวอย่างเมทอด เช่น

<code>bi = new BigInteger(s);</code>	สร้างจำนวน BigInteger ด้วยสตริงของเลขฐาน 10
<code>bi = BigInteger.ONE;</code>	กำหนดค่าให้เป็น 1 แบบ BigInteger จากค่าที่กำหนดไว้แล้ว
<code>bi = BigInteger.ZERO;</code>	กำหนดค่าให้เป็น 0 แบบ BigInteger จากค่าที่กำหนดไว้แล้ว
<code>bi = BigInteger.valueOf(lng);</code>	สร้างจำนวน BigInteger จากนิพจน์ของประเภท Long และ int
<code>bi1 = bi2.abs();</code>	หาค่า absolute แบบ BigInteger
<code>bi1 = bi2.add(bi3);</code>	บวกจำนวน (bi2 + bi3) แบบ BigInteger
<code>bi1 = bi2.subtract(bi3);</code>	ลบจำนวน (bi2 - bi3) แบบ BigInteger
<code>bi1 = bi2.multiply(bi3);</code>	คูณจำนวน (bi2 * bi3) แบบ BigInteger
<code>bi1 = bi2.divide(bi3);</code>	หารจำนวน (bi2 / bi3) แบบ BigInteger