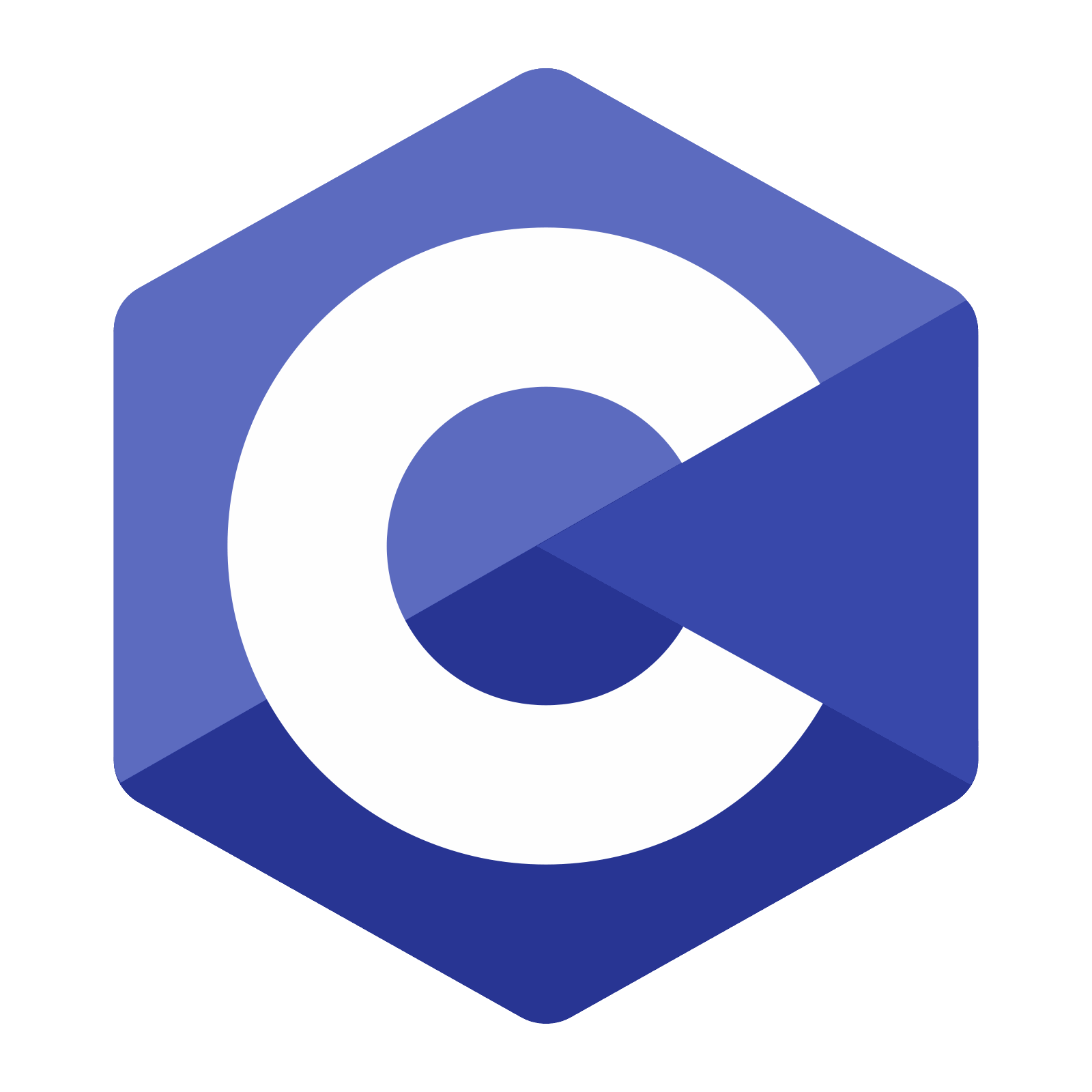
 ผังงาน Flowchart เป็นผังงานที่แสดงให้เห็นถึงแนวคิด และขั้นตอนการทำงานของโปรแกรม  
อีกทั้งยังช่วยให้มองเห็นภาพรวมของโปรแกรมทำให้เขียนโปรแกรมได้ง่ายขึ้น การเขียน Flowchart นั้นจะใช้สัญลักษณ์ต่าง ๆ แทนคำอธิบายและกระบวนการทำงานของโปรแกรมในแต่ละส่วนตั้งแต่เริ่มแรก จนถึงสิ้นสุดการทำงานของโปรแกรม เพื่อให้ผู้พัฒนาโปรแกรมได้เข้าใจแนวคิดและการทำงานที่ชัดเจนมากที่สุด

