



โครงการการขับเคลื่อนการจัดการเรียนรู้โค้ดดิ้งในโรงเรียน การอบรมครู หลักสูตรการจัดการเรียนรู้วิทยาการคำนวณสำหรับวิทยากรแกนนำ

CCT มัธยมศึกษา รุ่นที่ 2.1

ออนไลน์ วันที่ 10-16 พ.ค. 2564

ครูวัชรระ วงษ์ดี สพม.ตาก

โรงเรียนตากพิทยาคม www.tps.ac.th

นำเสนอสื่อการสอนเรื่องขั้นตอน

การแก้ปัญหาและการเขียนโปรแกรมแบบ

ลำดับ ทางเลือก และวนซ้ำ ด้วยเครื่องมือ

Flowgorithm และเว็บไซต์ replit.com



เนื้อหาเพิ่มเติมที่เกี่ยวข้องได้แก่

- การเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner
- การแปลง Coding เป็นผังงานด้วยโปรแกรม Visustin v8 Flow chart generator
- การแปลงรหัสจำลองเป็นผังงานทางออนไลน์อัตโนมัติด้วยแอป code2flow

ที่มาของสื่อต่างๆ : เว็บไซต์วิทยาการคอมพิวเตอร์ สำหรับโรงเรียนตากพิทยาคม <http://tps.comsci.info>



ข้อเสนอแนะการนำสื่อการสอนนี้ไปใช้สอนวิชาวิทยาการคำนวณในโรงเรียนของท่าน ในแต่ละระดับชั้น
แนะนำโดยครูวัชระ วงษ์ดี ฉบับปรับปรุงเมื่อวันที่ 18 พ.ค. 2564 เวลา 03.00 น.
สื่อการสอนนี้มีการพัฒนาเปลี่ยนแปลงข้อมูลตลอดเวลา เผยแพร่ที่ www.tps.ac.th หรือ
เว็บไซต์วิทยาการคอมพิวเตอร์ สำหรับโรงเรียนตากพิทยาคม <http://tps.comsci.info>
(ติดต่อครูวัชระ วงษ์ดี รร.ตากพิทยาคม ทางอีเมล www.facebook.com/watcom610 หรือ
watcom610@gmail.com)

- ม.1 สามารถนำสไลด์ทั้งหมดไปใช้สอนการเขียนโปรแกรมพื้นฐาน ไม่แนะนำวนซ้ำ 2 รอบ กรณีบาง รร.ไม่พร้อมฯ
- ม.2 ใช้ทบทวนและนำสไลด์ทั้งหมดไปใช้สอนเสริมเกี่ยวกับการแก้ปัญหาด้วยไพทอน วนซ้ำ 1-2 รอบ
- ม.3 ใช้ทบทวนก่อนเรียนวิชาวิทยาการคำนวณ ม.3 เรื่องซอฟต์แวร์ที่ใช้ในการพัฒนาแอปพลิเคชัน (อจท.หน้า89)
- ม.4 ใช้ทบทวนก่อนเรียนวิชาวิทยาการคำนวณ ม.4 หรือใช้เสริมการทำโครงงานเกมส์ทดสอบแม่สูตรคูณ
ซึ่งอยู่ระหว่างการพัฒนาสื่อการสอนสอดแทรกเนื้อหาเกี่ยวกับโครงงาน ซึ่งอ้างอิงหนังสือ สสวท.
รวมทั้งการวิเคราะห์และออกแบบระบบของหนังสือบริษัท อจท. (System Analysis and design : SA)
- ม.5 ใช้ทบทวนตามความเหมาะสม เน้นบูรณาการกับเนื้อหา Data Science (อยู่ระหว่างพัฒนาสื่อฯ)
- ม.6 ใช้ทบทวนตามความเหมาะสม

หลักการเขียนโปรแกรมคอมพิวเตอร์เบื้องต้น

ในการเขียนโปรแกรมเพื่อแก้ปัญหาต่าง ๆ จำเป็นต้องมีการวางแผน และออกแบบโปรแกรมไว้ล่วงหน้า โดยกำหนดขั้นตอนวิธีการทำงานให้ชัดเจน ซึ่งกระบวนการวิเคราะห์ และออกแบบโปรแกรมเรียกว่า วัฏจักรการพัฒนาระบบงาน **System Development Lift Cycle (SDLC)** ซึ่งมีกระบวนการทำงานเริ่มต้นจากการวิเคราะห์ปัญหาไปจนถึงการนำโปรแกรมไปใช้งาน และปรับปรุงพัฒนาระบบให้ดีขึ้น มีขั้นตอนของ วัฏจักรการพัฒนาระบบงาน ดังต่อไปนี้

- ขั้นตอนที่ 1 การวิเคราะห์ปัญหา (Problem Analysis)
- ขั้นตอนที่ 2 การออกแบบโปรแกรม (Program Design)
- ขั้นตอนที่ 3 การเขียนโปรแกรม (Program Coding)
- ขั้นตอนที่ 4 การทดสอบและแก้ไขโปรแกรม (Program Testing & Verification)
- ขั้นตอนที่ 5 การจัดทำเอกสารและคู่มือการใช้งาน (Program Documentation)
- ขั้นตอนที่ 6 การใช้งานจริง (Program Implement)
- ขั้นตอนที่ 7 การปรับปรุงและพัฒนาโปรแกรม (Program Maintenance)

ตัวอย่างโจทย์ปัญหา การหาแม่สูตรคูณ

educatepark.com/สูตรคูณ/

สูตรคูณ



2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
2 x 11 = 22
2 x 12 = 24

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
3 x 11 = 33
3 x 12 = 36

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
4 x 11 = 44
4 x 12 = 48

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
5 x 11 = 55
5 x 12 = 60

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
6 x 11 = 66
6 x 12 = 72

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
7 x 11 = 77
7 x 12 = 84



วิเคราะห์ปัญหา การหาแม่สูตรคูณ 1 ผลคูณ

INPUT =

OUTPUT =

PROCESS =

← → ↻ educatepark.com/สูตรคูณ/ 📄 🔍 ☆ ☰ M

วิธีการอ่านสูตรคูณ

ตัวอย่าง $4 \times 5 = 20$

จากตัวอย่างด้านบน จะเห็นเลขอยู่ 2 ตัว คือ เลข 4 กับ เลข 5 เลข 4 ตัวหน้า เรียกว่าตัวตั้ง ส่วนเลข 5 จะเรียกว่าตัวคูณนะครั้น เวลาท่องก็จะเป็น สี่ ห้า ยี่สิบ

สูตรคูณแม่ 2

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$2 \times 4 = 8$$

$$2 \times 5 = 10$$

$$2 \times 6 = 12$$

$$2 \times 7 = 14$$

$$2 \times 8 = 16$$

$$2 \times 9 = 18$$

$$2 \times 10 = 20$$

$$2 \times 11 = 22$$

$$2 \times 12 = 24$$



วิเคราะห์ปัญหาการหาแม่สูตรคูณ 1 ผลคูณ

(PROBLEM ANALYSIS)

INPUT = front , back

OUTPUT = multi

(PROBLEM ANALYSIS)

PROCESS =

- 1) INPUT front**
- 2) INPUT back**
- 3) COMPUTE $multi = front * back$**
- 4) OUTPUT multi**



การออกแบบโปรแกรมหรือออกแบบขั้นตอนวิธี (Program Design or Algorithms Design)

- ออกแบบโดยใช้ภาษาธรรมชาติ คือ เป็นการใช้ภาษาพูดที่เข้าใจง่ายเพื่ออธิบายขั้นตอนการทำงานของโปรแกรม
- ออกแบบโดยใช้รหัสจำลอง (Pseudo Code) คือ เป็นการใช้ภาษาที่มีโครงสร้างกระชับ และไม่ขึ้นกับภาษาของโปรแกรมใด
- ออกแบบโดยใช้ผังงาน (Flowchart) คือ เป็นการใช้สัญลักษณ์เพื่ออธิบายขั้นตอนการทำงานของโปรแกรม

ออกแบบรหัสจำลอง (Pseudo Code) การหาแม่สูตรคูณ 1 ผลคูณ

- PROBLEM ANALYSIS

INPUT = front , back

OUTPUT = multi

PROCESS =

- 1) INPUT front
- 2) INPUT back
- 3) COMPUTE $multi = front * back$
- 4) OUTPUT multi

PSEUDO CODE

- 1) START
- 2) DECLARE front, back, multi
- 3) INPUT front
- 4) INPUT back
- 5) COMPUTE $multi = front * back$
- 6) OUTPUT multi
- 7) STOP

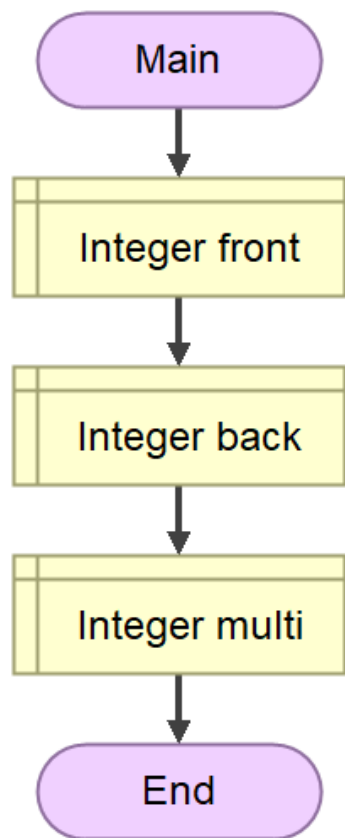


โครงสร้างโปรแกรม มี 3 แบบ

- โครงสร้างการทำงานแบบเรียงลำดับ คือ โปรแกรมทำงานต่อเนื่องกันเป็นเส้นตรงจากบนลงล่าง
- โครงสร้างโปรแกรมแบบเลือกทำ แบบ 1, 2 และหลายทางเลือก
- โครงสร้างโปรแกรมแบบวนซ้ำ แบบ while และ do while

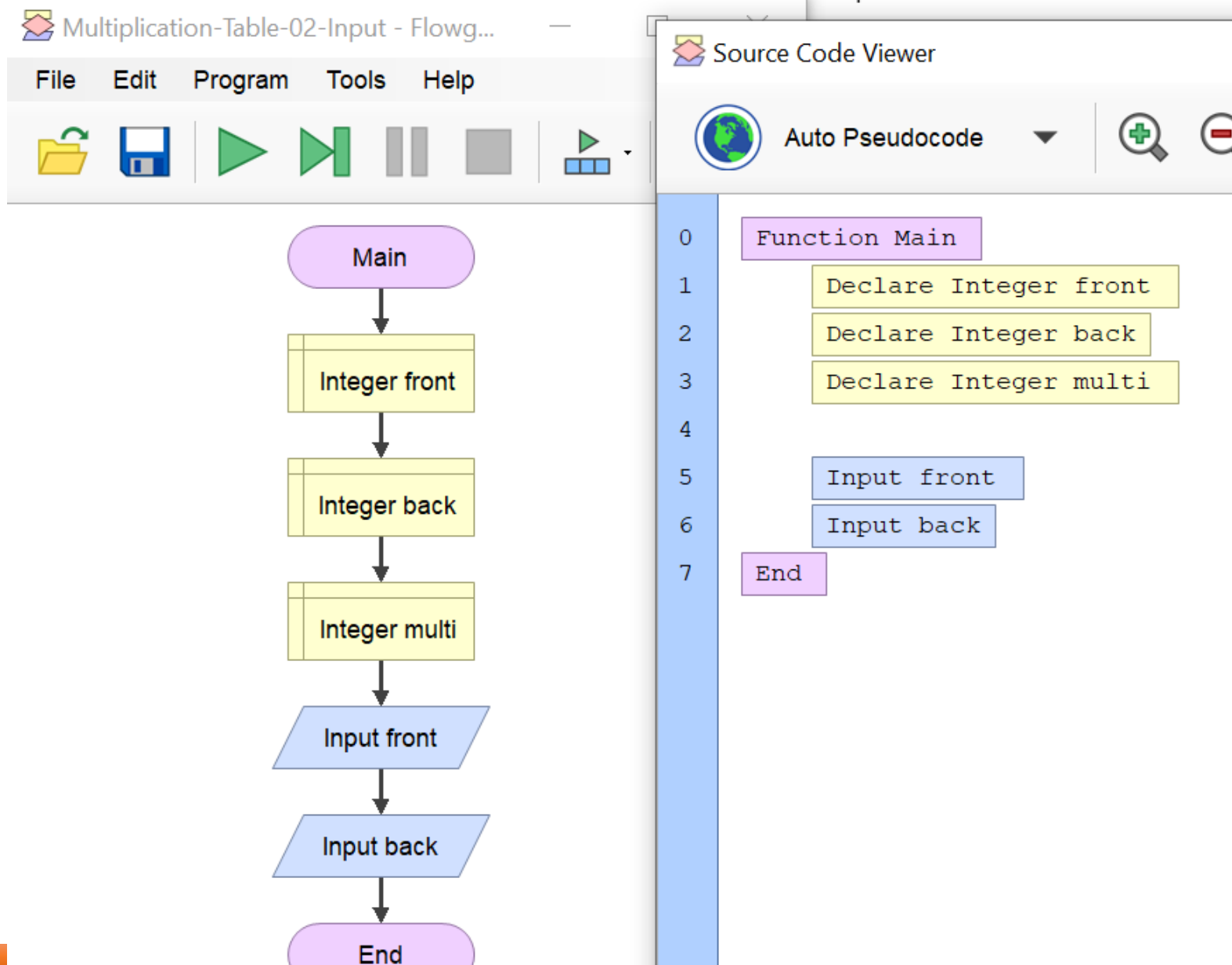
โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

ตัวอย่าง

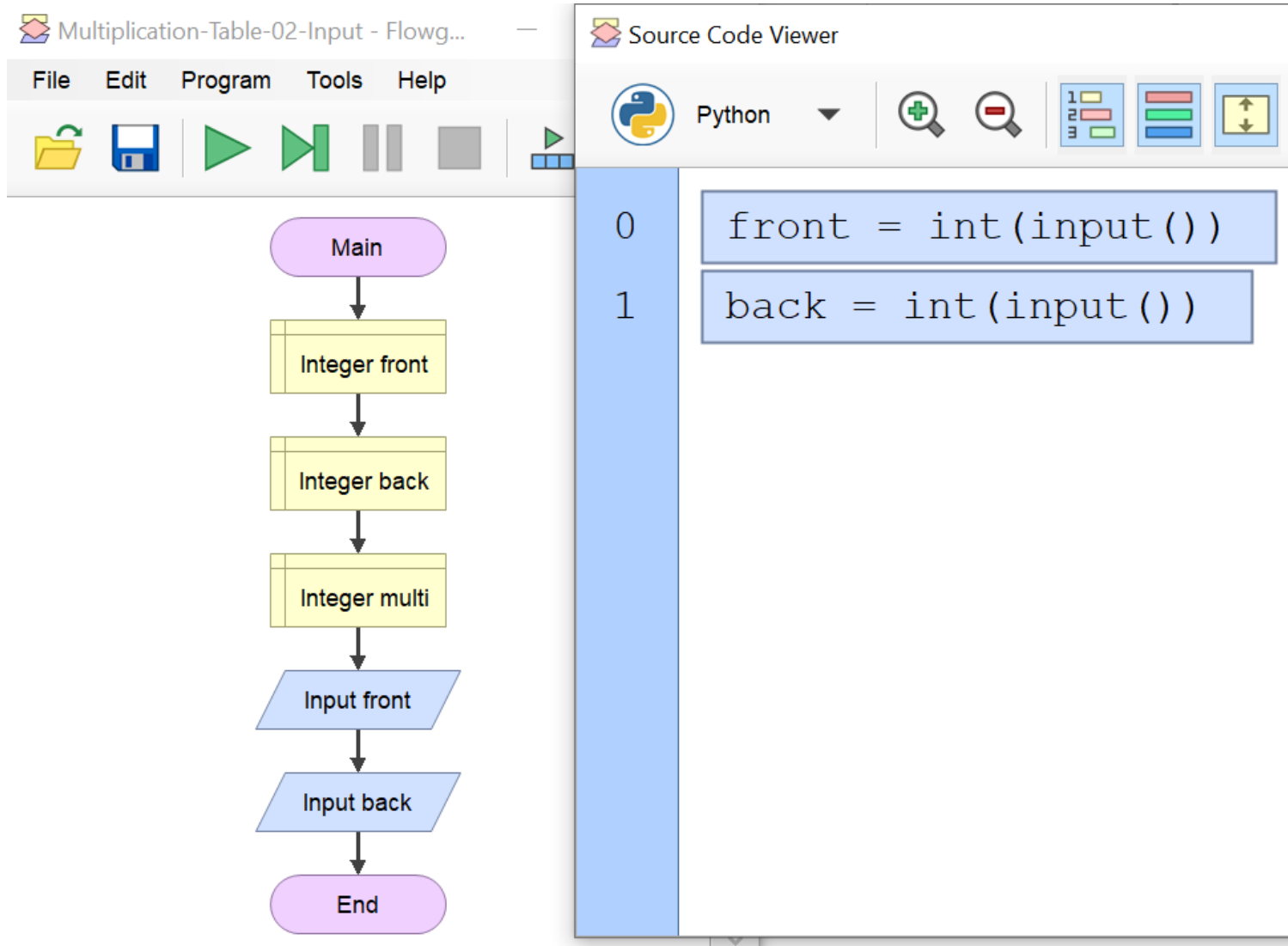


โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

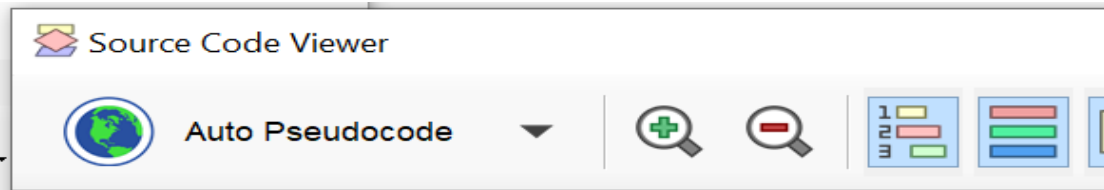
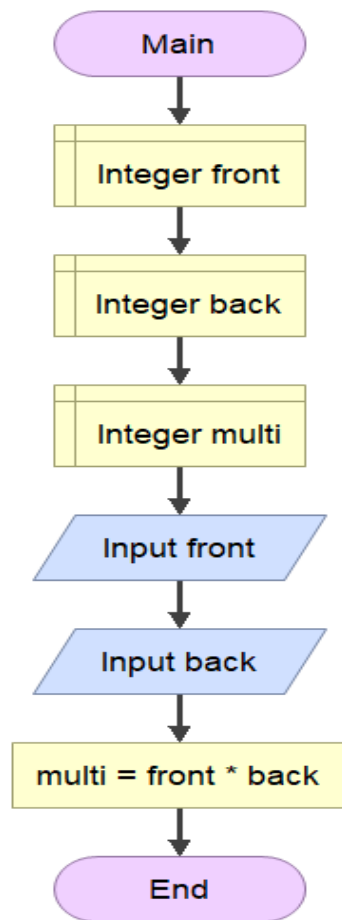
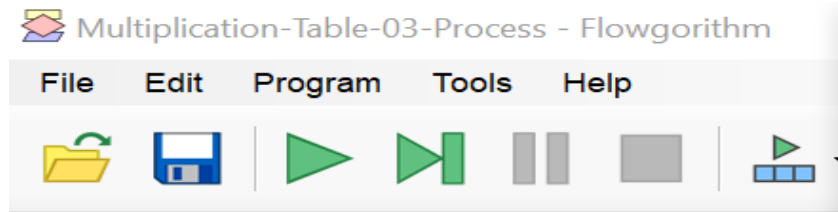
ตัวอย่าง



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



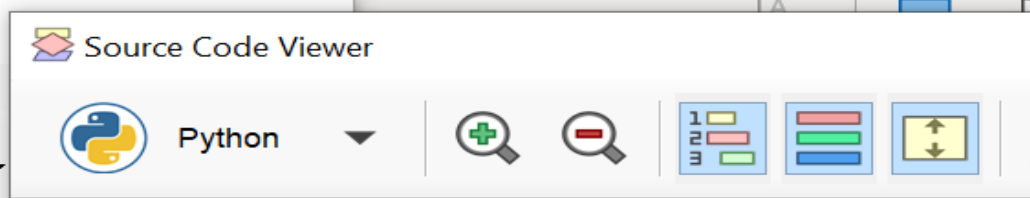
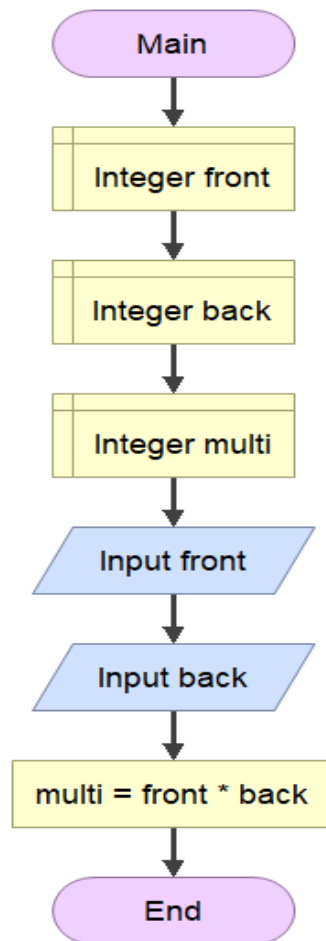
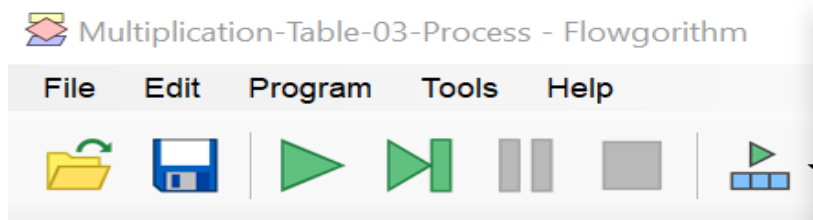
โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



```
0 Function Main
1     Declare Integer front
2     Declare Integer back
3     Declare Integer multi
4
5     Input front
6     Input back
7     Assign multi = front * back
8 End
```



