

การตรวจสอบวงเล็บปีกกา (BracketChecking)

จงรับข้อความซึ่งเป็นชุดคำสั่ง (block statment) แล้วระบุว่าข้อความดังกล่าวมีการใช้เครื่องหมายปีกกา “{ }” ได้อย่างถูกต้อง

อัลกอริทึมสำหรับการหาคู่ของเครื่องหมาย

Function CheckBracket(String s) : boolean

1.1 (a) หาดำแหน่งเครื่องหมายวงเล็บปีกกาเปิด (p0) ปรากฏเป็นครั้งแรก

(b) หาดำแหน่งเครื่องหมายวงเล็บปีกกาปิด (p1) ที่ปรากฏสุดท้าย

ถ้าไม่มีวงเล็บใด ให้คืนค่า false

ถ้ามีทั้งวงเล็บปีกกาเปิดและวงเล็บปีกกาปิด ให้คืนค่า (CheckBracket(s0+s1+s2))

คือตรวจสอบข้อความที่เอาวงเล็บที่จับคู่ออกไป

โดย s0 คือข้อความก่อนวงเล็บปีกกา

s1 คือข้อความระหว่างวงเล็บ

และ s2 คือข้อความที่อยู่หลังวงเล็บปีกกาทั้งหมด

1.2 ถ้าไม่มี คืนค่า true

หมายเหตุ วงเล็บปีกกาอาจจะไม่เรียงว่าต้องเป็นวงเล็บเปิดก่อนหรือปิดก่อน แต่ถ้ามี 2 อันก็ถือว่าใช่ แต่การจับคู่จะจับตัวใกล้กันก่อนเสมอ จะจับ { กับ } เสมอ

ตัวอย่างเช่นข้อความ “if (true) {x=8;} else {x=0;}”

CheckBracket(“if (true) {x=8;} else {x=0;}”)

จะได้ { ณ ตำแหน่ง 10 และ } ณ ตำแหน่ง 27 เรียก CheckBracket(“if (true) x=8;} else {x=0;”)

จะได้ { ณ ตำแหน่ง 21 และ } ณ ตำแหน่ง 14 เรียก CheckBracket(“if (true) x=8; else x=0;”)

ไม่พบเครื่องหมายทั้งวงเล็บปีกกาเปิดและวงเล็บปีกกาปิด คืนค่าเป็น true

รูปแบบข้อมูลเข้า

มี 1 บรรทัดเป็นข้อความ

รูปแบบข้อมูลแสดงผล

เป็นค่า true เมื่อวงเล็บปีกกาจับคู่กันได้ หรือ false เมื่อวงเล็บปีกกาจับคู่ไม่ได้

ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
Hello MyDay { int x = 5+7; }	true
a = {{5},{4},{6}} print a;	true
{{3},{Monday,Tuesday,Friday}};	true
{9*6	false
If x> 8 : { print x; if (y > 0) : {print error;} }	true
y = 9 }	false
ans = { 5 }, { 8 }	true
}}{}}	true
Sawasdee Silpakorn	true
{}}	true

ตัวอย่างโปรแกรม

```

public class BracketChecking{
    public static boolean CheckBracket (String s){ เขียนโค้ดในส่วนนี้}
    public static void main (String[] args){
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        System.out.println(CheckBracket(str));
    }
}

```

คำแนะนำ ฟังก์ชันที่ช่วย

คลาส String

`int indexOf(String t)` คืนค่าตำแหน่งแรกของสตริง `t` เป็นพารามิเตอร์ในวงเล็บที่ปรากฏในสตริงที่เรียก

เช่น `s = "CP.SU.AC.TH"; s.indexOf(".")` จะได้คำตอบเป็น 2

`int indexOf(String t)` คืนค่าตำแหน่งสุดท้ายของสตริง `t` เป็นพารามิเตอร์ในวงเล็บที่ปรากฏในสตริงที่เรียก

เช่น `s = "CP.SU.AC.TH"; s.lastIndexOf(".")` จะได้คำตอบเป็น 8

`String substring(int ตำแหน่งเริ่มต้น, int ตำแหน่งสิ้นสุด)` คืนค่าเป็นสตริงที่ถูกตัด

เช่น `s = "CP.SU.AC.TH"; s.substring(0,2)` จะได้คำตอบเป็น "CP" ไม่รวมตำแหน่ง 2

`String substring(int ตำแหน่งเริ่มต้น)` คืนค่าเป็นสตริง ณ ตำแหน่งเริ่มต้น จนถึงสิ้นสุดสตริง

เช่น `s = "CP.SU.AC.TH"; s.substring(8)` จะได้คำตอบเป็น ".TH"

`int length()` คือค่าเป็นจำนวนตัวอักษรในสตริง

เช่น `s = "CP.SU.AC.TH"; s.length()` จะได้คำตอบเป็น 11

คลาส Math

`int min(int a, int b)` หาค่าน้อยสุด วิธีใช้เรียก `Math.min(3,4)` ได้คำตอบเป็น 3

`int max(int a, int b)` หาค่ามากที่สุด วิธีใช้เรียก `Math.max(3,4)` ได้คำตอบเป็น 4