

ASSIGNMENT 1 -6207050164-

Source Code

```
1 package ass;
2
3 import java.util.*;
4
5 public class Ass1 {
6
7     static int factorial(int n, int begin) {
8         int ans;
9         if(n == 0) {
10             System.out.printf("0! is base case return answer of 0! = 1");
11             ans = 1;
12         }
13         else
14         {
15             System.out.printf("%d! is recursive case. Answer = %d * recursion of %d!\n", n, n, (n-1));
16             System.out.printf("    Recursion to calculate %d!\n", (n-1));
17             ans = (n*factorial(n-1, begin));
18         }
19
20         if(n != begin) {
21             System.out.printf("\nCalculate %d! complete.",n);
22             System.out.printf("\n    Return answer from %d! = %d to calculate %d = [%d * %d!] = %d * %d = %d", n, ans, n+1, n+1, n, n+1, ans, (n+1)*ans);
23         }
24         else {
25             //System.out.printf("\nCalculate %d! complete.",n);
26             System.out.printf("\nComplete calculation of %d! answer = %d\n",begin ,ans);
27         }
28         return ans;
29     }
30
31     static int getint(int min, int max) {
32
33         int n, a = 0;
34         boolean success = false;
35         Scanner in = new Scanner(System.in);
36         while(!success) {
37             try {
38                 a = in.nextInt();
39
40                 if(a >= min && a <= max) {
41                     success = true;
42                 }
43                 else {
44                     System.out.printf("Input error, please enter between %d - %d", min, max);
45                     System.out.printf("\nEnter number: ");
46                 }
47             } catch(Exception e) {
48                 System.out.printf("Input error, please enter between %d - %d", min, max);
49                 in.nextLine();
50                 System.out.printf("\nEnter number: ");
51             }
52         }
53         return a;
54     }
55
56     public static void main(String[] args) {
57         int n;
58         Scanner in = new Scanner(System.in);
59         char ch;
60         int ans;
61         System.out.printf("My Recursion Programs.");
62         System.out.printf("\nProgram calculate n! by recursion (n<=15)");
63         System.out.printf("\n-----|");
64         do{
65             System.out.printf("\nEnter number: ");
66             n = getint(0,15);
67             ans = factorial(n,n);
68
69             System.out.printf("\npress [y] to continue, others to exit.");
70             ch = in.next().charAt(0);
71         }while (ch == 'y' || ch == 'Y');
72         System.out.printf("\n*****");
73         System.out.printf("\nEnd Program");
74         System.out.printf("\nProgram written by 62070501064 Onwipa Kujaroenpaisan");
75         System.out.printf("\n*****");
76     }
```

หลักการทำงาน

คำนวณ Factorial โดยใช้หลักการของ Recursion นั่นคือการเรียกใช้ตัวเองหรือวนซ้ำ เพื่อลดความซับซ้อนของโปรแกรมนั่นเอง

โดยที่เมื่อโปรแกรมรับค่าที่ป้อนมาแล้ว หากพบ error เช่น มีตัวอักษร หรือไม่อยู่ระหว่าง 1- 15 โปรแกรมจะเตือนและให้ผู้ใช้ป้อนค่าใหม่ แต่หากไม่พบ error จะนำไปสู่การ recursion ในฟังก์ชัน factorial โดยเช็คค่าเลขที่รับมา เท่ากับ 0 หรือไม่

หากค่าที่ป้อนมา (ให้เป็น n) เท่ากับ 0 โปรแกรมจะให้ตัวแปร `ans` เป็น 1 แล้วไปสู่ `if - else` ในการแสดงข้อความการ recursion และ return คำตอบ

หากค่า n มากกว่า 0 โปรแกรมจะส่งไปคำนวณต่อโดยการเรียกใช้ฟังก์ชันตัวเอง และคำนวณต่อจนกว่า จะถึง base case หรือ $n = 0$ นั่นเอง เมื่อคำนวณเสร็จแล้วจึงเข้าสู่ `if - else` ที่แสดงข้อความการ recursion และ return คำตอบ

เมื่อออกจาก recursion แล้ว โปรแกรมจะมีการถามผู้ใช้ว่าต้องการจะคำนวณต่อหรือไม่ หากผู้ใช้ตกลงคำนวณต่อ โดยการ ป้อน `y` โปรแกรมจะทำการรันรันใหม่ โดยให้ป้อนค่าใหม่ หากผู้ใช้ ป้อนตัวอักษรอื่น โปรแกรมจะจบการทำงาน

Test Case

กรณี ป้อนเลข 4

```
My Recursion Programs.
Program calculate n! by recursion (n<=15)
-----
Enter number: 4
4! is recursive case. Answer = 4 * recursion of 3!
  Recursion to calculate 3!
3! is recursive case. Answer = 3 * recursion of 2!
  Recursion to calculate 2!
2! is recursive case. Answer = 2 * recursion of 1!
  Recursion to calculate 1!
1! is recursive case. Answer = 1 * recursion of 0!
  Recursion to calculate 0!
0! is base case return answer of 0! = 1
Calculate 0! complete.
  Return answer from 0! = 1 to calculate 1 = [1 * 0!] = 1 * 1 = 1
Calculate 1! complete.
  Return answer from 1! = 1 to calculate 2 = [2 * 1!] = 2 * 1 = 2
Calculate 2! complete.
  Return answer from 2! = 2 to calculate 3 = [3 * 2!] = 3 * 2 = 6
Calculate 3! complete.
  Return answer from 3! = 6 to calculate 4 = [4 * 3!] = 4 * 6 = 24
Complete calculation of 4! answer = 24

press [y] to continue, others to exit.y
```

เมื่อค่าที่ได้รับ ไม่พบ error โปรแกรมจึงเข้าสู่ recursion และคำนวณจนถึง base case และแสดงข้อความการ recursion และได้คำตอบของ $4! = 24$ นั้นเอง

กรณี ป้อนค่าผิด

```
Enter number: 3x
Input error, please enter between 0 - 15
Enter number: x
Input error, please enter between 0 - 15
Enter number: -1
Input error, please enter between 0 - 15
Enter number: 20
Input error, please enter between 0 - 15
```

เมื่อพบ error โปรแกรมจึงเตือนและให้ป้อนค่าใหม่

กรณี ป้อนเลข 0 และ 1

```
Enter number: 0
0! is base case return answer of 0! = 1
Complete calculation of 0! answer = 1

press [y] to continue, others to exit.y

Enter number: 1
1! is recursive case. Answer = 1 * recursion of 0!
    Recursion to calculate 0!
0! is base case return answer of 0! = 1
Calculate 0! complete.
    Return answer from 0! = 1 to calculate 1 = [1 * 0!] = 1 * 1 = 1
Complete calculation of 1! answer = 1

press [y] to continue, others to exit.n

*****
End Program
Program written by 62070501064 Onwipa Kujaroenpaisan
*****
```

เมื่อค่าที่ได้รับ ไม่พบ error โปรแกรมจึงเข้าสู่ recursion และคำนวณจนถึง base case และแสดงข้อความการ recursion และได้คำตอบของ 0! และ 1! เท่ากับ 1 นั่นเอง

ปัญหาที่พบในการทำ ASSIGNMENT

- ยังไม่คล่องในการเขียนภาษาจาวา
- ปีกกาไม่ครบ
- ยังสับสนคำสั่งของจาวา และ C

Self-Assessment : 3 เข้าใจแต่มีปัญหาบางช่วงยังต้องขอความช่วยเหลือ