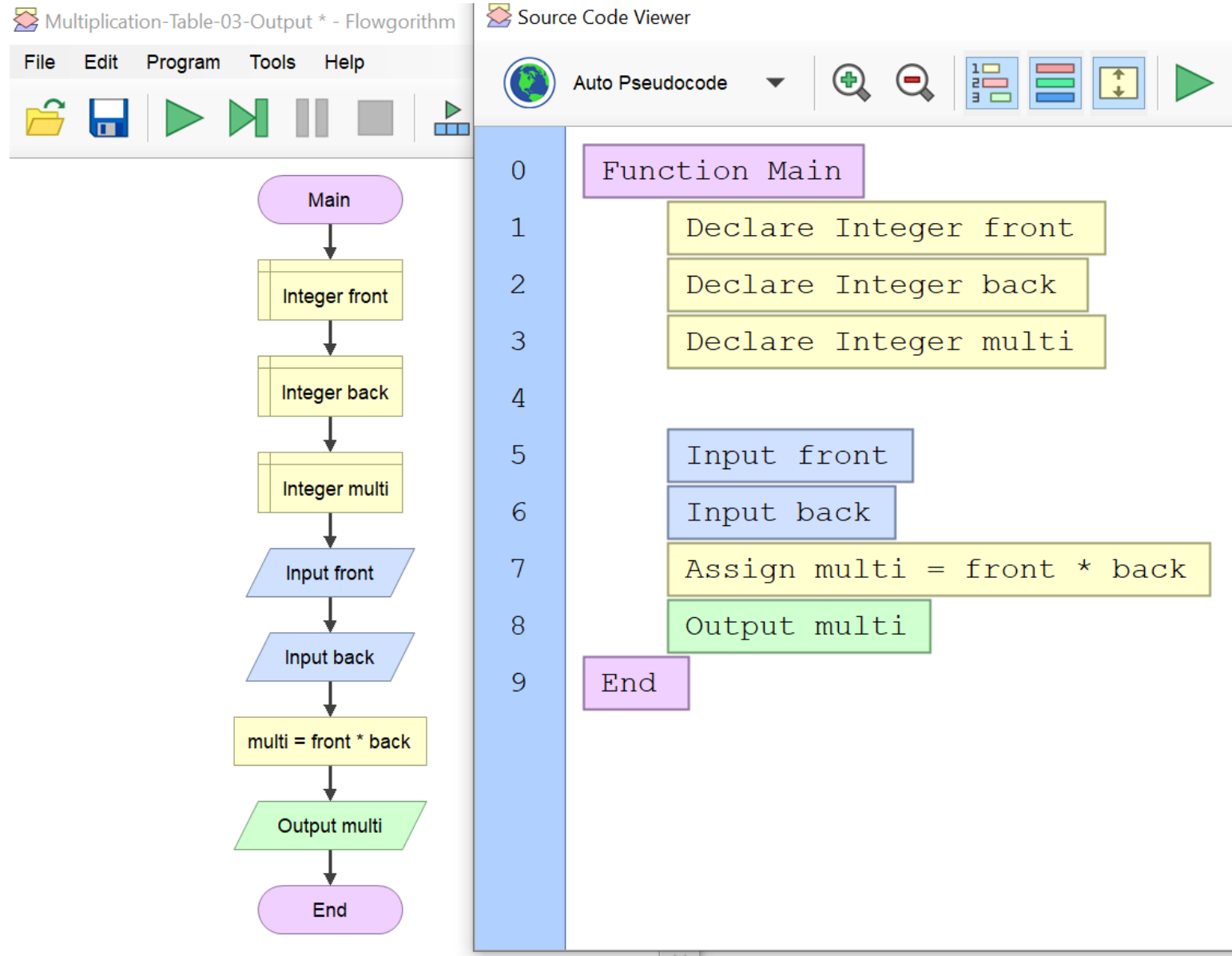
โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



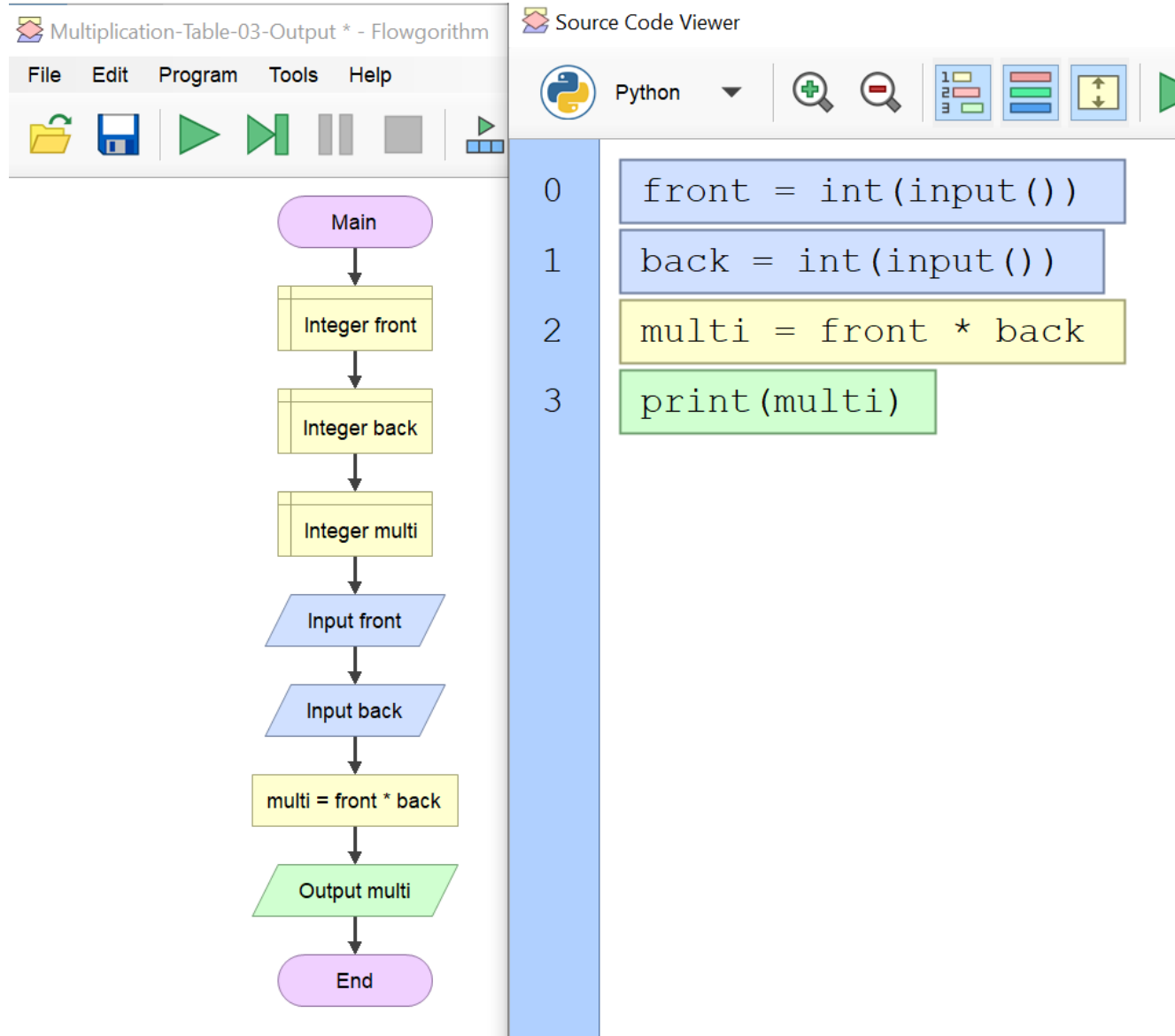
```
0 front = int(input())
1 back = int(input())
2 multi = front * back
```



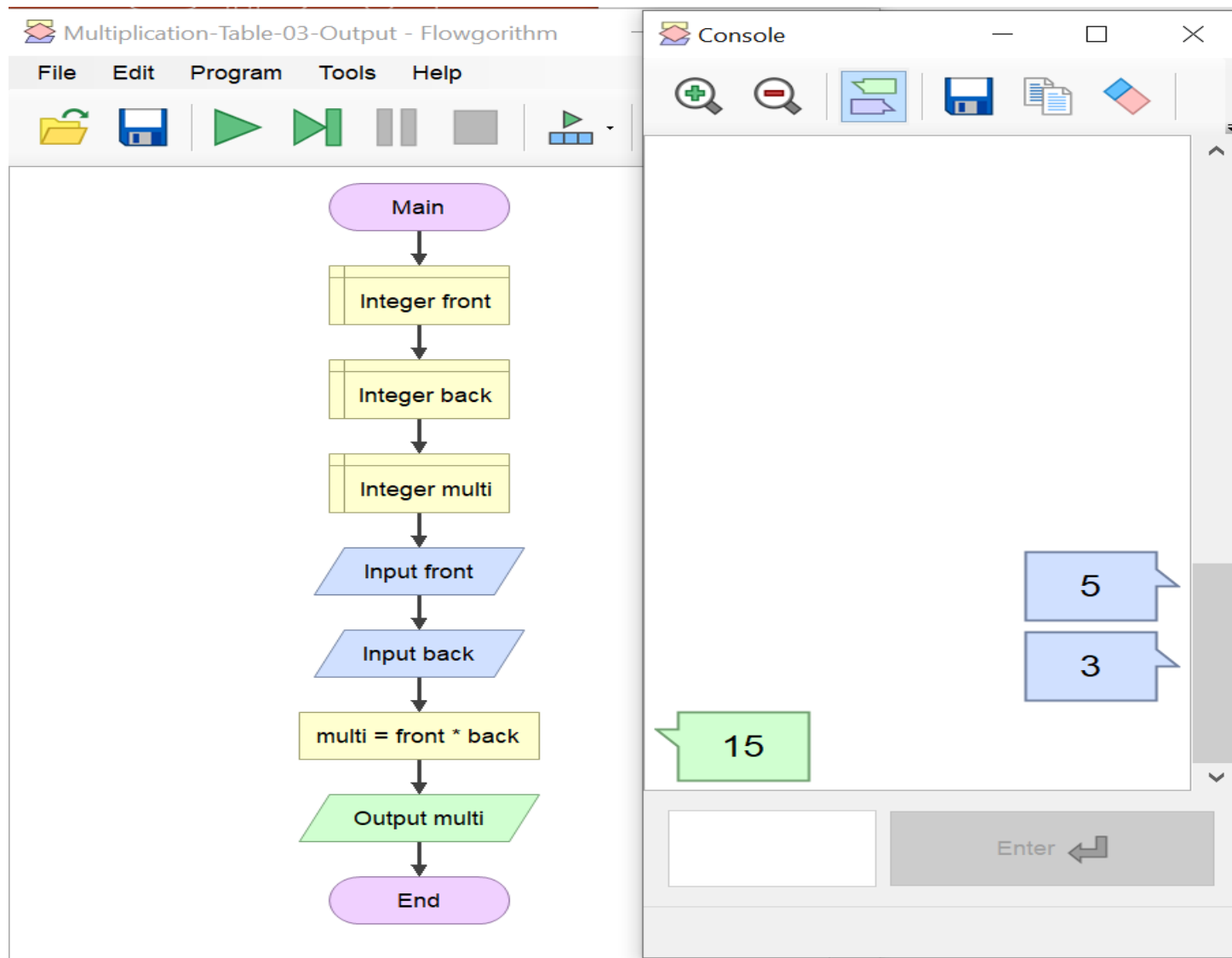
โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



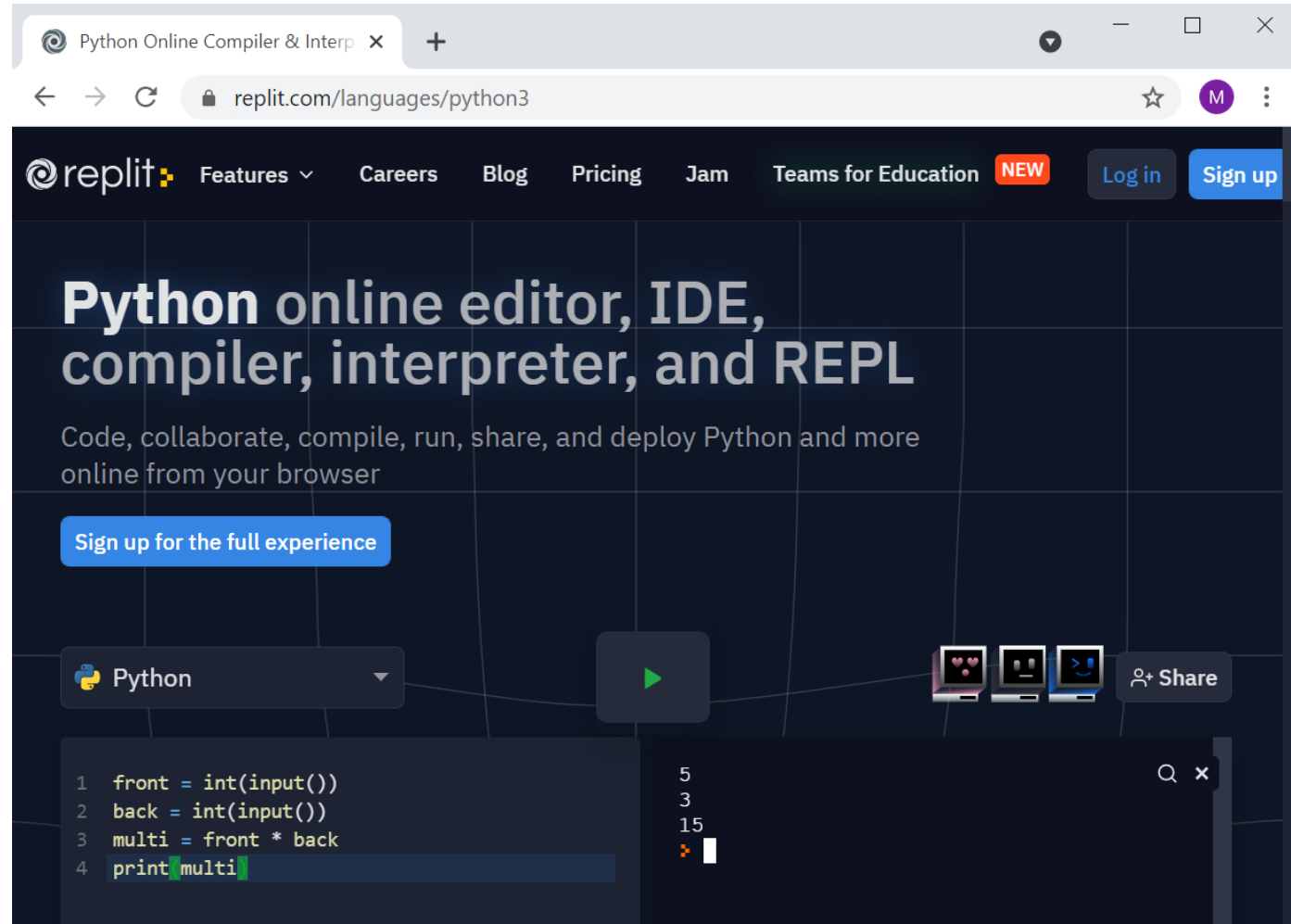
โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

นำ Source Code ภาษาไพทอนมาทดสอบโปรแกรมที่ลิงก์นี้ <https://replit.com/languages/python3> จะได้ผลลัพธ์ด้านขวานี้

```
front = int(input())  
back = int(input())  
multi = front * back  
print(multi)
```



The screenshot shows the Replit Python online editor interface. The browser address bar displays `replit.com/languages/python3`. The page header includes the Replit logo and navigation links for Features, Careers, Blog, Pricing, Jam, Teams for Education, Log in, and Sign up. The main heading reads "Python online editor, IDE, compiler, interpreter, and REPL". Below this, a sub-heading states "Code, collaborate, compile, run, share, and deploy Python and more online from your browser". A blue button labeled "Sign up for the full experience" is visible. The code editor shows the following Python code:

```
1 front = int(input())  
2 back = int(input())  
3 multi = front * back  
4 print(multi)
```

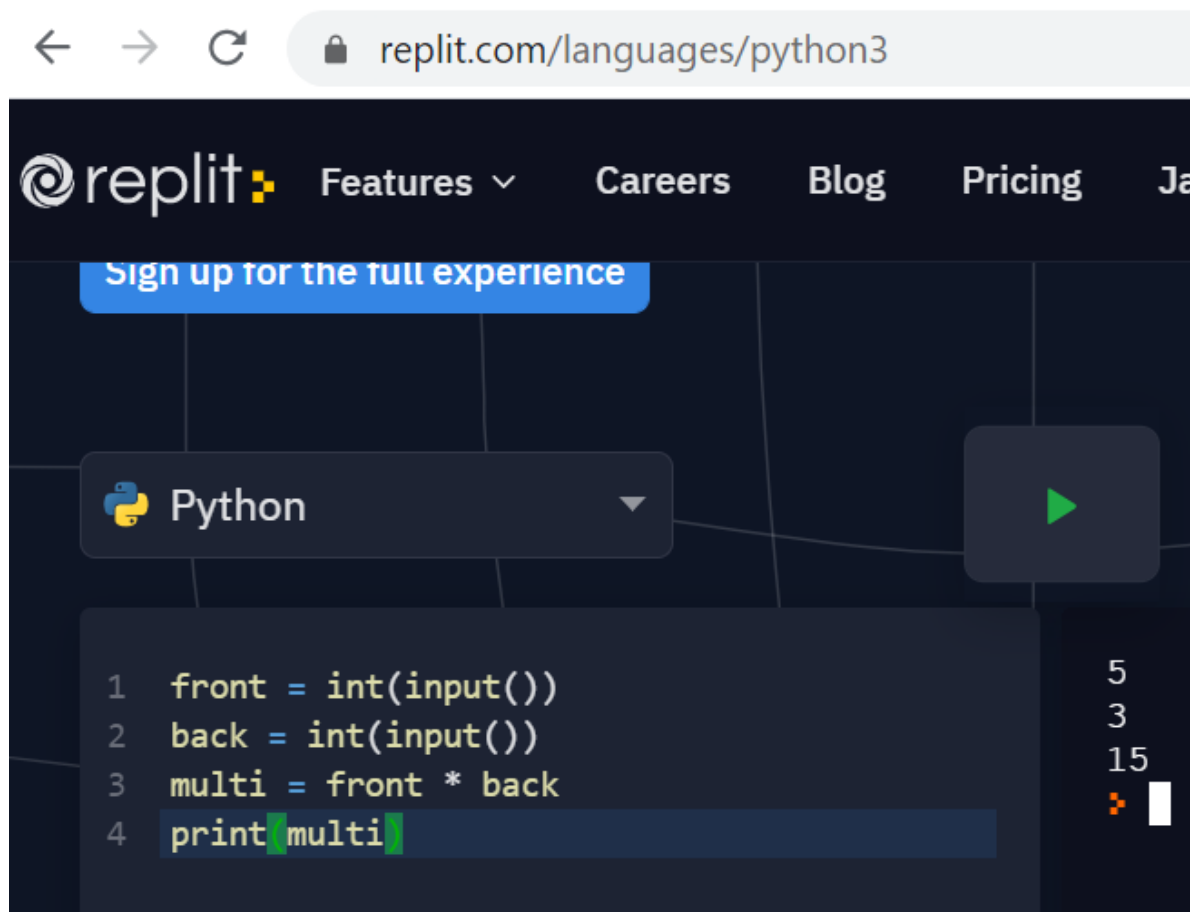
The output on the right side of the editor shows the results of running the code: `5`, `3`, and `15`. The interface also includes a language selector set to Python, a run button, and a share button.



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

นำ Source Code ภาษาไพทอนมาทดสอบโปรแกรมที่ลิงก์นี้ <https://replit.com/languages/python3> จะได้ผลลัพธ์ด้านขวานี้

```
front = int(input())  
back = int(input())  
multi = front * back  
print(multi)
```



The screenshot shows a web browser window with the URL `replit.com/languages/python3`. The page features the Replit logo and navigation links for Features, Careers, Blog, Pricing, and Jobs. A blue button prompts users to "Sign up for the full experience". Below this, a Python REPL interface is visible, including a Python logo, a "Python" label, and a green play button. The code being executed is:

```
1 front = int(input())  
2 back = int(input())  
3 multi = front * back  
4 print(multi)
```

The output on the right side of the REPL shows the results of the program's execution:

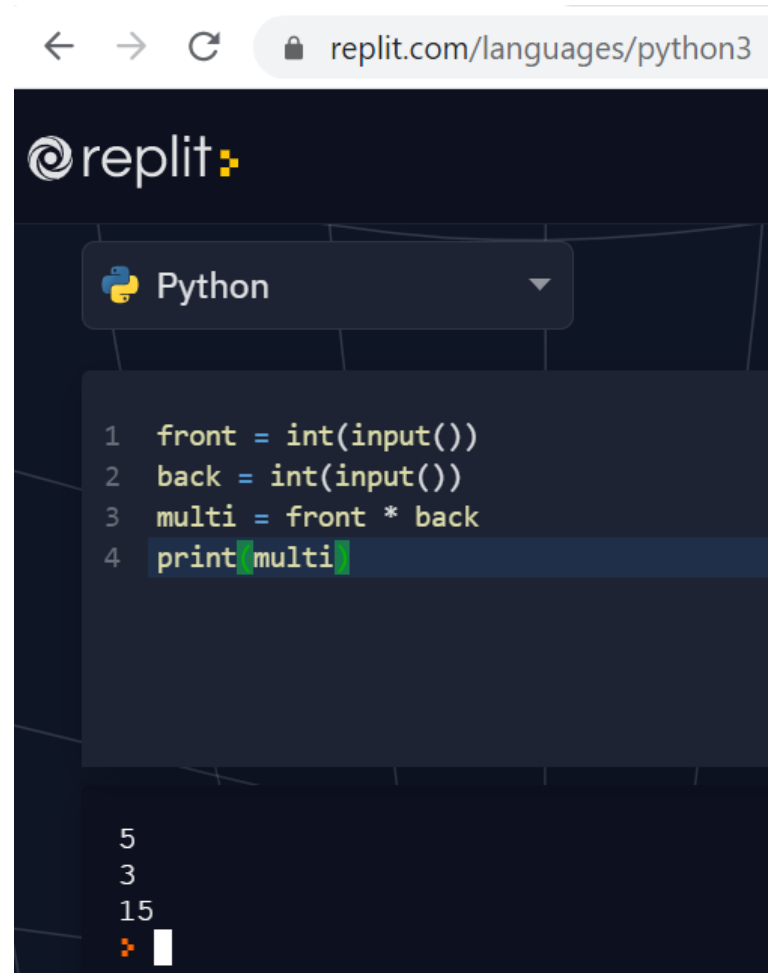
```
5  
3  
15
```



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

นำ Source Code ภาษาไพทอนมาทดสอบโปรแกรมที่ลิงก์นี้ <https://replit.com/languages/python3>
จะได้ผลลัพธ์ด้านขวานี้

```
front = int(input())  
back = int(input())  
multi = front * back  
print(multi)
```



The screenshot shows a web browser window with the URL `replit.com/languages/python3`. The page displays the Replit logo and a dropdown menu set to "Python". Below the menu is a code editor with the following Python code:

```
1 front = int(input())  
2 back = int(input())  
3 multi = front * back  
4 print(multi)
```

At the bottom of the editor, the output of the program is shown as:

```
5  
3  
15
```



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

นำ Source Code ภาษาไพทอนมาทดสอบโปรแกรมที่ลิงก์นี้ <https://replit.com/languages/python3>
จะได้ผลลัพธ์ด้านขวานี้