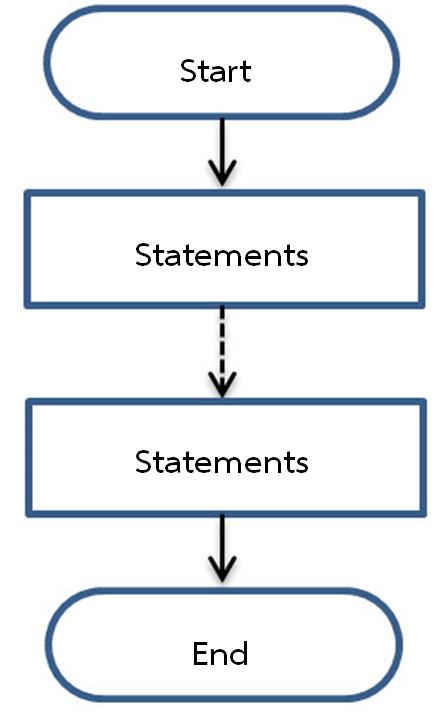
|  |
| --- |
| ในการเขียนผังงาน Flowchart นั้นได้มีการกำหนดสัญลักษณ์ที่ใช้งานให้เป็นสากล และเป็นมาตรฐานเดียวกัน โดยการกำหนดมาตรฐานนี้ได้ถูกกำหนดตามแบบของ ANSI (American National Standards Institute) และ ISO (International Standard Organization) เพื่อสื่อความหมาย และให้เกิดความเข้าใจในสัญลักษณ์ที่ใช้ในการเขียนผังงาน Flowchart ทั่วโลก โดยสัญลักษณ์ต่าง ๆ นั้น สามารถแบ่งออกได้เป็นหลายสัญลักษณ์ตามรายละเอียดดังตารางที่ 1 ——————————————————————————————————————————— **ตารางที่ 1 สัญลักษณ์ผังงาน Flowchart** 1 |
| |  |  | | --- | --- | | สัญลักษณ์ | ความหมายและการใช้งาน | | https://www.yupparaj.ac.th/thanphisit/bot2/1/1.jpg Terminal | ใช้เริ่มต้นและสิ้นสุดการทำงานของโปรแกรม | | https://www.yupparaj.ac.th/thanphisit/bot2/1/2.jpg Process | ใช้ประมวลผลการทำงานของโปรแกรม | | https://www.yupparaj.ac.th/thanphisit/bot2/1/3.jpg Decision | เพื่อกำหนดทางเลือก หรือตัดสินใจการทำงานของโปรแกรม | | https://www.yupparaj.ac.th/thanphisit/bot2/1/4.jpg Document | ใช้เพื่อแสดงผลลัพธ์ออกทางเครื่องพิมพ์ (Printer) | | https://www.yupparaj.ac.th/thanphisit/bot2/1/5.jpg Input | ใช้เพื่อรับหรือแสดงผลโดยไม่กำหนดชนิด ของอุปกรณ์ในการทำงาน | | https://www.yupparaj.ac.th/thanphisit/bot2/1/6.jpg Manual Input | ใช้เพื่อรับข้อมูลเข้ามาในระบบ  เช่น ค่าของตัวแปร | | https://www.yupparaj.ac.th/thanphisit/bot2/1/7.jpg Display | ใช้เพื่อแสดงผลลัพธ์ออกทางจอภาพ (Monitor) | | https://www.yupparaj.ac.th/thanphisit/bot2/1/8.jpg On-Page Reference | ใช้เพื่อเชื่อมต่อการทำงานจากจุดหนึ่ง ไปยังอีกจุดหนึ่ง | | https://www.yupparaj.ac.th/thanphisit/bot2/1/9.jpg Off-Page Reference | ใช้เพื่อเชื่อมต่อผังงานที่มีความยาวมากกว่า  1 หน้า ให้สามารถเชื่อมโยงกันได้ | | https://www.yupparaj.ac.th/thanphisit/bot2/1/10.jpg Connector | ใช้เชื่อมต่อระหว่างสัญลักษณ์หนึ่งไปยังอีกสัญลักษณ์หนึ่ง เพื่อให้ทราบการไหลและทิศทาง ของข้อมูล | |