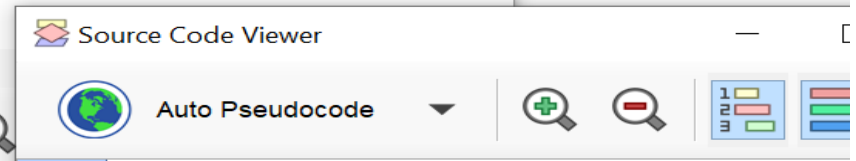
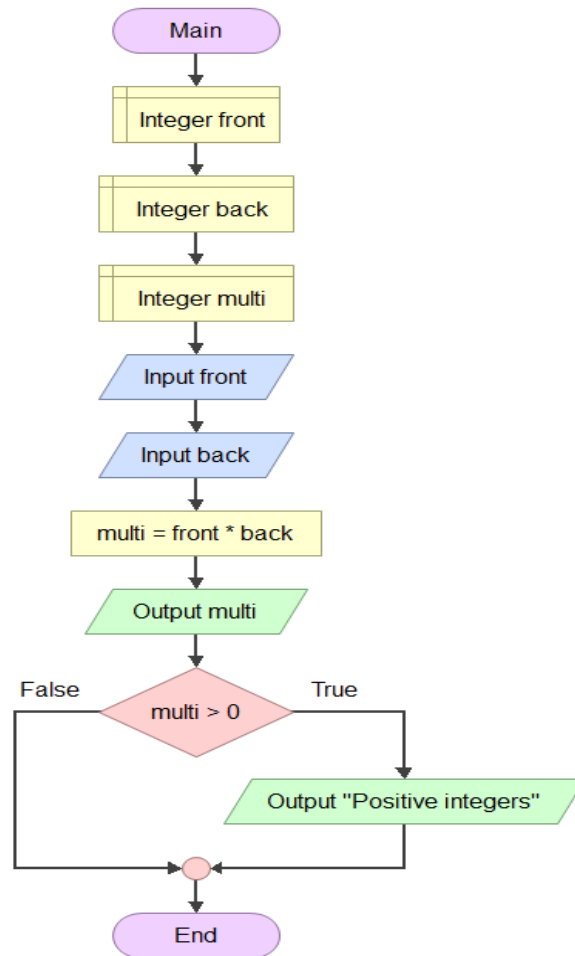
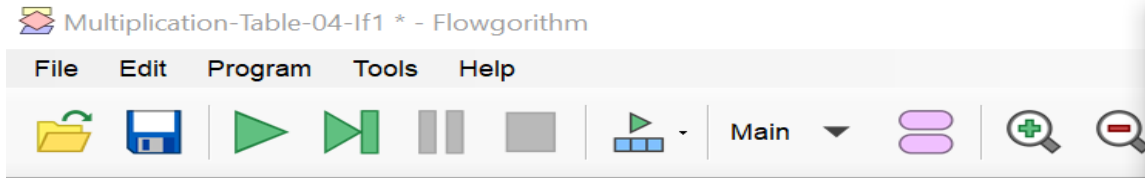
```
front = int(input())  
back = int(input())  
multi = front * back  
print(multi)
```

```
1 front = int(input())  
2 back = int(input())  
3 multi = front * back  
4 print(multi)
```

```
5  
3  
15  
❏
```



โครงสร้างโปรแกรมแบบ 1 ทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม



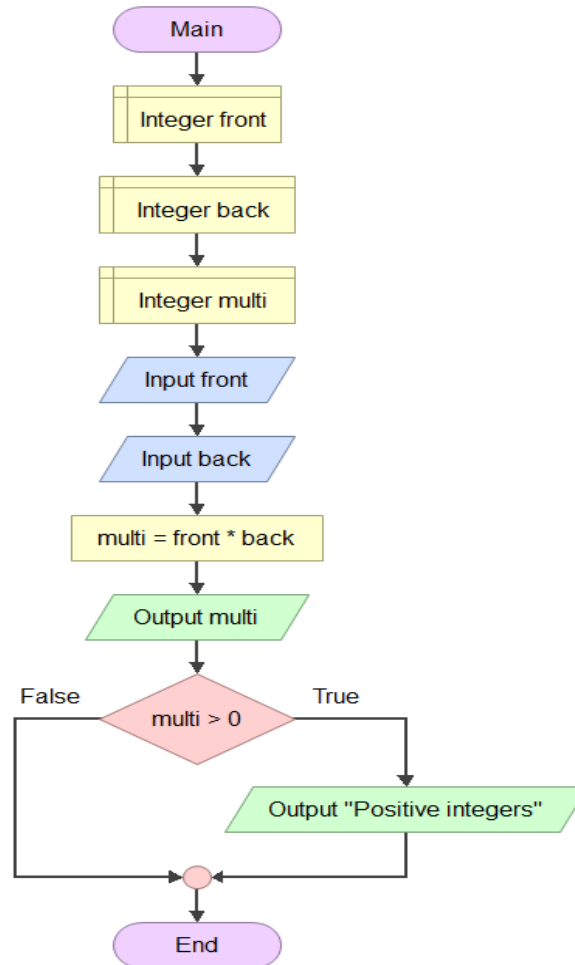
```
0  Function Main
1      Declare Integer front
2      Declare Integer back
3      Declare Integer multi
4
5      Input front
6      Input back
7      Assign multi = front * back
8      Output multi
9      If multi > 0
10         Output "Positive integers"
11     End
12 End
```



โครงสร้างโปรแกรมแบบ 1 ทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม

Multiplication-Table-04-If1 - Flowgorithm

File Edit Program Tools Help



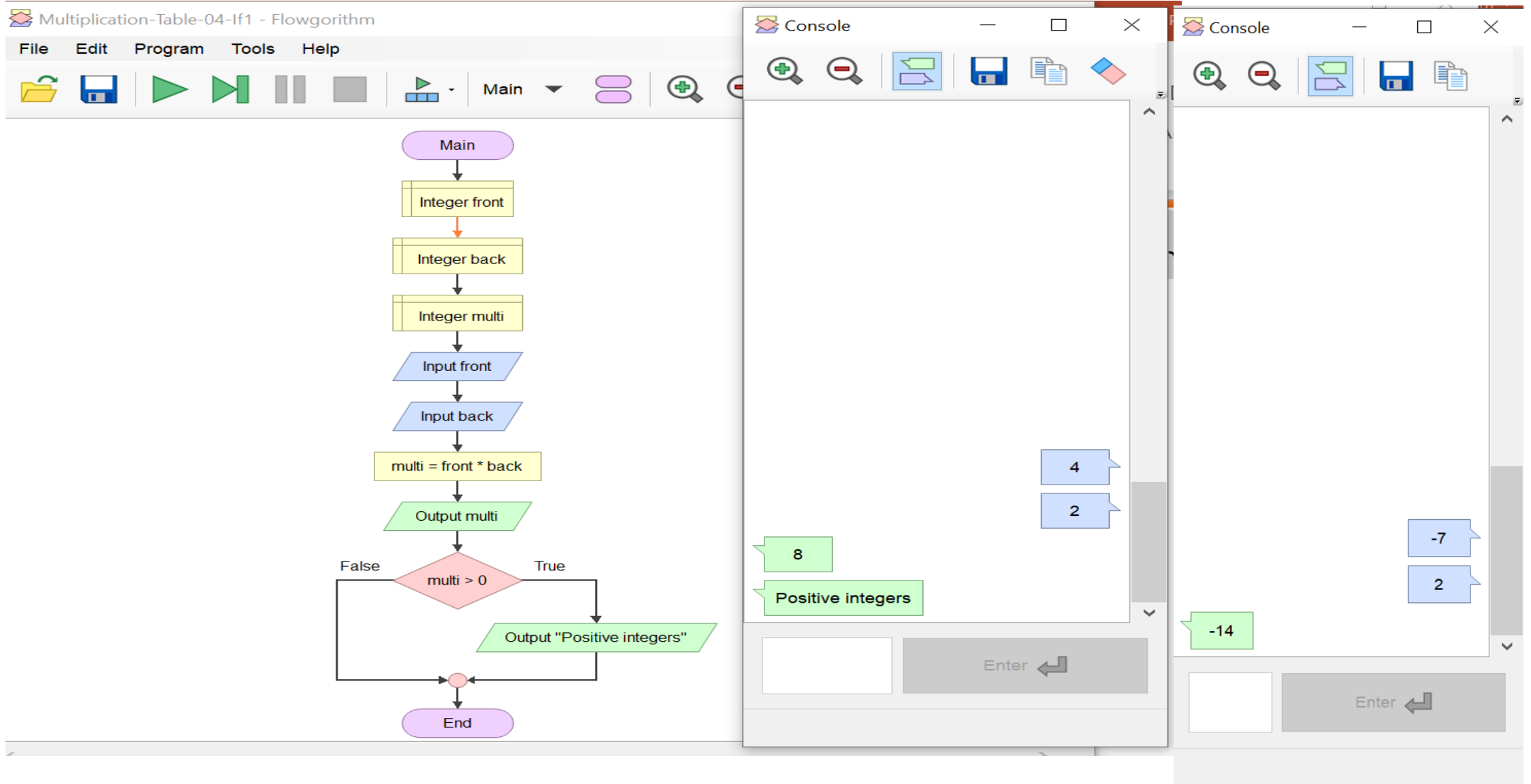
Source Code Viewer



```
0 front = int(input())
1 back = int(input())
2 multi = front * back
3 print(multi)
4 if multi > 0:
5     print("Positive integers")
```



โครงสร้างโปรแกรมแบบ 1 ทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

นำ Source Code ภาษาไพทอนมาทดสอบโปรแกรมที่ลิงก์นี้ <https://replit.com/languages/python3>
จะได้ผลลัพธ์ด้านขวานี้

```
front = int(input())  
back = int(input())  
multi = front * back  
print(multi)  
if multi > 0:  
    print("Positive integers")
```

← → ↻ replit.com/languages/python3

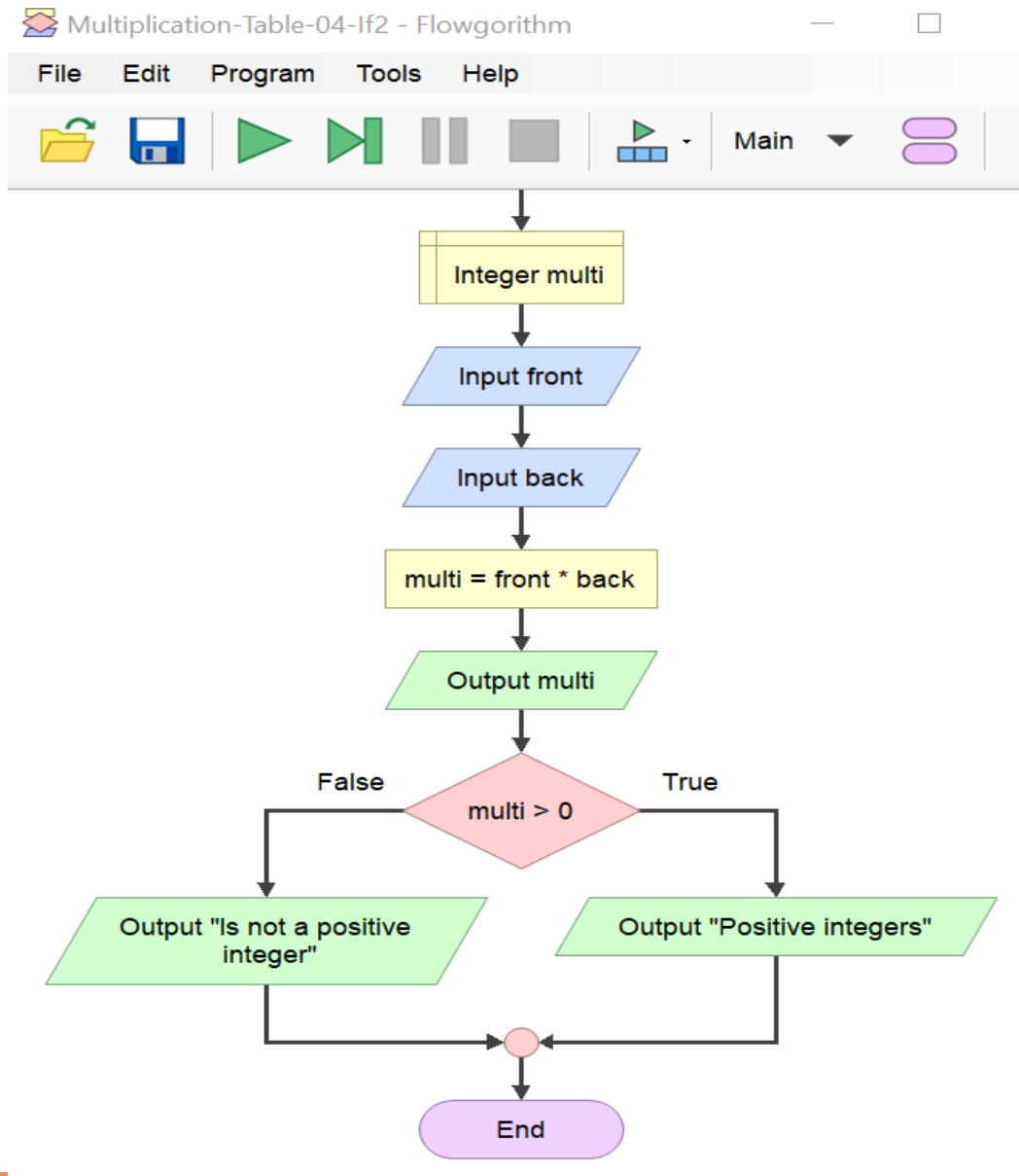
```
replit  
1 front = int(input())  
2 back = int(input())  
3 multi = front * back  
4 print(multi)  
5 if multi > 0:  
6     print("Positive integers")  
7
```

```
4  
5  
20  
Positive integers  
█
```

```
-4  
5  
-20  
█
```



โครงสร้างโปรแกรมแบบ 2 ทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม

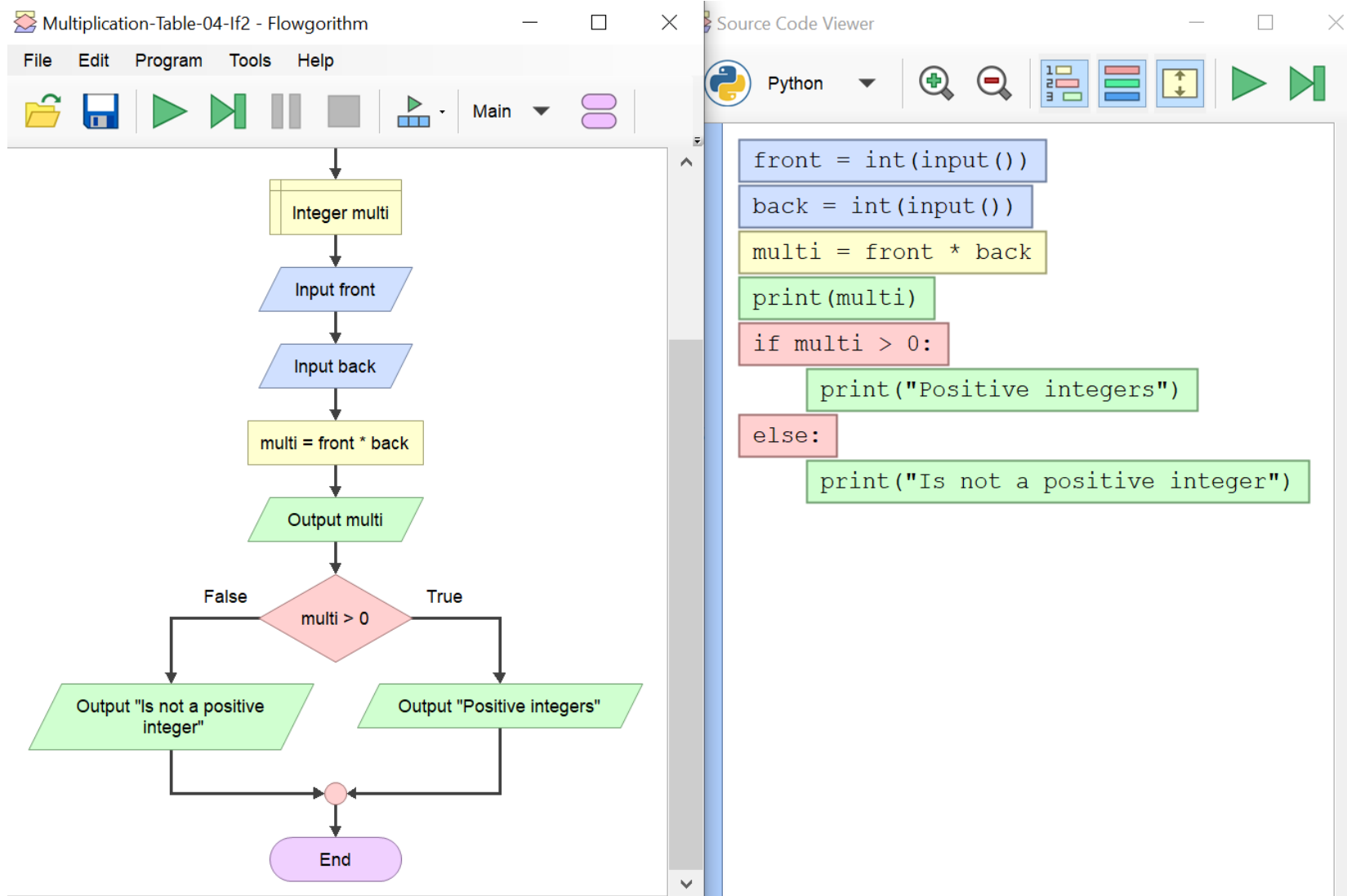


```
0  Function Main
1      Declare Integer front
2      Declare Integer back
3      Declare Integer multi
4
5      Input front
6      Input back
7      Assign multi = front * back
8      Output multi
9      If multi > 0
10         Output "Positive integers"
11     False:
12         Output "Is not a positive integer"
13     End
14 End
```

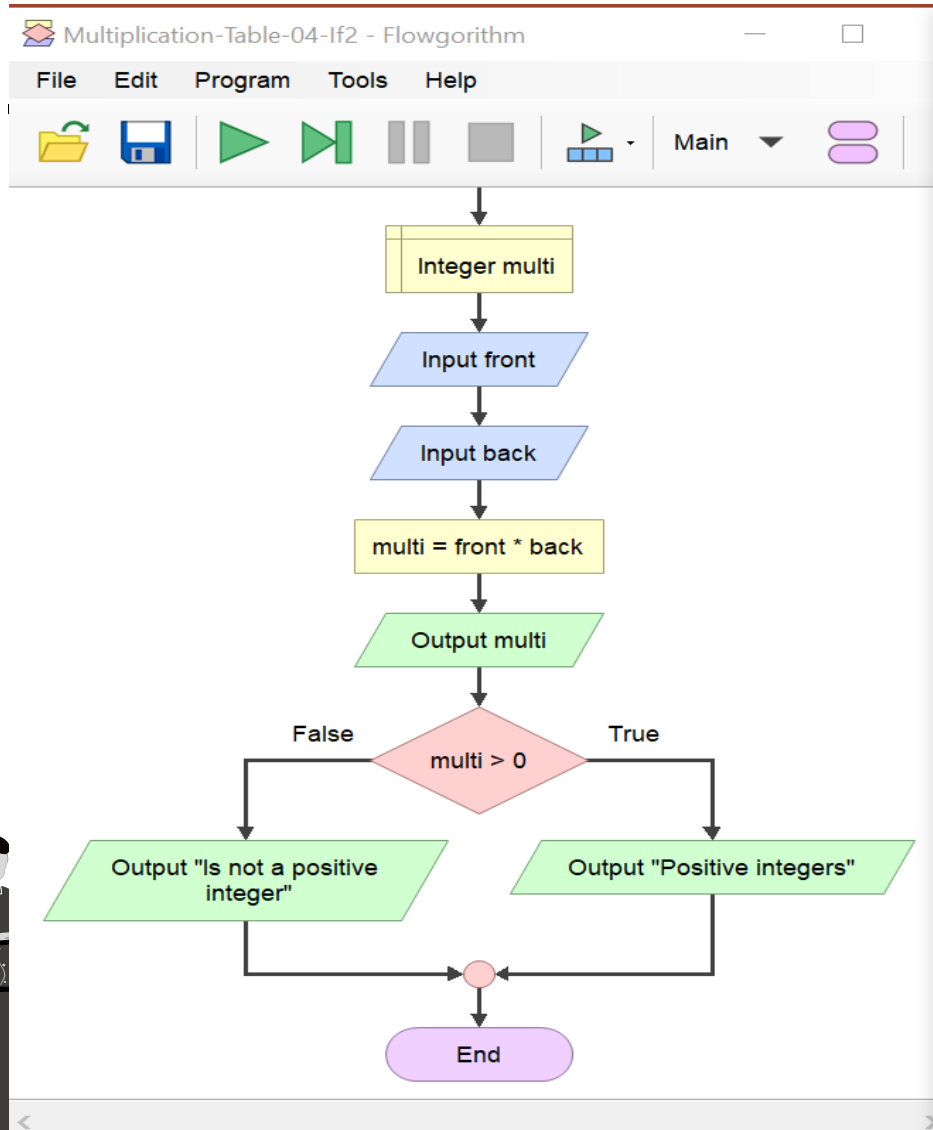
The Source Code Viewer window displays the code corresponding to the flowchart. It includes a menu bar with 'File', 'Edit', 'Program', 'Tools', and 'Help'. The code is organized into a 'Function Main' block. It declares three integer variables: 'front', 'back', and 'multi'. It then takes input for 'front' and 'back', calculates 'multi = front * back', and outputs the result. A conditional statement 'If multi > 0' branches the flow: if true, it outputs 'Positive integers'; if false, it outputs 'Is not a positive integer'. The code concludes with an 'End' statement.



โครงสร้างโปรแกรมแบบ 2 ทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม



โครงสร้างโปรแกรมแบบ 2 ทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม



Console window showing the execution of the program. The user has entered the numbers 5 and 8. The program outputs the result 40 and the message "Positive integers".

```
5
8
40
Positive integers
```

Console window showing the execution of the program. The user has entered the numbers -4 and 2. The program outputs the result -8 and the message "Is not a positive integer".

```
-4
2
-8
Is not a positive integer
```



โครงสร้างโปรแกรมแบบลำดับของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

นำ Source Code ภาษาไพทอนมาทดสอบโปรแกรมที่ลิงก์นี้ <https://replit.com/languages/python3>
จะได้ผลลัพธ์ด้านขวานี้

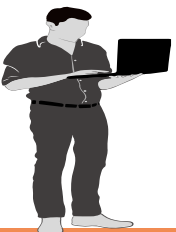
```
front = int(input())
back = int(input())
multi = front * back
print(multi)
if multi > 0:
    print("Positive integers")
else:
    print("Is not a positive integer")
```

```
replit

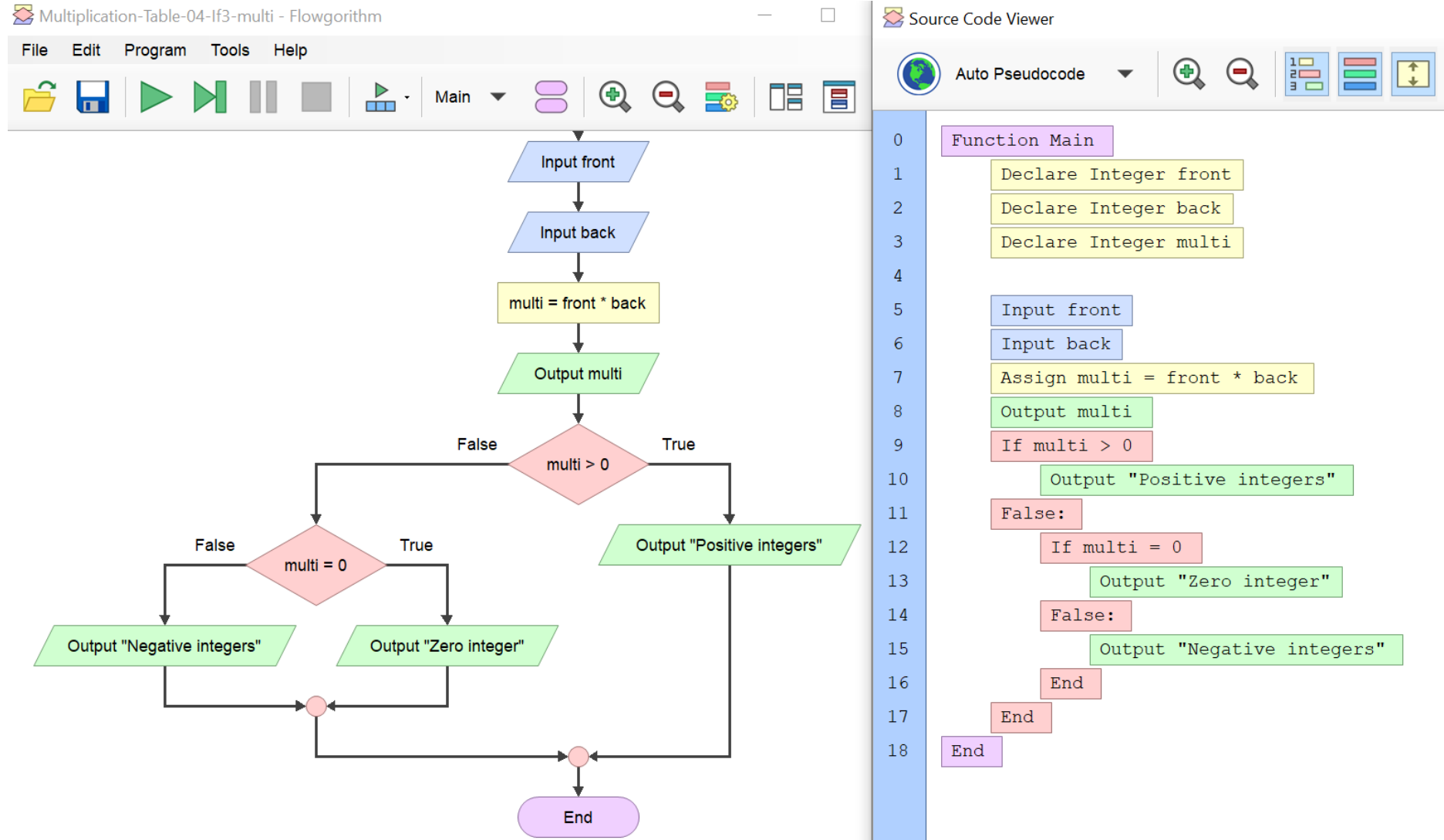
1 front = int(input())
2 back = int(input())
3 multi = front * back
4 print(multi)
5 if multi > 0:
6     print("Positive integers")
7 else:
8     print("Is not a positive integer")
```

```
2
7
14
Positive integers
>
```

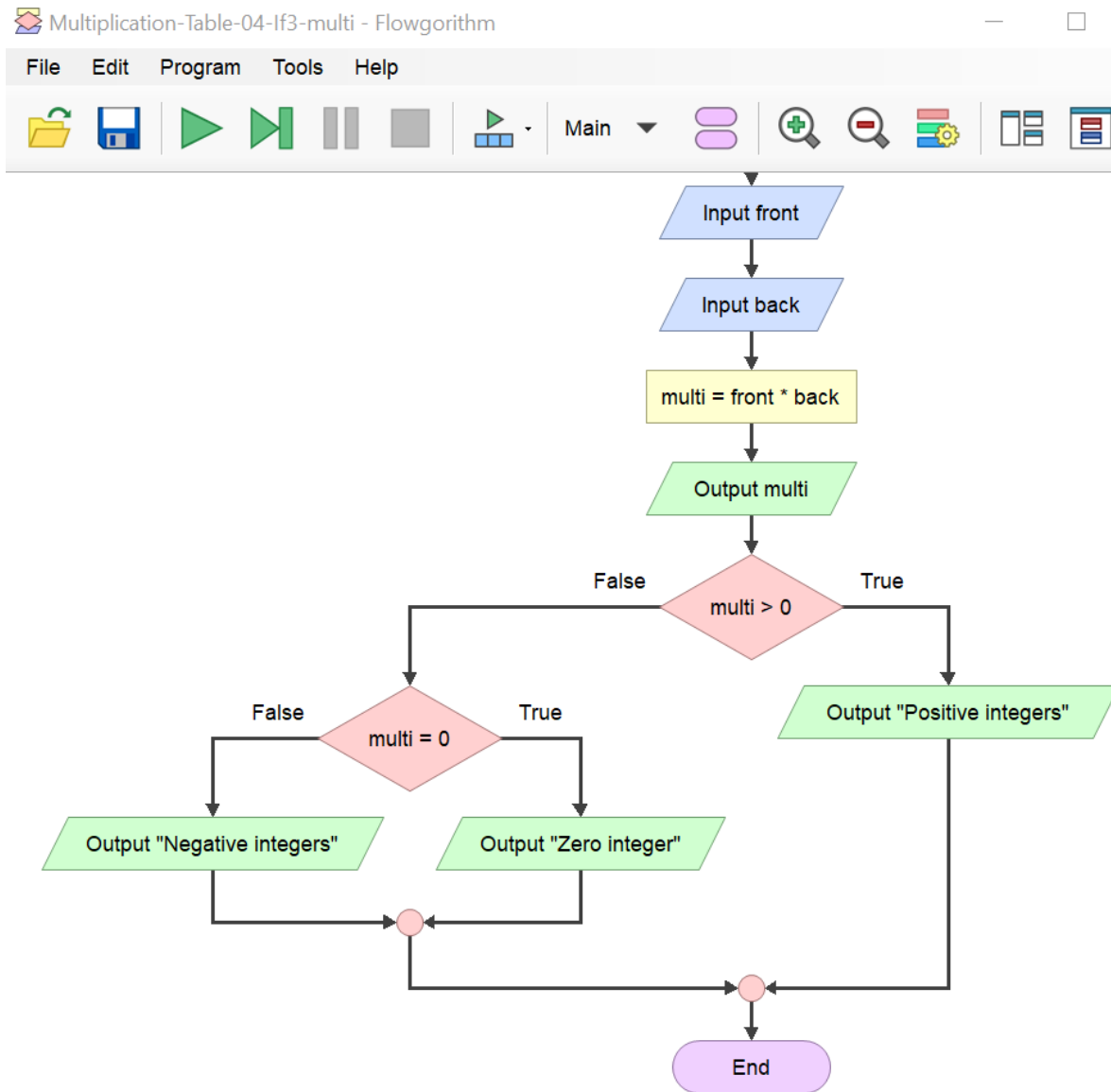
```
6
-3
-18
Is not a positive integer
>
```



โครงสร้างโปรแกรมแบบหลายทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม



โครงสร้างโปรแกรมแบบหลายทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม



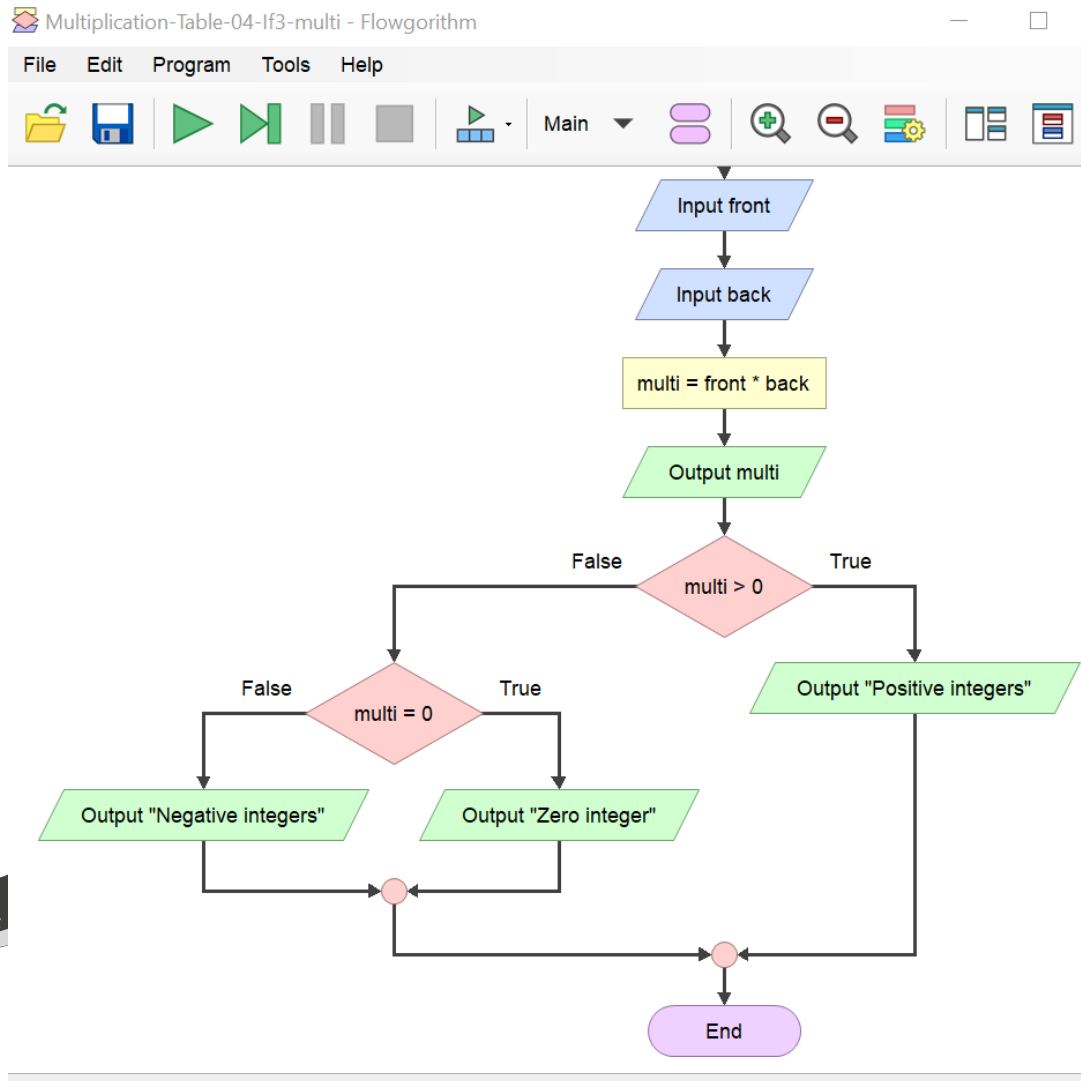
Source Code Viewer

Python

```
0 front = int(input())
1 back = int(input())
2 multi = front * back
3 print(multi)
4 if multi > 0:
5     print("Positive integers")
6 else:
7     if multi == 0:
8         print("Zero integer")
9     else:
10        print("Negative integers")
```



โครงสร้างโปรแกรมแบบหลายทางเลือกของแม่สูตรคูณ1ผลคูณและวิเคราะห์จำนวนเต็ม



Console window showing the execution of the program with inputs -4 and 5. The output is -20, which is classified as "Negative integers".

```
Input: -4  
Input: 5  
Output: -20  
Classification: Negative integers
```

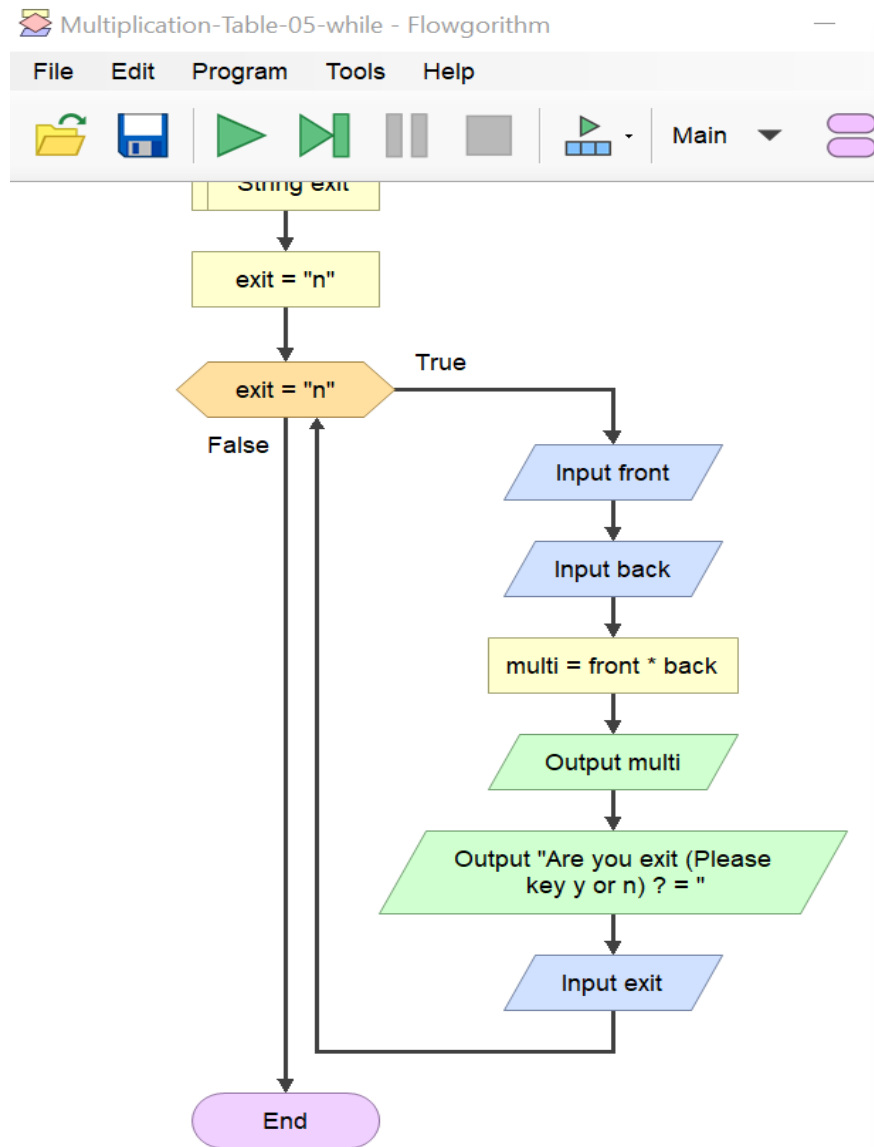
Console window showing the execution of the program with inputs 0 and 0. The output is 0, which is classified as "Zero integer".

```
Input: 0  
Input: 0  
Output: 0  
Classification: Zero integer
```

Console window showing the execution of the program with inputs 8 and 3. The output is 24, which is classified as "Positive integers".

```
Input: 8  
Input: 3  
Output: 24  
Classification: Positive integers
```


โครงสร้างโปรแกรมวนซ้ำแบบ while ของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

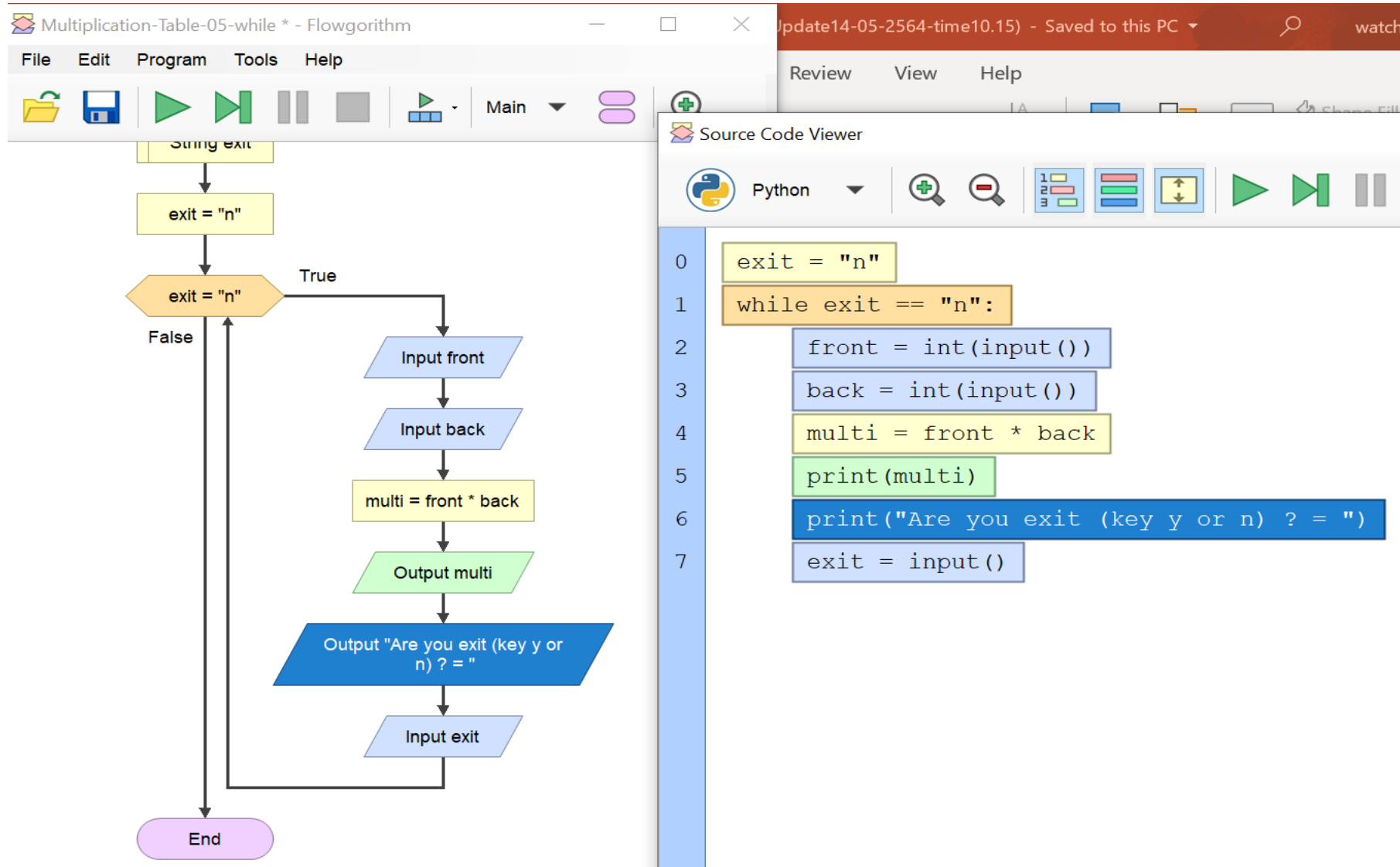


Source Code Viewer

```
0 Function Main
1   Declare Integer front
2   Declare Integer back
3   Declare Integer multi
4   Declare String exit
5
6   Assign exit = "n"
7   While exit = "n"
8     Input front
9     Input back
10    Assign multi = front * back
11    Output multi
12    Output "Are you exit (Please key y or n) ? = "
13    Input exit
14  End
15 End
```



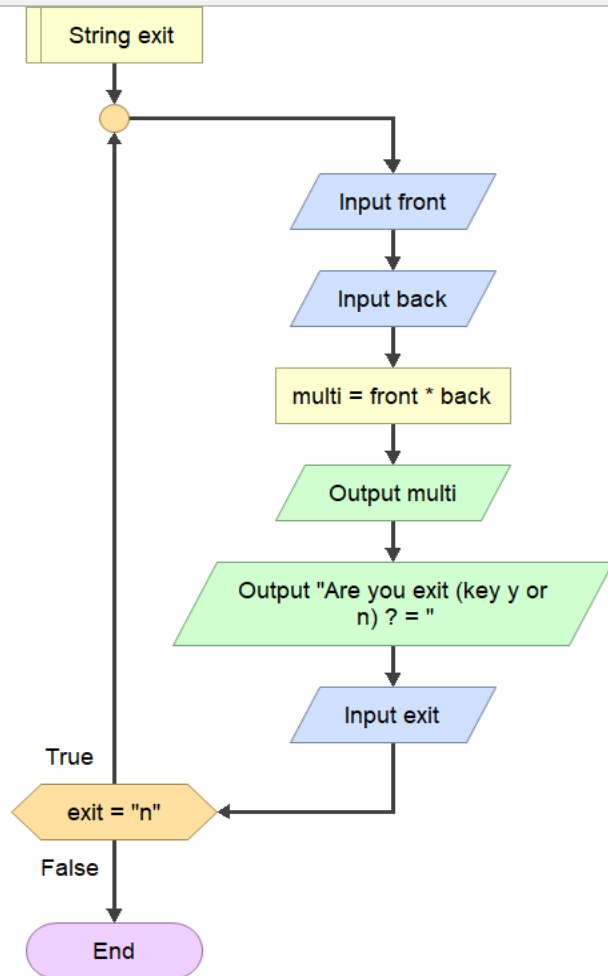
โครงสร้างโปรแกรมวนซ้ำแบบ while ของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



โครงสร้างโปรแกรมวนซ้ำแบบ do while ของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ

Multiplication-Table-06-do-while * - Flowgorithm

File Edit Program Tools Help



Source Code Viewer



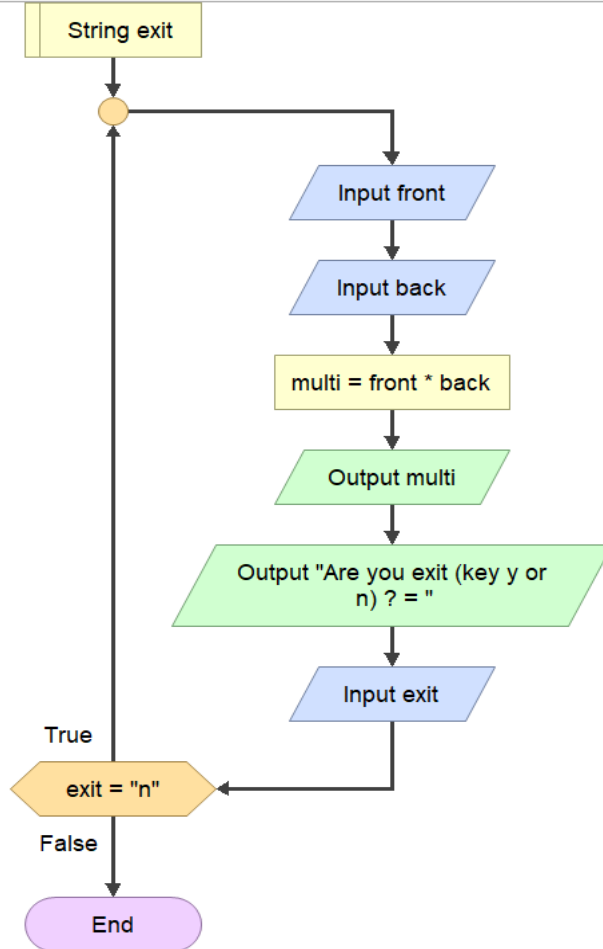
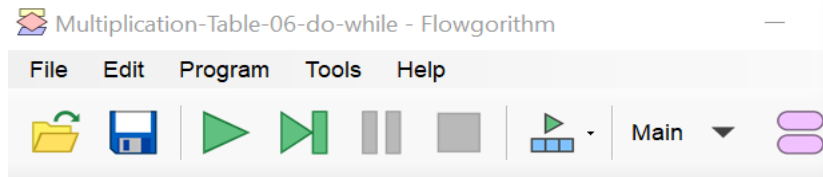
Auto Pseudocode



```
0 Function Main
1   Declare Integer front
2   Declare Integer back
3   Declare Integer multi
4   Declare String exit
5
6   Loop
7     Input front
8     Input back
9     Assign multi = front * back
10    Output multi
11    Output "Are you exit (key y or n) ? = "
12    Input exit
13    Do exit = "n"
14  End
```