|  |
| --- |
| ประเภทของผังงาน Flowchart นั้น สามารถแบ่งออกได้เป็น 2 ประเภท คือ  **1. ผังงานระบบ (System Flowchart)** นั้นจะแสดงให้เห็นถึง ขั้นตอนการทำงานในระบบหนึ่ง ๆ แบบเป็นภาพกว้างของระบบ โดยไม่ได้เจาะลึกให้เห็นว่าแต่ละระบบย่อยนั้นมีการทำงานอย่างไร ส่วนใหญ่แล้ว จะแสดงให้เห็นเพียงจุดเริ่มต้นการทำงาน ลักษณะการทำงานระบบใหญ่ ๆ ว่ามีการประมวลผลอย่างไร แล้ว จะทำให้เกิดผลลัพธ์อย่างไร และข้อมูลนั้นถูกจัดเก็บไว้ที่ไหน ดังภาพที่ 1 |
| https://www.yupparaj.ac.th/thanphisit/bot2/2-44.jpg ภาพที่ 1 ตัวอย่างผังงานระบบ (System Flowchart) |
| **2. ผังงานโปรแกรม (Program Flowchart)** นั้นจะแสดงให้เห็นถึงลำดับขั้นตอนในการทำงานของโปรแกรมตั้งแต่การรับข้อมูล การประมวลผล รวมถึงผลลัพธ์ของการทำงานในแต่ละขั้นตอนจึงทำให้นักพัฒนาโปรแกรมคอมพิวเตอร์สามารถนาเอาขั้นตอนดังกล่าวไปเขียนโปรแกรมได้สะดวกมากขึ้น ดังภาพที่ 2 |
| https://www.yupparaj.ac.th/thanphisit/bot2/2-444.jpg |
| ภาพที่ 2 ตัวอย่างของผังงานโปรแกรม (Program Flowchart) |

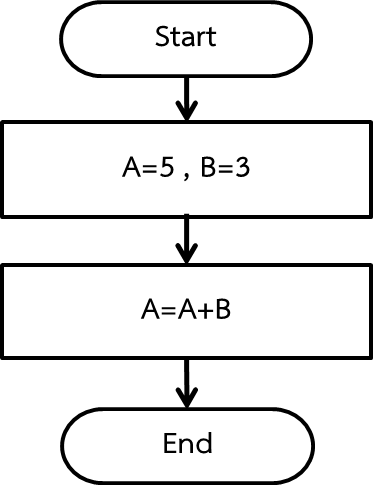
หลักการเขียนผังงาน Flowchart นั้น จะใช้สัญลักษณ์ต่าง ๆ แทนคำอธิบาย แนวคิด และ ขั้นตอน  
การทำงานของโปรแกรม โดยจะเขียนเป็นลำดับการทำงานแบบบนลงล่าง (Top-To-Down) และใช้ลูกศรเชื่อมโยงการทำงานจากสัญลักษณ์หนึ่งไปยังอีกสัญลักษณ์หนึ่ง และภายในสัญลักษณ์ต่าง ๆ นั้นก็ต้องเขียน  
คำอธิบายสั้น ๆ กำกับไว้ด้วย ตั้งแต่ขั้นตอนการทำงานเริ่มแรก จนถึงขั้นตอนการทำงานสุดท้าย เพื่อให้เห็นลำดับการทำงานของโปรแกรมทั้งหมดที่ถูกต้อง ก่อนที่จะเริ่มเขียน Flowchart นั้น ควรที่จะทำการศึกษาถึงสัญลักษณ์ต่าง ๆ ให้เข้าใจก่อน เพื่อให้ทราบถึงหลักการเขียนผังงาน Flowchart ที่ถูกต้อง ขออธิบายหลัก  
การเขียน Flowchart ดังนี้  
  