โครงสร้างโปรแกรมวนซ้ำแบบ do while ของโปรแกรมหาแม่สูตรคูณ 1 ผลคูณ



Source Code Viewer

Python

```
0 while True: #This simulates a Do Loop
1     front = int(input())
2     back = int(input())
3     multi = front * back
4     print(multi)
5     print("Are you exit (key y or n) ? = ")
6     exit = input()
7     if not(exit == "n"): break #Exit loop
```



โปรแกรมหาแม่สูตรคูณ แม่ 2 หรือ 1 แม่สูตรคูณ

Console



2 X 1 = 2

2 X 2 = 4

2 X 3 = 6

2 X 4 = 8

2 X 5 = 10

2 X 6 = 12

2 X 7 = 14

2 X 8 = 16

2 X 9 = 18

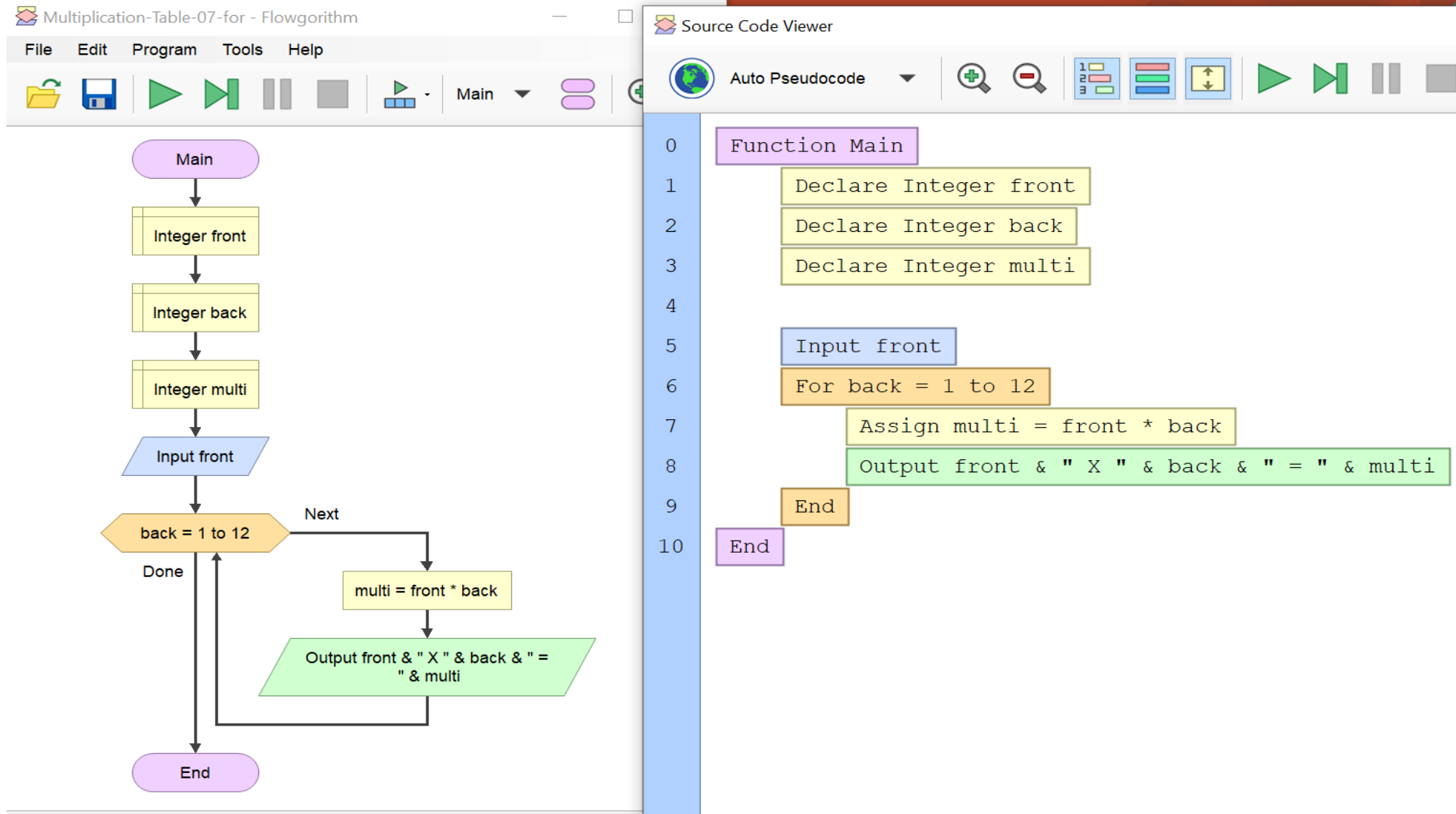
2 X 10 = 20

2 X 11 = 22

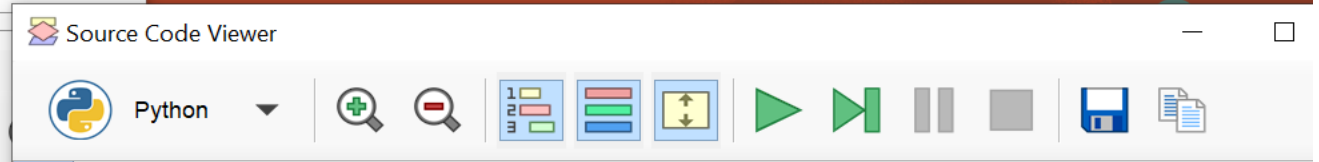
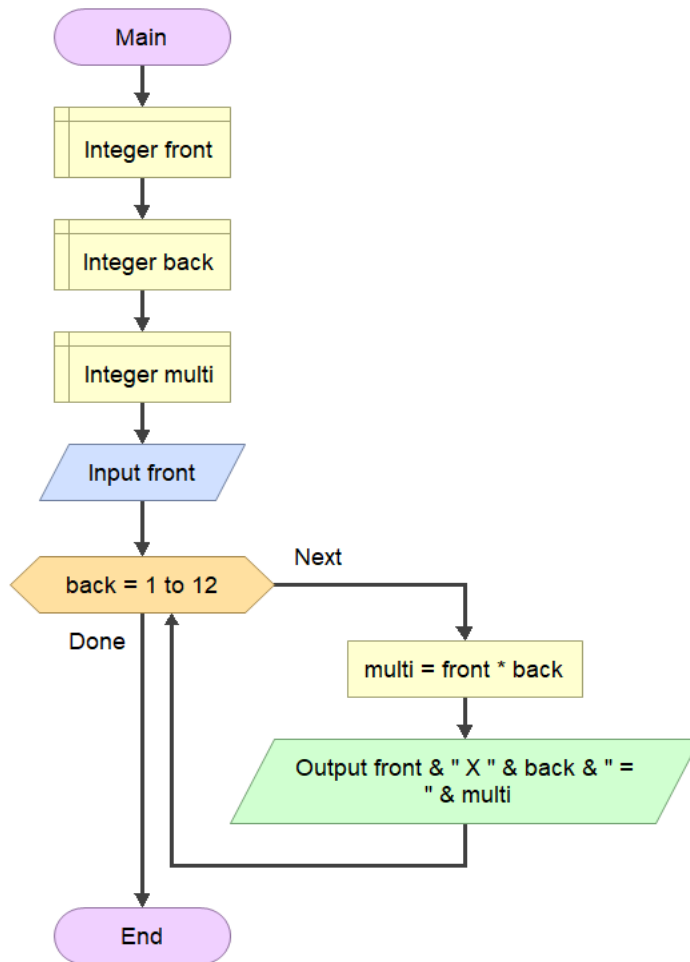
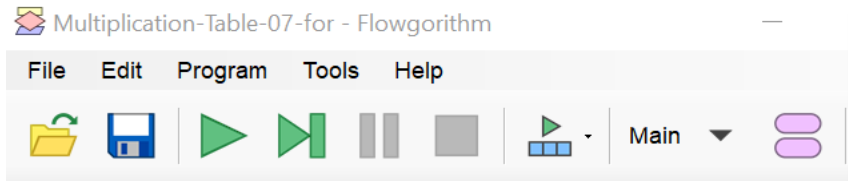
2 X 12 = 24



โครงสร้างโปรแกรมวนซ้ำแบบ for 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ



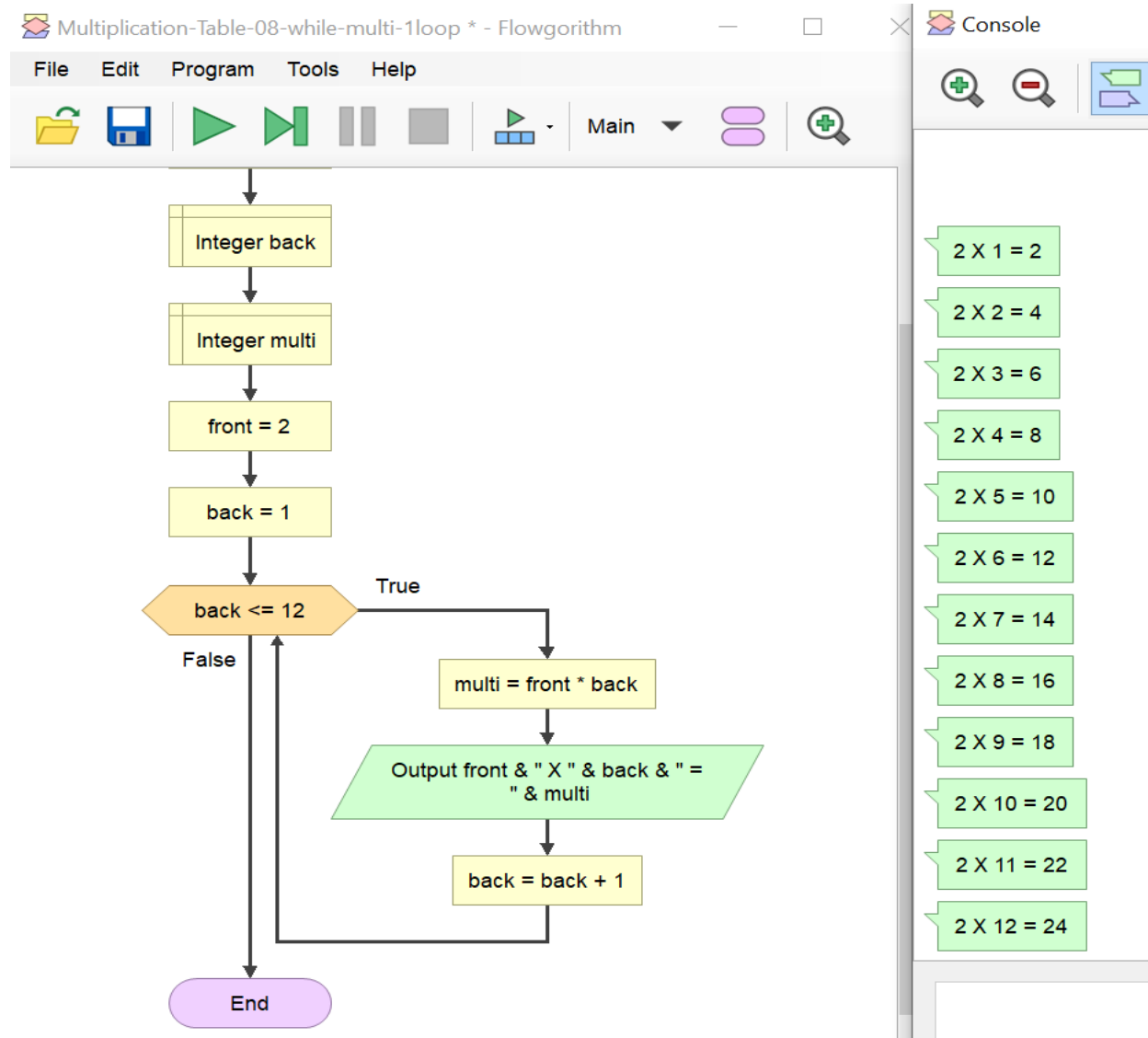
โครงสร้างโปรแกรมวนซ้ำแบบ for 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ



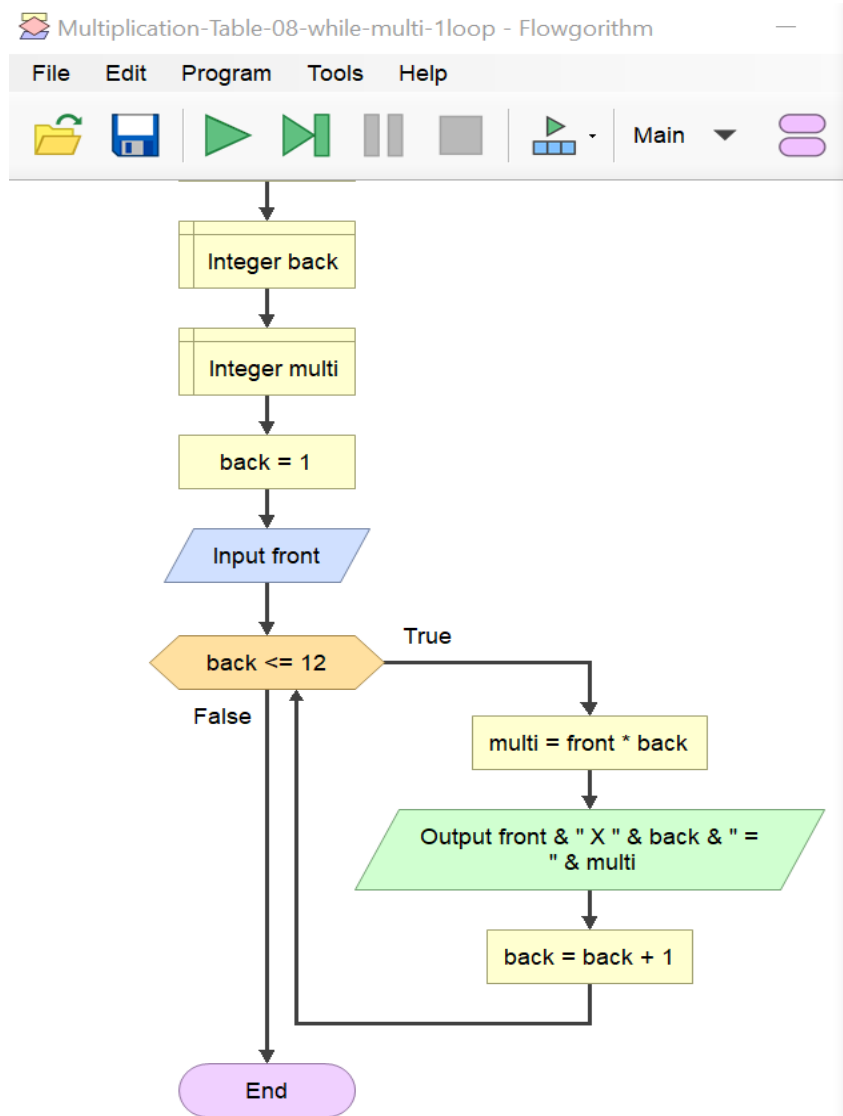
```
0 front = int(input())
1 for back in range(1, 12 + 1, 1):
2     multi = front * back
3     print(str(front) + " X " + str(back) + " = " + str(multi))
```



โครงสร้างโปรแกรมวนซ้ำแบบ for 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ



โครงสร้างโปรแกรมวนซ้ำแบบ while 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ

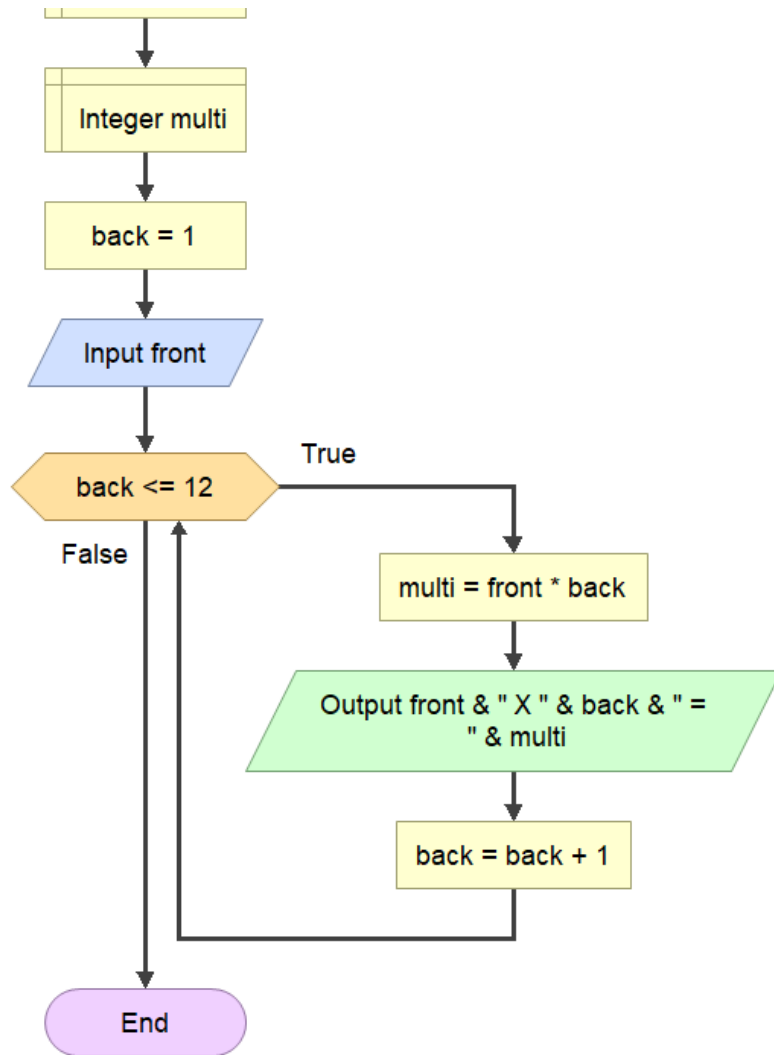


Source Code Viewer

```
0  Function Main
1  Declare Integer front
2  Declare Integer back
3  Declare Integer multi
4
5  Assign back = 1
6  Input front
7  While back <= 12
8      Assign multi = front * back
9      Output front & " X " & back & " = " & multi
10     Assign back = back + 1
11 End
12 End
```



โครงสร้างโปรแกรมวนซ้ำแบบ while 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ

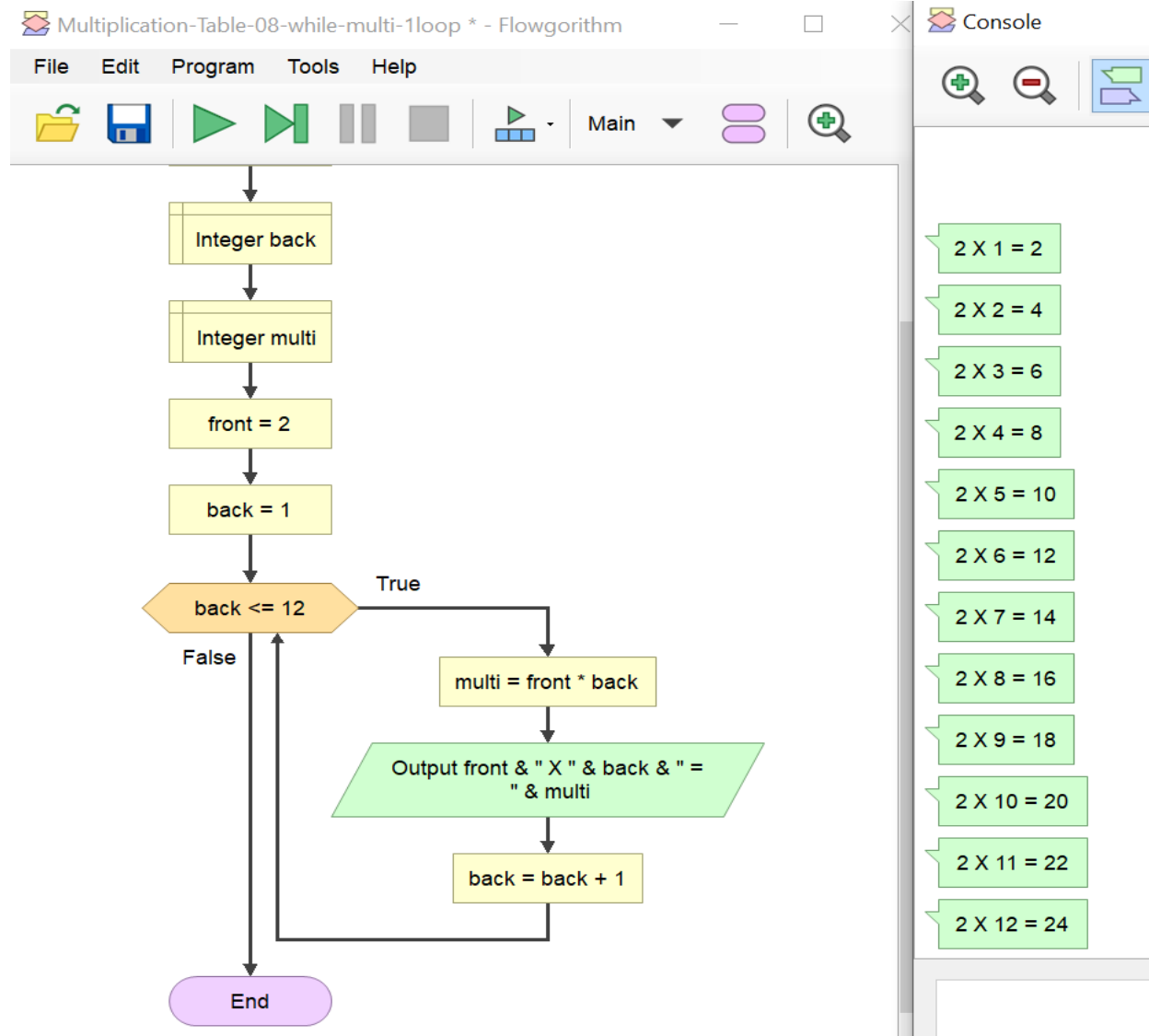


```
Source Code Viewer
Python
back = 1
front = int(input())
while back <= 12:
    multi = front * back
    print(str(front) + " X " + str(back) + " = " + str(multi))
    back = back + 1
```

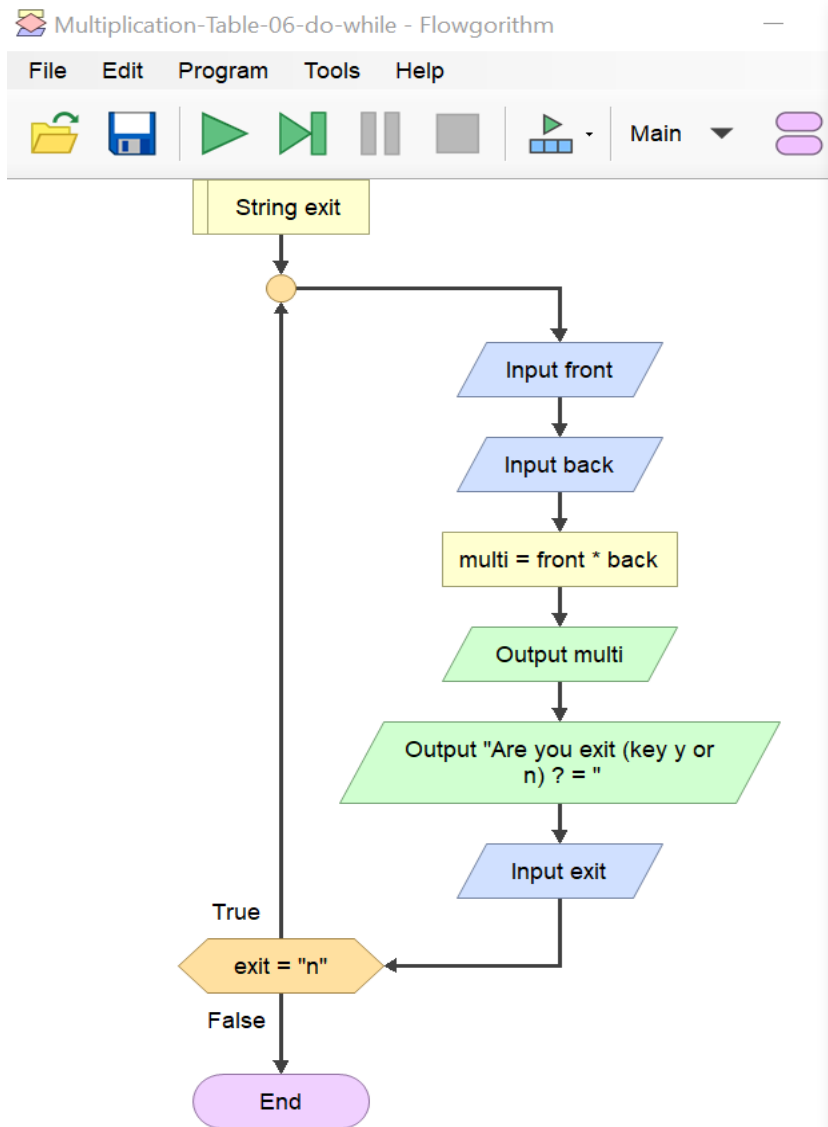
The screenshot shows a 'Source Code Viewer' window with a Python file. The code implements the logic shown in the flowchart: it initializes 'back' to 1, takes 'front' as user input, and enters a 'while back <= 12:' loop. Inside the loop, it calculates 'multi = front * back', prints the result in the format 'front X back = multi', and increments 'back' by 1. The code is displayed with line numbers 0 through 5.



โครงสร้างโปรแกรมวนซ้ำแบบ while 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ



โครงสร้างโปรแกรมวนซ้ำแบบ do while 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ



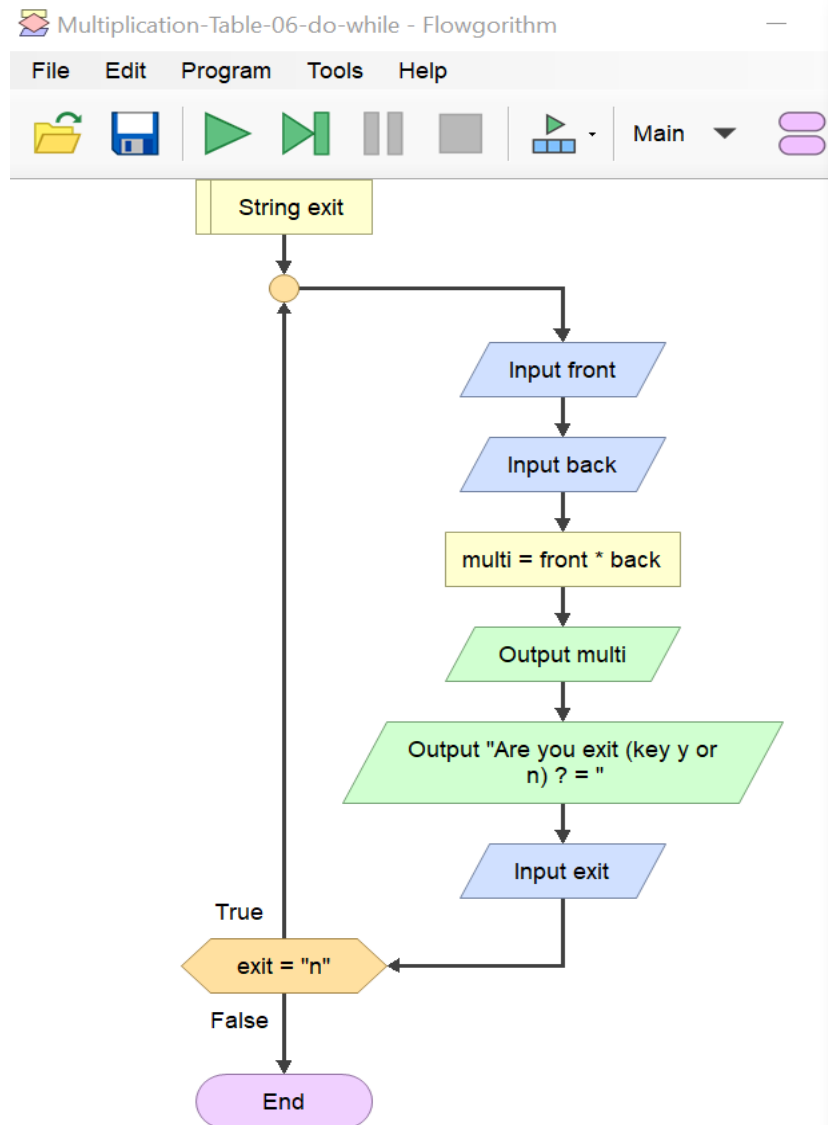
Source Code Viewer

```
Python
```

```
0 while True: #This simulates a Do Loop
1     front = int(input())
2     back = int(input())
3     multi = front * back
4     print(multi)
5     print("Are you exit (key y or n) ? = ")
6     exit = input()
7     if not(exit == "n"): break #Exit loop
```



โครงสร้างโปรแกรมวนซ้ำแบบ do while 1 รอบ ของโปรแกรมหา 1 แม่สูตรคูณ



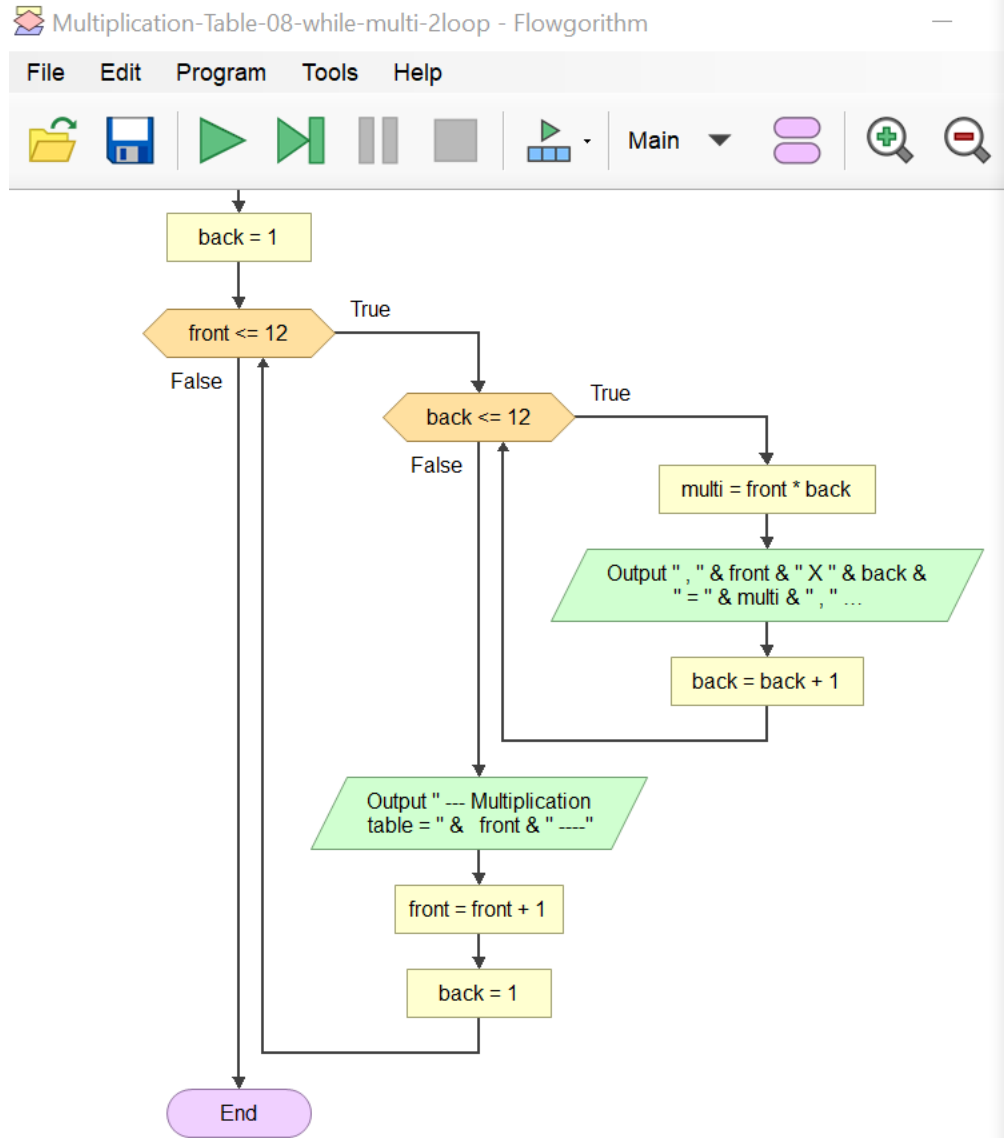
Source Code Viewer

```
Python
```

```
0 while True: #This simulates a Do Loop
1     front = int(input())
2     back = int(input())
3     multi = front * back
4     print(multi)
5     print("Are you exit (key y or n) ? = ")
6     exit = input()
7     if not(exit == "n"): break #Exit loop
```



โครงสร้างโปรแกรมวนซ้ำแบบ do while 2 รอบ ของแม่สูตรคูณ แม่ 1-12

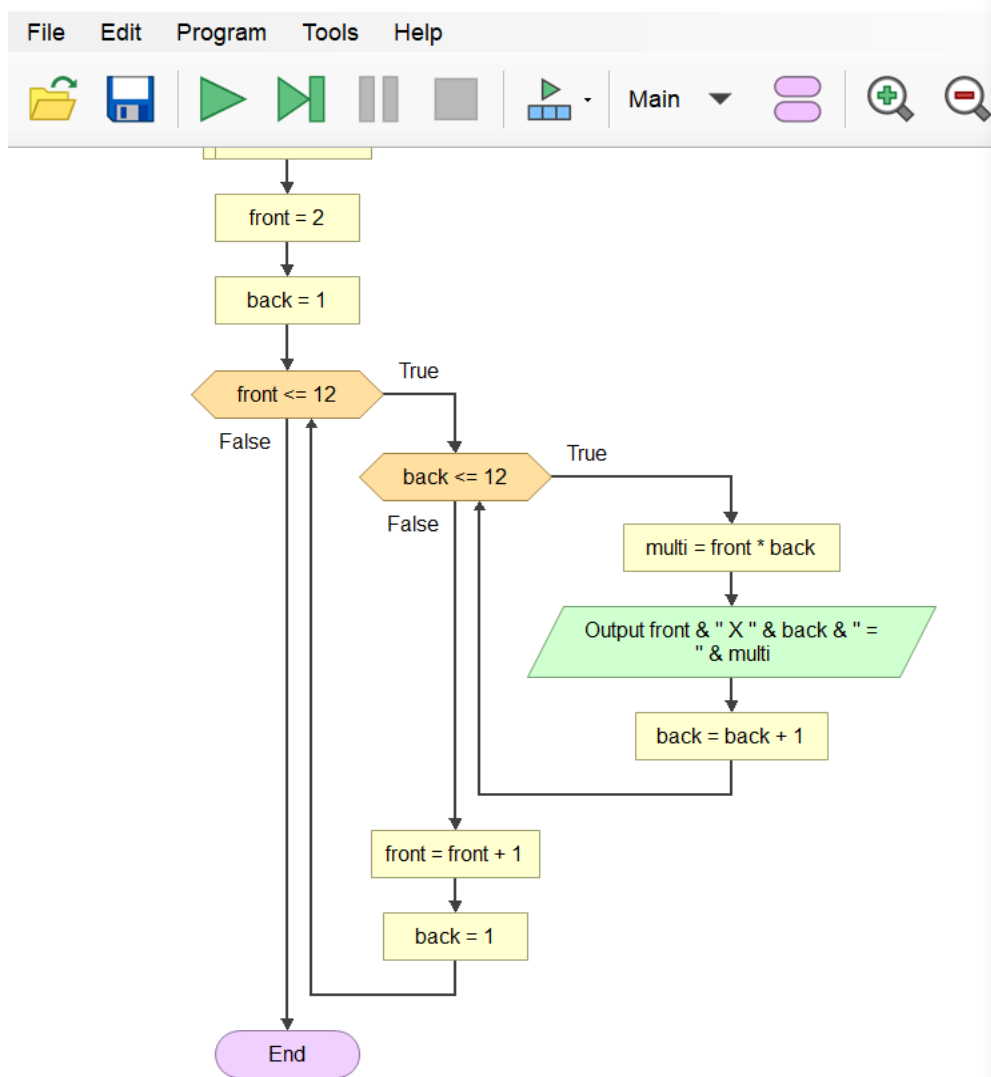


```
12 = 48 , --- Multiplication table = 4 ----  
  
, 5 X 1 = 5 , , 5 X 2 = 10 , , 5 X 3 = 15 , , 5 X 4 = 20 , , 5 X 5 = 25 , , 5 X 6 = 30 , , 5 X 7 = 35 , , 5 X 8 = 40 , , 5 X 9 = 45 , , 5 X 10 = 50 , , 5 X 11 = 55 , , 5 X 12 = 60 , --- Multiplication table = 5 ----  
  
, 6 X 1 = 6 , , 6 X 2 = 12 , , 6 X 3 = 18 , , 6 X 4 = 24 , , 6 X 5 = 30 , , 6 X 6 = 36 , , 6 X 7 = 42 , , 6 X 8 = 48 , , 6 X 9 = 54 , , 6 X 10 = 60 , , 6 X 11 = 66 , , 6 X 12 = 72 , --- Multiplication table = 6 ----  
  
, 7 X 1 = 7 , , 7 X 2 = 14 , , 7 X 3 = 21 , , 7 X 4 = 28 , , 7 X 5 = 35 , , 7 X 6 = 42 , , 7 X 7 = 49 , , 7 X 8 = 56 , , 7 X 9 = 63 , , 7 X 10 = 70 , , 7 X 11 = 77 , , 7 X 12 = 84 , --- Multiplication table = 7 ----  
  
, 8 X 1 = 8 , , 8 X 2 = 16 , , 8 X 3 = 24 , , 8 X 4 = 32 , , 8 X 5 = 40 , , 8 X 6 = 48 , , 8 X 7 = 56 , , 8 X 8 = 64 , , 8 X 9 = 72 , , 8 X 10 = 80 , , 8 X 11 = 88 , , 8 X 12 = 96 , --- Multiplication table = 8 ----  
  
, 9 X 1 = 9 , , 9 X 2 = 18 , , 9 X 3 = 27 , , 9 X 4 = 36 , , 9 X 5 = 45 , , 9 X 6 = 54 , , 9 X 7 = 63 , , 9 X 8 = 72 , , 9 X 9 = 81 , , 9 X 10 = 90 , , 9 X 11 = 99 , , 9 X 12 = 108 , --- Multiplication table = 9 ----  
  
, 10 X 1 = 10 , , 10 X 2 = 20 , , 10 X 3 = 30 , , 10 X 4 = 40 , , 10 X 5 = 50 , , 10 X 6 = 60 , , 10 X 7 = 70 , , 10 X 8 = 80 , , 10 X 9 = 90 , , 10 X 10 = 100 , , 10 X 11 = 110 , , 10 X 12 = 120 , --- Multiplication table = 10 ----  
  
, 11 X 1 = 11 , , 11 X 2 = 22 , , 11 X 3 = 33 , , 11 X 4 = 44 , , 11 X 5 = 55 , , 11 X 6 = 66 , , 11 X 7 = 77 , , 11 X 8 = 88 , , 11 X 9 = 99 , , 11 X 10 = 110 , , 11 X 11 = 121 , , 11 X 12 = 132 , --- Multiplication table = 11 ----
```



โครงสร้างโปรแกรมวนซ้ำแบบ do while 2 รอบ ของแม่สูตรคูณ แม่ 1-12

Multiplication-Table-08-while-multi-2loop - Flowgorithm



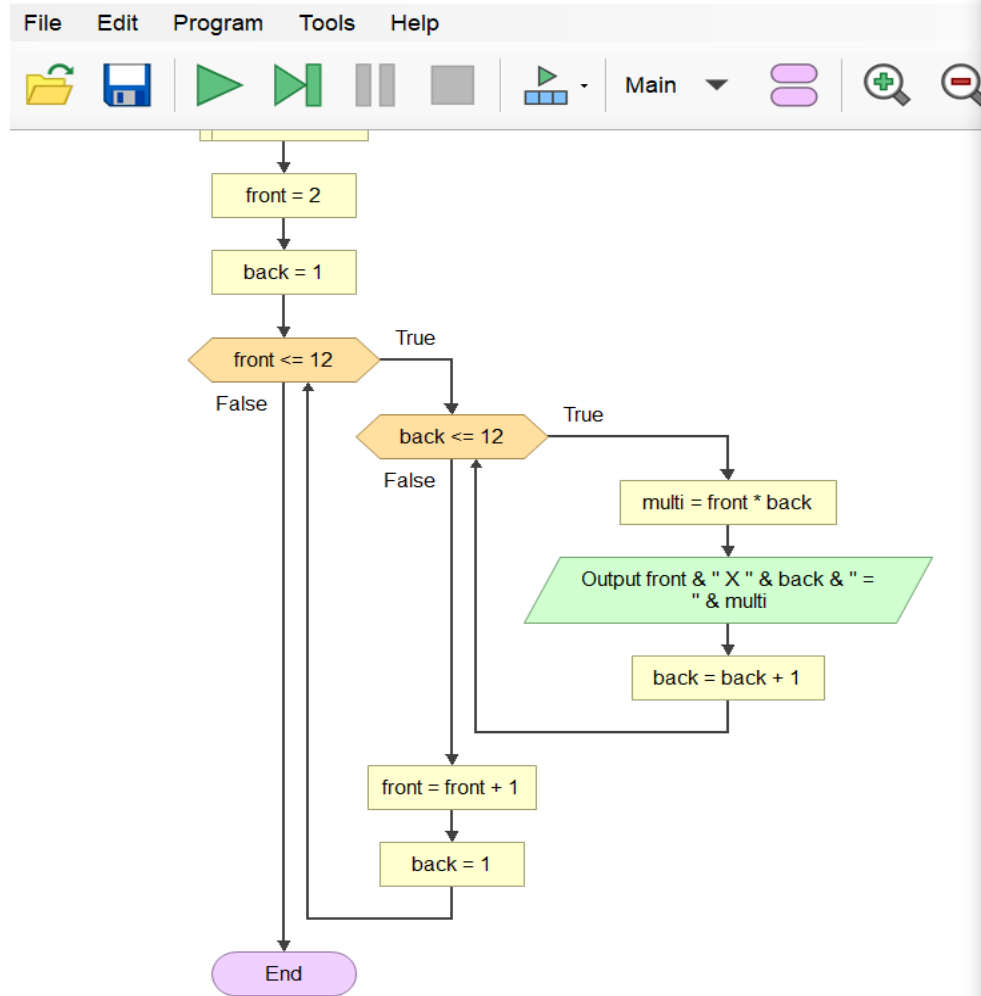
Source Code Viewer

```
0 Function Main
1   Declare Integer front
2   Declare Integer back
3   Declare Integer multi
4
5   Assign front = 2
6   Assign back = 1
7   While front <= 12
8     While back <= 12
9       Assign multi = front * back
10      Output front & " X " & back & " = " & multi
11      Assign back = back + 1
12    End
13    Assign front = front + 1
14    Assign back = 1
15  End
16 End
```



โครงสร้างโปรแกรมวนซ้ำแบบ do while 2 รอบ ของแม่สูตรคูณ แม่ 1-12

Multiplication-Table-08-while-multi-2loop - Flowgorithm

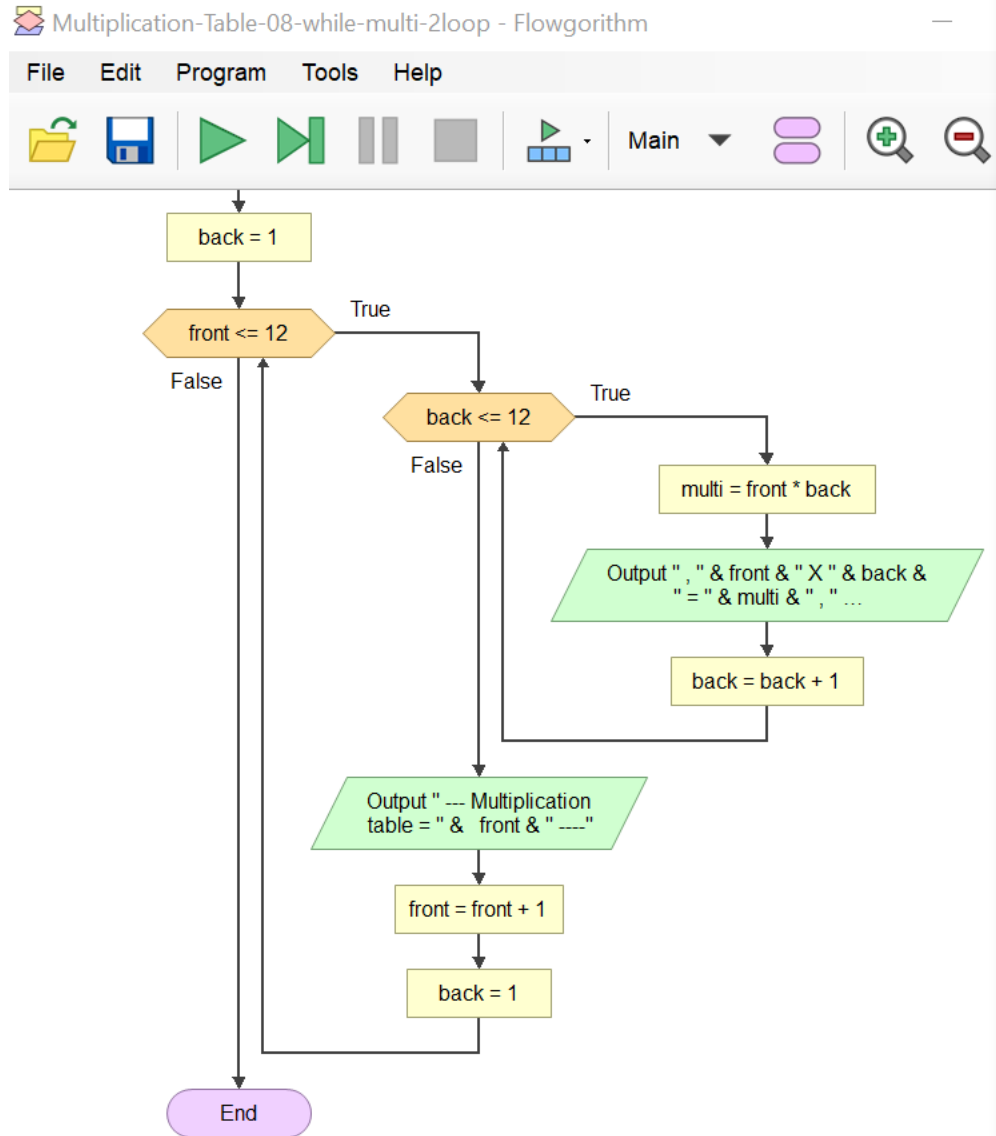


Source Code Viewer

```
Python  
0 front = 2  
1 back = 1  
2 while front <= 12:  
3     while back <= 12:  
4         multi = front * back  
5         print(str(front) + " X " + str(back) + " = " + str(multi))  
6         back = back + 1  
7     front = front + 1  
8     back = 1
```



โครงสร้างโปรแกรมวนซ้ำแบบ do while 2 รอบ ของแม่สูตรคูณ แม่ 1-12



```
12 = 48 , --- Multiplication table = 4 ----  
, 5 X 1 = 5 , , 5 X 2 = 10 , , 5 X 3 = 15 , , 5 X 4 = 20 , , 5 X 5 = 25 , , 5 X 6 = 30 , , 5 X 7 = 35 , , 5 X 8 = 40 , , 5 X 9 = 45 , , 5 X 10 = 50 , , 5 X 11 = 55 , , 5 X 12 = 60 , --- Multiplication table = 5 ----  
, 6 X 1 = 6 , , 6 X 2 = 12 , , 6 X 3 = 18 , , 6 X 4 = 24 , , 6 X 5 = 30 , , 6 X 6 = 36 , , 6 X 7 = 42 , , 6 X 8 = 48 , , 6 X 9 = 54 , , 6 X 10 = 60 , , 6 X 11 = 66 , , 6 X 12 = 72 , --- Multiplication table = 6 ----  
, 7 X 1 = 7 , , 7 X 2 = 14 , , 7 X 3 = 21 , , 7 X 4 = 28 , , 7 X 5 = 35 , , 7 X 6 = 42 , , 7 X 7 = 49 , , 7 X 8 = 56 , , 7 X 9 = 63 , , 7 X 10 = 70 , , 7 X 11 = 77 , , 7 X 12 = 84 , --- Multiplication table = 7 ----  
, 8 X 1 = 8 , , 8 X 2 = 16 , , 8 X 3 = 24 , , 8 X 4 = 32 , , 8 X 5 = 40 , , 8 X 6 = 48 , , 8 X 7 = 56 , , 8 X 8 = 64 , , 8 X 9 = 72 , , 8 X 10 = 80 , , 8 X 11 = 88 , , 8 X 12 = 96 , --- Multiplication table = 8 ----  
, 9 X 1 = 9 , , 9 X 2 = 18 , , 9 X 3 = 27 , , 9 X 4 = 36 , , 9 X 5 = 45 , , 9 X 6 = 54 , , 9 X 7 = 63 , , 9 X 8 = 72 , , 9 X 9 = 81 , , 9 X 10 = 90 , , 9 X 11 = 99 , , 9 X 12 = 108 , --- Multiplication table = 9 ----  
, 10 X 1 = 10 , , 10 X 2 = 20 , , 10 X 3 = 30 , , 10 X 4 = 40 , , 10 X 5 = 50 , , 10 X 6 = 60 , , 10 X 7 = 70 , , 10 X 8 = 80 , , 10 X 9 = 90 , , 10 X 10 = 100 , , 10 X 11 = 110 , , 10 X 12 = 120 , --- Multiplication table = 10 ----  
, 11 X 1 = 11 , , 11 X 2 = 22 , , 11 X 3 = 33 , , 11 X 4 = 44 , , 11 X 5 = 55 , , 11 X 6 = 66 , , 11 X 7 = 77 , , 11 X 8 = 88 , , 11 X 9 = 99 , , 11 X 10 = 110 , , 11 X 11 = 121 , , 11 X 12 = 132 , --- Multiplication table = 11 ----
```



เนื้อหาเพิ่มเติมที่เกี่ยวข้องกับการโปรแกรม

- การแปลงรหัสจำลองเป็นผังงานทางออนไลน์อัตโนมัติด้วยแอป code2flow (<https://app.code2flow.com>)
- การแปลง Coding เป็นผังงานด้วยโปรแกรม Visustin v8
(Flow chart generator)
- การเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System)
ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

การแปลงรหัสจำลองเป็นผังงานทางออนไลน์อัตโนมัติด้วยแอป code2flow (<https://app.code2flow.com>)

The screenshot shows the code2flow web application interface. On the left is a code editor with the following code:

```
1 Welcome to code2flow;  
2 if(In |doubt?) {  
3   Press Help;  
4   while(!Ready?)  
5     Read help;  
6 }  
7 // the flowchart  
8 // updates  
9 // as you type  
10 Improve your  
11 workflow!;
```

On the right is a flowchart visualization of the code. The flowchart starts with an oval 'Welcome to code2flow', followed by a diamond decision 'In doubt?'. If 'True', it goes to a rectangle 'Press Help', then to another diamond decision 'Ready?'. If 'Ready?' is 'True', it goes to an oval 'Improve your workflow!'. If 'Ready?' is 'False', it goes to a rectangle 'Read help', which loops back to 'Press Help'. If 'In doubt?' is 'False', it goes directly to 'Improve your workflow!'. A callout box points to the 'Improve your workflow!' node with the text 'the flowchart updates as you type'.

Copyright © 2013-2021, Code Charm, Inc.



การแปลง Coding เป็นผังงานด้วยโปรแกรม Visustin v8 (Flow chart generator)

The screenshot displays the Visustin v8 interface. On the left, the 'Code' tab shows the following C++ function:

```
double average(double sum, int count) {  
    /* Calculate average from sum and count */  
  
    if (count == 1)  
        return sum;  
    else if (count > 0)  
        return sum / count;  
    else  
        return 0; /* Invalid value of Count */  
}
```

On the right, the 'Structure' tab shows a flowchart that visualizes the code's logic. The flowchart starts with an oval containing the function signature 'double average(double sum, int count)'. It then proceeds to a process box 'Calculate average from sum and count'. A decision diamond asks 'count == 1?'. If 'Yes', it goes to a process box 'return sum;'. If 'No', it goes to another decision diamond 'count > 0?'. From 'count > 0?', a 'Yes' path leads to 'return sum / count;', and a 'No' path leads to 'return 0;'. A label 'Invalid value of Count' points to the 'No' path. All paths eventually lead to an 'End' terminal box.

```
graph TD  
    Start([double average(double sum, int count)]) --> Process[Calculate average from sum and count]  
    Process --> Dec1{count == 1?}  
    Dec1 -- Yes --> P1[return sum;]  
    Dec1 -- No --> Dec2{count > 0?}  
    Dec2 -- Yes --> P2[return sum / count;]  
    Dec2 -- No --> P3[return 0;]  
    P1 --> End[/End/]  
    P2 --> End  
    P3 --> End
```



การเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System)

ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

. CodeRunner is a free open-source question-type plug-in for Moodle that can run program code submitted by students in answer to a wide range of programming questions in many different languages. It is intended primarily for use in computer programming courses although it can be used to grade any question for which the answer is text. It is normally used in Moodle's adaptive quiz mode; students paste in their code in answer to each programming question and get to see their test-case results immediately. They can then correct their code and resubmit, typically for a small penalty. URL : <https://coderunner.org.nz>



Question 1
Correct
Mark 1.00 out of 1.00
Flag question

Write a Python3 function `sqr(n)` that returns the square of its numeric parameter `n`.

For example:

Test	Result
<code>print(sqr(-3))</code>	9
<code>print(sqr(11))</code>	121

Answer:

```
1 def sqr(n):  
2     return n * n
```

Check

Test	Expected	Got	
<code>print(sqr(-3))</code>	9	9	✓
<code>print(sqr(11))</code>	121	121	✓
<code>print(sqr(-4))</code>	16	16	✓
<code>print(sqr(0))</code>	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

การเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System)

ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

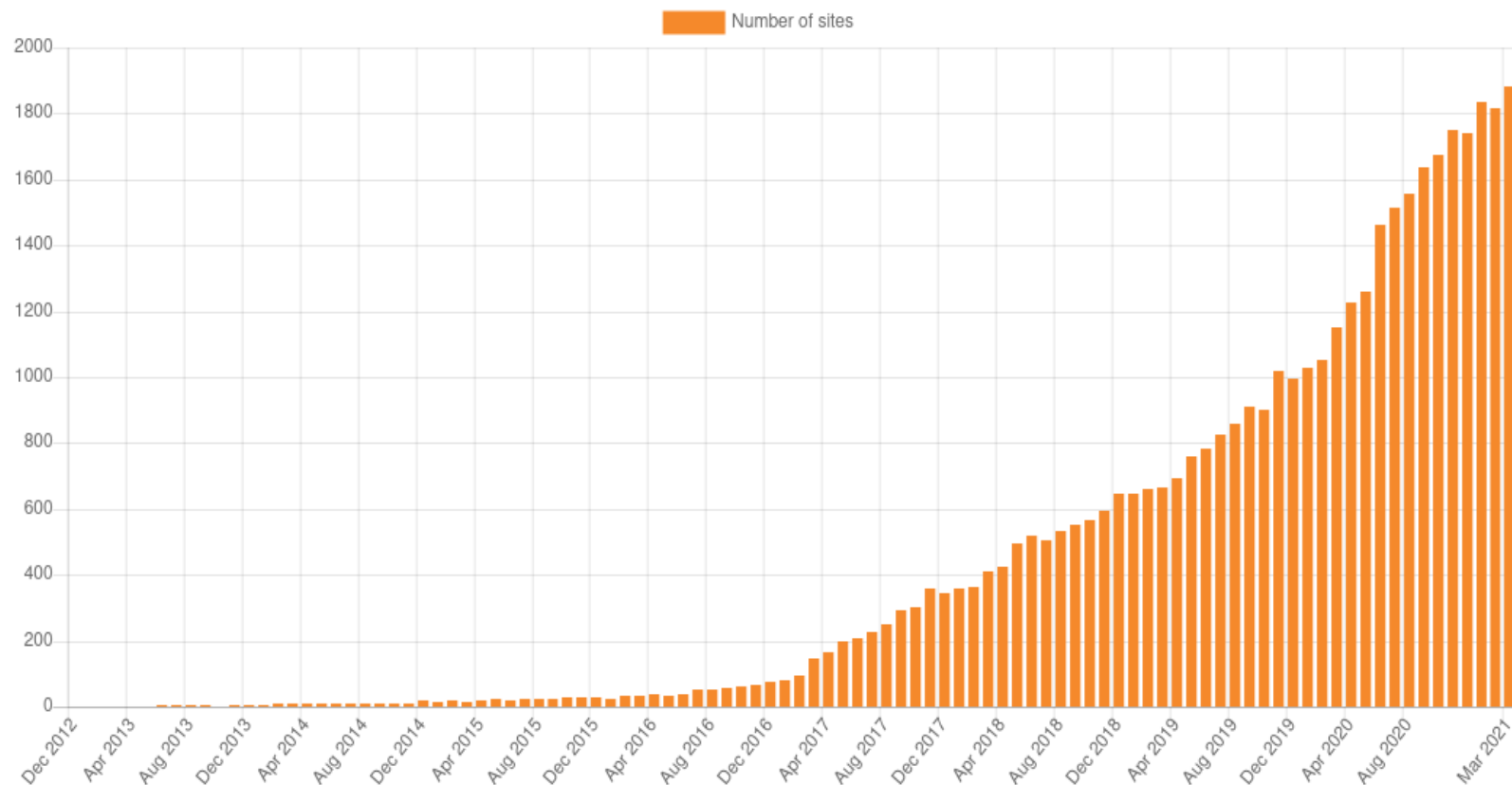
CodeRunner is currently used on over 1800 registered Moodle sites worldwide, as shown in the following chart from the Moodle plugin repository:

<https://coderunner.org.nz>



Usage stats

Number of sites using the plugin: 1884



การเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System)

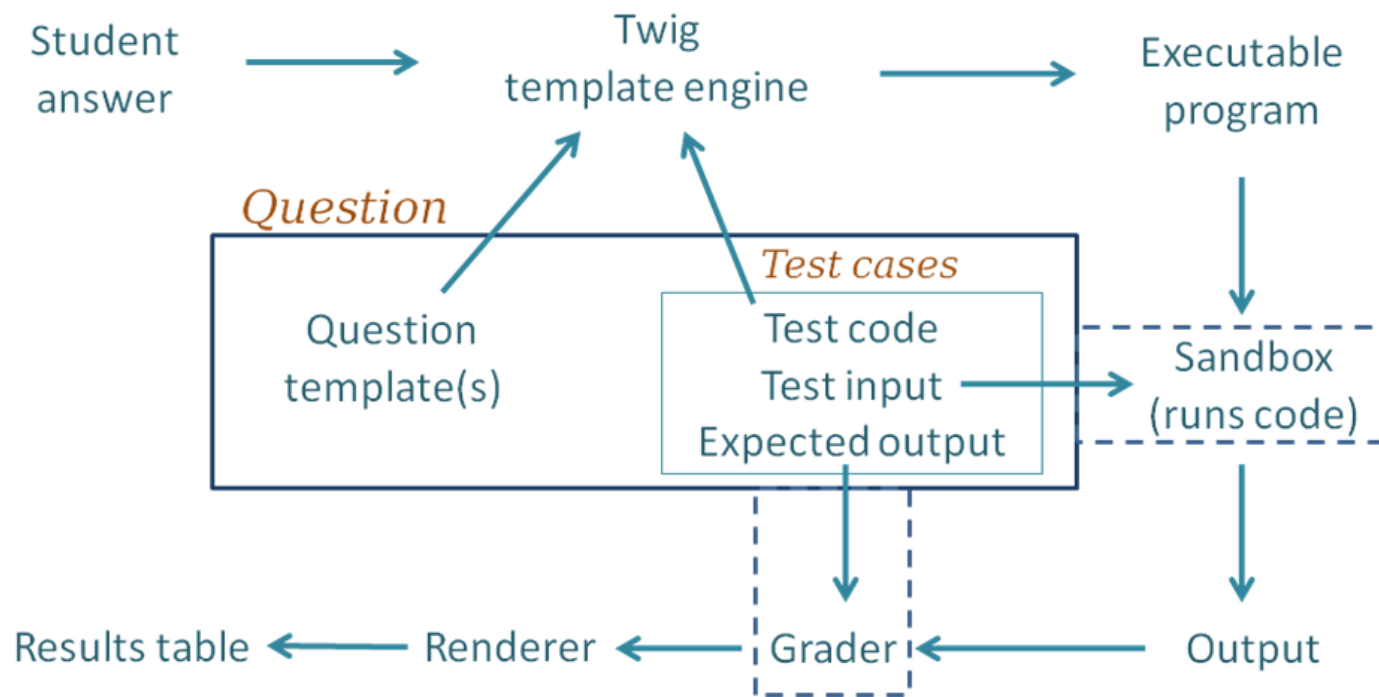
ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

The Architecture of CodeRunner

Although it's straightforward to write simple questions using the built-in question types, anything more advanced than that requires an understanding of how CodeRunner works.

The block diagram below shows the components of CodeRunner and the path taken as a student submission is graded.

https://moodle.org/plugins/qtype_coderunner/3.1.0/12971



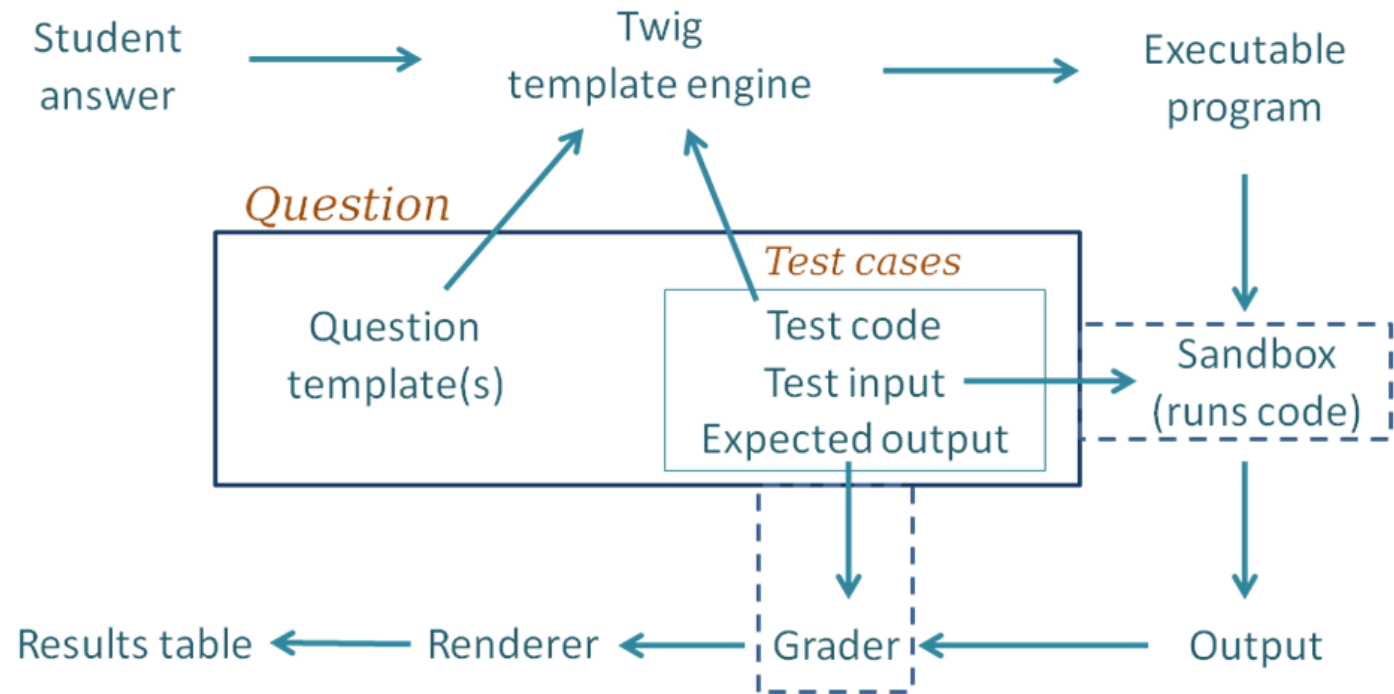
การเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System)

ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

The Architecture of CodeRunner

The Python quiz is recommended to give you a feel for CodeRunner's capabilities, even if you don't understand Python3; the questions could (for the most part) be written in any of the supported Languages (Python2, Python3, C, C++, Java, PHP, JavaScript, Octave).

https://moodle.org/plugins/qtype_coderunner/3.1.0/12971



การเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

Question 1

Correct

Mark 5.00 out of 5.00

Paste the 5 lines of output from your simulator run into the answer box below. You get 1 mark for each correct line.

Answer:

```
Thing: 7
Number: 23
Blah: 11
Twaddle this is
Last line
```

Check

	Expected	Got	Comment	Mark	
✓	Thing: 7 Number: 23 Blah: 11 Twaddle this is Last line	Thing: 7 Number: 23 Blah: 11 Twaddle this is Last line	Line 0 right Line 1 right Line 2 right Line 3 right Line 4 right	5	✓

Passed all tests! ✓

Correct

Marks for this submission: 5.00/5.00.

Question 1

Partially correct

Mark 5.00 out of 5.00

Paste the 5 lines of output from your simulator run into the answer box below. You get 1 mark for each correct line.

Answer:

```
Thing: 77
Number: 23
Blah: 11
Twaddle this is
Lost line
```

Check

	Expected	Got	Comment	Mark	
✓	Thing: 7 Number: 23 Blah: 11 Twaddle this is Last line	Thing: 77 Number: 23 Blah: 11 Twaddle this is Lost line	Line 0 wrong Line 1 right Line 2 right Line 3 right Line 4 wrong	3	✓

Partially correct

Marks for this submission: 3.00/5.00.



ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

เข้าสู่เว็บไซต์วิทยาการคอมพิวเตอร์สำหรับโรงเรียนตากพิทยาคม <http://tps.comsci.info>



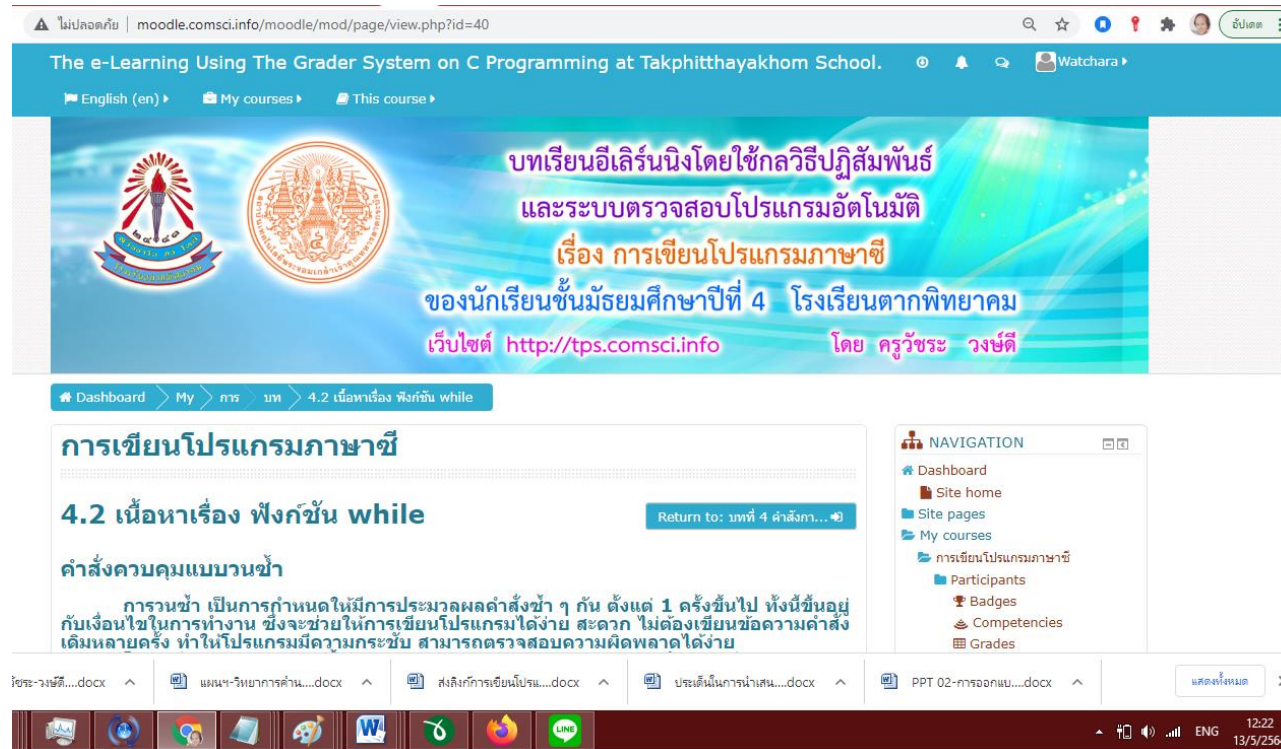
(Computer Science for Takphitthayakhom School)
สนับสนุนโดย www.comsci.info



NEW -11/January/2018 : 11 ม.ค. 2561 : เข้าสู่ระบบบทเรียนอีเลิร์นนิ่งโดยใช้กลยุทธ์สัมพันธ์และระบบตรวจสอบโปรแกรมอัตโนมัติ เรื่อง การเขียนโปรแกรมภาษาซี ของนักเรียนชั้นมัธยมศึกษาปีที่ 4 โรงเรียนตากพิทยาคม **คลิกที่ลิงก์นี้** <http://moodle.comsci.info/moodle/enrol/index.php?id=3> (<-- The development of e-Learning Using Transactional Strategies Instruction and the Automatic Applications Grader System on C Language Programming for Grade 10 Students at Takphitthayakhom School.)



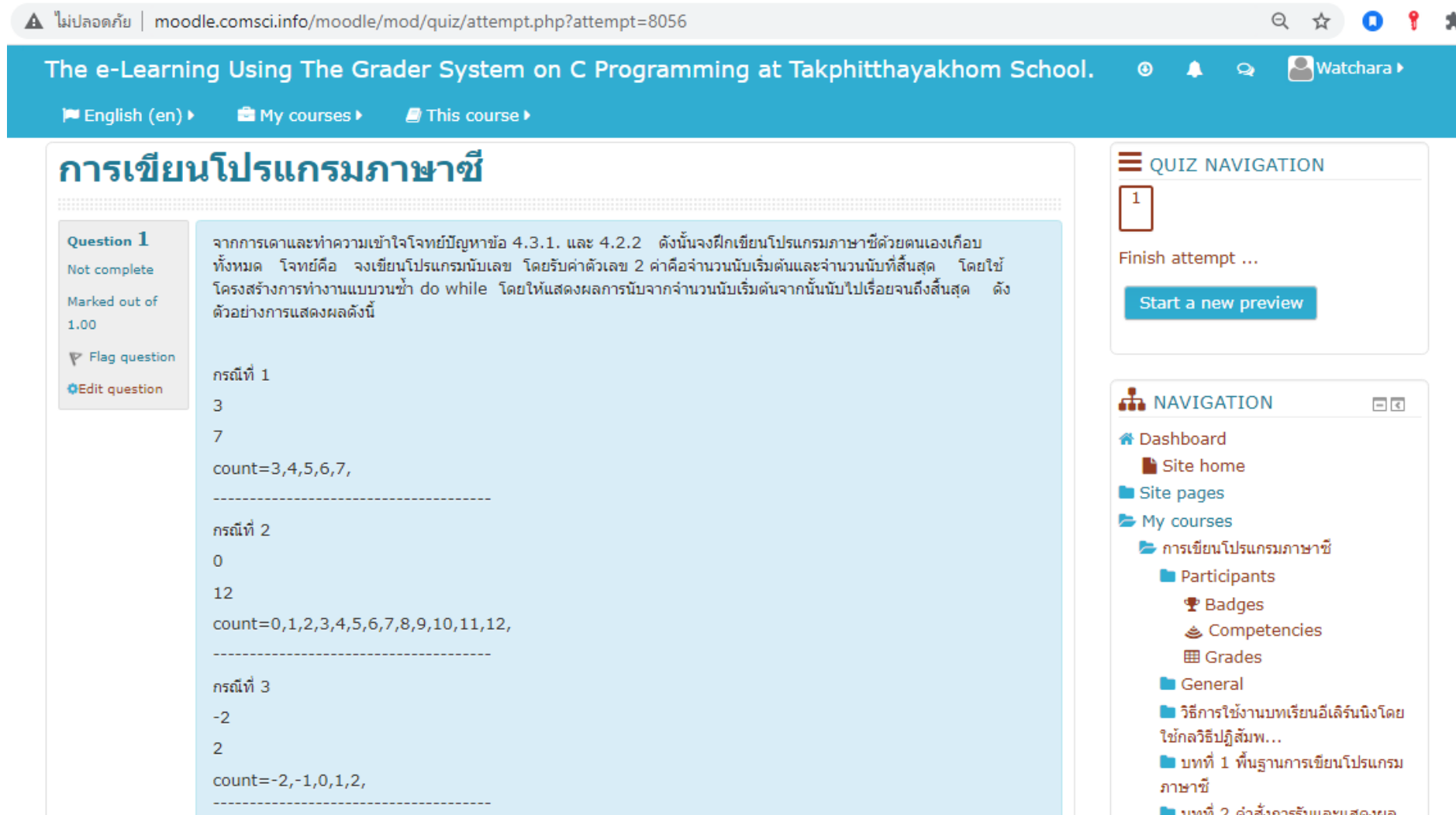
ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner



หลังจากนั้นเข้าสู่ระบบบทเรียนนี้เรียนรู้โดยใช้กลวิธีปฏิสัมพันธ์และระบบตรวจสอบโปรแกรมอัตโนมัติ (ด้วยระบบ CodeRunner ของ Moodle) ที่เว็บไซต์ <http://moodle.comsci.info/moodle/enrol/index.php?id=3>
ตัวอย่าง Username คือ y61-tps-grb-c44-n47 Password คือ 12345678aA#



ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner



The screenshot shows a Moodle quiz interface. The browser address bar is `moodle.comsci.info/moodle/mod/quiz/attempt.php?attempt=8056`. The page title is "The e-Learning Using The Grader System on C Programming at Takphitthayakhom School." The main content area is titled "การเขียนโปรแกรมภาษาซี" (C Programming). The question is "Question 1" and is marked out of 1.00. The question text is in Thai: "จากการเดาและทำความเข้าใจโจทย์ปัญหาข้อ 4.3.1. และ 4.2.2 ดังนั้นจงฝึกเขียนโปรแกรมภาษาซีด้วยตนเองเกือบทั้งหมด โจทย์คือ จงเขียนโปรแกรมนับเลข โดยรับค่าตัวเลข 2 ค่าคือจำนวนนับเริ่มต้นและจำนวนนับที่สิ้นสุด โดยใช้โครงสร้างการทำงานแบบวนซ้ำ do while โดยให้แสดงผลการนับจากจำนวนนับเริ่มต้นจากนั้นนับไปเรื่อยๆจนถึงสิ้นสุด ดังตัวอย่างการแสดงผลดังนี้". The question has three parts, each with a code snippet and its output:

```
กรณีที่ 1
3
7
count=3,4,5,6,7,
-----

กรณีที่ 2
0
12
count=0,1,2,3,4,5,6,7,8,9,10,11,12,
-----

กรณีที่ 3
-2
2
count=-2,-1,0,1,2,
-----
```

On the right side, there is a "QUIZ NAVIGATION" section with a "1" in a box, "Finish attempt ..." text, and a "Start a new preview" button. Below that is a "NAVIGATION" section with a tree view of the site structure, including "Dashboard", "Site home", "Site pages", "My courses", and "การเขียนโปรแกรมภาษาซี" with sub-items like "Participants", "Badges", "Competencies", "Grades", "General", and "วิธีการใช้งานบทเรียนอีเลิร์นนิ่งโดยใช้กลวิธีปฏิสัมพันธ์...".



ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

The e-Learning Using The Grader System on C Programming at Takphitthayakhom School. Watchara

English (en) My courses This course

Answer: (penalty regime: 10, 20, ... %)

```
1 #include<stdio.h>
2 int main()
3 {
4     int n , end;
5     scanf("%d", &n );
6     .....
7     printf("count=");
8     do {
9         // ให้นักเรียนเขียนโปรแกรมเองทั้งหมดลงในด้านล่างนี้
10        // Please programming in this below.
11
12
13
14 }
```

Precheck Check

Finish attempt ...

Return to: บทที่ 4 คำสั่งกา... ↩

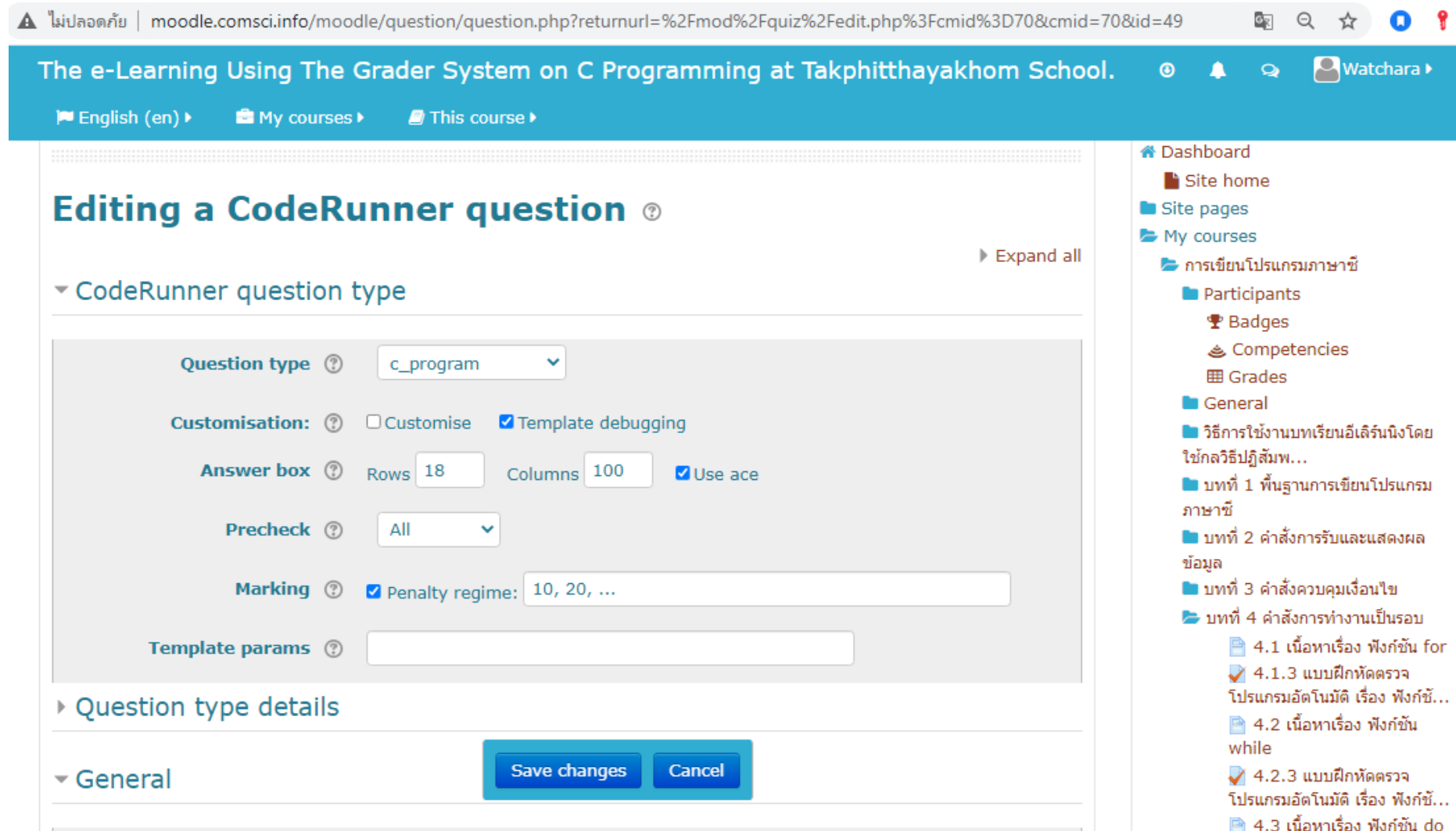
ข้อมูล

- บทที่ 3 คำสั่งควบคุมเงื่อนไข
- บทที่ 4 คำสั่งการทำงานเป็นรอบ
 - 4.1 เนื้อหาเรื่อง ฟังก์ชัน for
 - 4.1.3 แบบฝึกหัดตรวจโปรแกรมอัตโนมัติ เรื่อง ฟังก์ชั...
 - 4.2 เนื้อหาเรื่อง ฟังก์ชัน while
 - 4.2.3 แบบฝึกหัดตรวจโปรแกรมอัตโนมัติ เรื่อง ฟังก์ชั...
 - 4.3 เนื้อหาเรื่อง ฟังก์ชัน do while
 - 4.3.3 แบบฝึกหัดตรวจโปรแกรมอัตโนมัติ เรื่อง ฟังก์ชั...
 - 4.4 แบบทดสอบท้ายบทที่ 4 คำสั่งการทำงานเป็นรอบ
 - 4.4.6 ***แบบทดสอบท้ายบทที่ 4 เรื่อง คำสั่งการทำงาน...
- เนื้อหาการเขียนโปรแกรมภาษาซีเพิ่มเติม
- แบบทดสอบหลังเรียน 40 ข้อ 100 คะแนน 90 นาที
- C Language Programming



ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System)

ในบทเรียนออนไลน์ด้วย Moodle CodeRunner



The screenshot shows the Moodle interface for editing a CodeRunner question. The browser address bar displays the URL: moodle.comsci.info/moodle/question/question.php?returnurl=%2Fmod%2Fquiz%2Fedit.php%3Fcmid%3D70&cmid=70&id=49. The page title is "The e-Learning Using The Grader System on C Programming at Takphitthayakhom School." The user is logged in as "Watchara".

The main content area is titled "Editing a CodeRunner question". The "CodeRunner question type" section is expanded, showing the following settings:

- Question type:** c_program
- Customisation:** Customise Template debugging
- Answer box:** Rows: 18, Columns: 100, Use ace
- Precheck:** All
- Marking:** Penalty regime: 10, 20, ...
- Template params:** (empty field)

At the bottom of the form, there are "Save changes" and "Cancel" buttons. The right sidebar shows a navigation menu with items like "Dashboard", "Site home", "Site pages", "My courses", and a list of course sections under "การเขียนโปรแกรมภาษาซี".



ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner

▼ Test cases

Test case 1 ? case1

Standard Input ? 3
7

Expected output ? count=3,4,5,6,7,

Extra template data ?

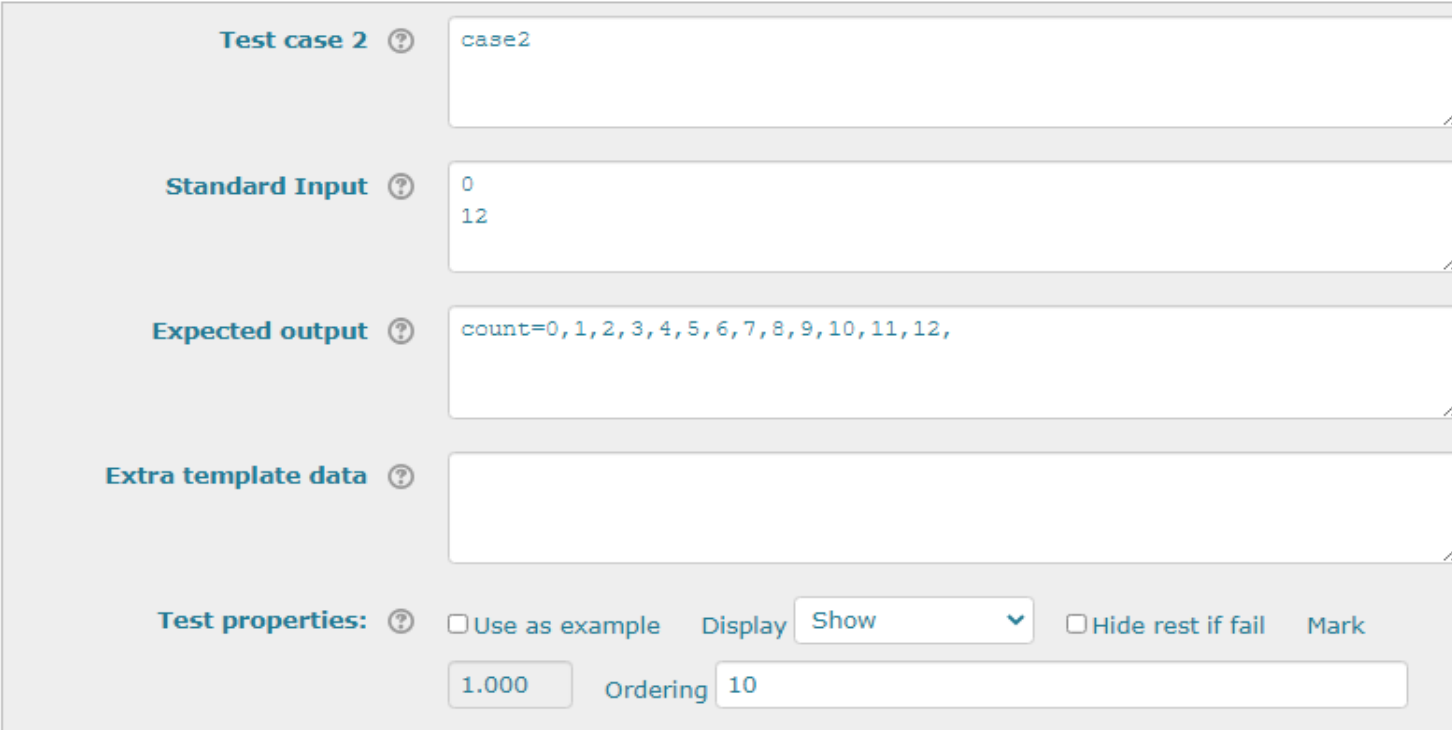
Test properties: ? Use as example Display Show ▼ Hide rest if fail Mark

1.000 Ordering 0

การวัดผลประเมินผลกรณีที่ 1 ถ้าระบบ CodeRunner ใส่เลข 3 และ 7 และดำเนินการ
อัตโนมัติ สามารถแสดงผลการนับเลข 3 – 7 ได้ถูกต้อง ระบบจะเพิ่ม 1 คะแนนโดยอัตโนมัติ



ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner



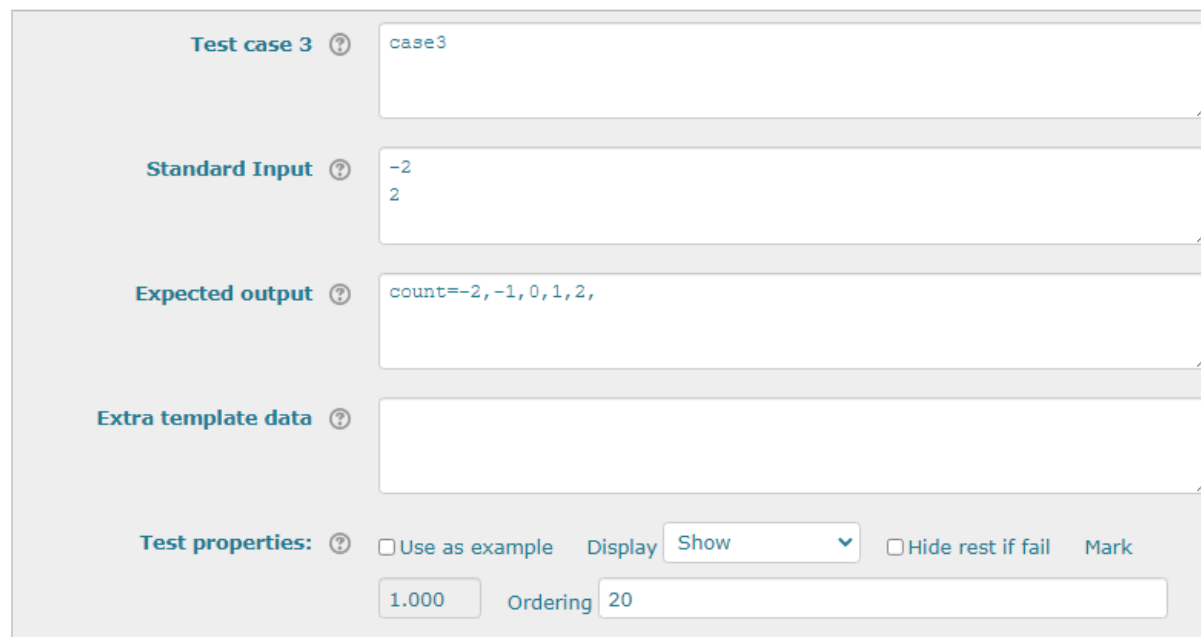
The screenshot shows the Moodle CodeRunner configuration interface for a test case. It includes the following fields and options:

- Test case 2**: A text input field containing "case2".
- Standard Input**: A text input field containing "0" and "12" on separate lines.
- Expected output**: A text input field containing "count=0,1,2,3,4,5,6,7,8,9,10,11,12,".
- Extra template data**: An empty text input field.
- Test properties:** A section containing:
 - Use as example
 - Display: (dropdown menu)
 - Hide rest if fail
 - Mark:
 - Ordering:

. การวัดผลประเมินผลกรณีที่ 2 ถ้าระบบ CodeRunner ใส่เลข 0 และ 12 และดำเนินการอัตโนมัติ สามารถแสดงผลการนับเลข 0 – 12 ได้ถูกต้อง ระบบจะเพิ่ม 1 คะแนนโดยอัตโนมัติ



ตัวอย่างการเพิ่มระบบตรวจสอบโปรแกรมอัตโนมัติ (Grader System) ในบทเรียนออนไลน์ด้วย Moodle CodeRunner



The screenshot shows the Moodle CodeRunner configuration interface for a test case. It includes the following fields and options:

- Test case 3**: case3
- Standard Input**: -2
2
- Expected output**: count=-2,-1,0,1,2,
- Extra template data**: (empty)
- Test properties**:
 - Use as example
 - Display: Show (dropdown)
 - Hide rest if fail
 - Mark: 1.000
 - Ordering: 20

. การวัดผลประเมินผลกรณีที่ 3 ถ้าระบบ CodeRunner ใส่เลข -2 และ 2 และดำเนินการอัตโนมัติ สามารถแสดงผลการนับเลข -2 ถึงเลข 2 ได้ถูกต้อง ระบบจะเพิ่ม 1 คะแนนโดยอัตโนมัติ