        1. เลือกใช้สัญลักษณ์เพื่อสื่อความหมายให้ถูกต้อง เช่น การกำหนดค่าตัวแปรให้เลือกใช้กรอบสี่เหลี่ยมผืนผ้า การตรวจสอบเงื่อนไขเพื่อตัดสินใจด้วยคำสั่ง if ให้เลือกใช้สี่เหลี่ยมข้าวหลามตัด เป็นต้น   
        2. ใช้ลูกศรกำหนดการทำงานของโปรแกรมจากบนลงล่าง จากซ้ายไปขวา โดยเรียงลำดับ  
การทำงานของคำสั่ง ไม่ควรสลับการทำงานกัน   
        3. ผังงานทุกผังงานนั้นต้องมีลูกศรเข้าและออก ยกเว้นจุดเริ่มต้นและจุดสิ้นสุดการทำงานเท่านั้น   
        4. ลูกศรทุกตัวต้องออกจากผังงานและชี้ที่ผังงานเสมอ   
        5. คำอธิบายภายในผังงานควรสั้น ๆ และเข้าใจง่าย   
        6. ไม่ควรใช้ลูกศรชี้ไกลเกินไป หากจำเป็นให้ใช้การเชื่อมต่อแทน

การเขียนโปรแกรมด้วยภาษาซี เป็นการเขียนโปรแกรมแบบมีโครงสร้าง (Stucture Programming) จะประกอบด้วยโครงสร้างพื้นฐาน 3 รูปแบบต่อไปนี้คือ

1. การทำงานแบบตามลำดับ (Sequence)   
      เป็นรูปแบบการเขียนโปรแกรมที่ง่ายที่สุด และไม่มีความซับซ้อน มีลำดับการทำงานจากบนลงล่าง   
มีการทำงานทีละคำสั่งจนจบการทำงาน ซึ่งมีรูปแบบการทำงานดังรูปที่ 1

****  
รูปที่ 1 การทำงานแบบตามลำดับ (Sequence)

โครงสร้างลักษณะนี้เป็นโครงสร้างพื้นฐานของผังงาน และเป็นลักษณะขั้นตอนการทำงานที่พบมากที่สุด คือ ทำงานทีละขั้นตอนตามลำดับ   
——————————————————————————————————————————  
**ตัวอย่างที่ 1 การเขียนและอธิบายการทำงานของ Flowchart แบบตามลำดับดังนี้**

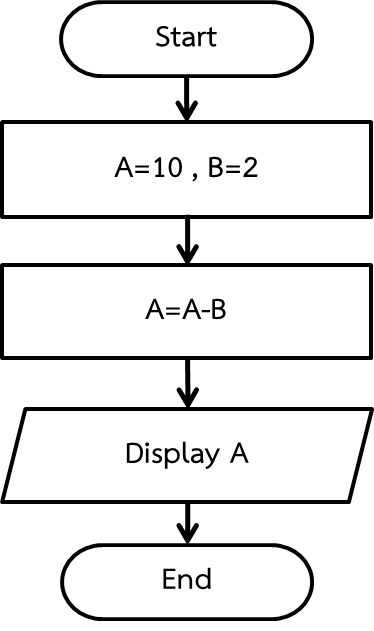


**อธิบายการทำงานของ Flowchart ได้ดังนี้**  
1. เริ่มการทำงานของโปรแกรม โดยกำหนดค่าตัวแปร A และ B มีค่าเท่ากับ 5 และ 3   
ตามลำดับ  
 2. กำหนดตัวแปร A มีค่าเท่ากับ A+B จบการทำงานของโปรแกรม  
  
**จาก Flowchart สามารถเขียนโค้ดโปรแกรมดังต่อไปนี้**

|  |  |
| --- | --- |
| บรรทัดที่ | โค้ดโปรแกรม |
| 1 2 3 4 5 6 7 8 | #include <stdio.h> void main() { int A, B; A=5; B=5; A=A+B; } |

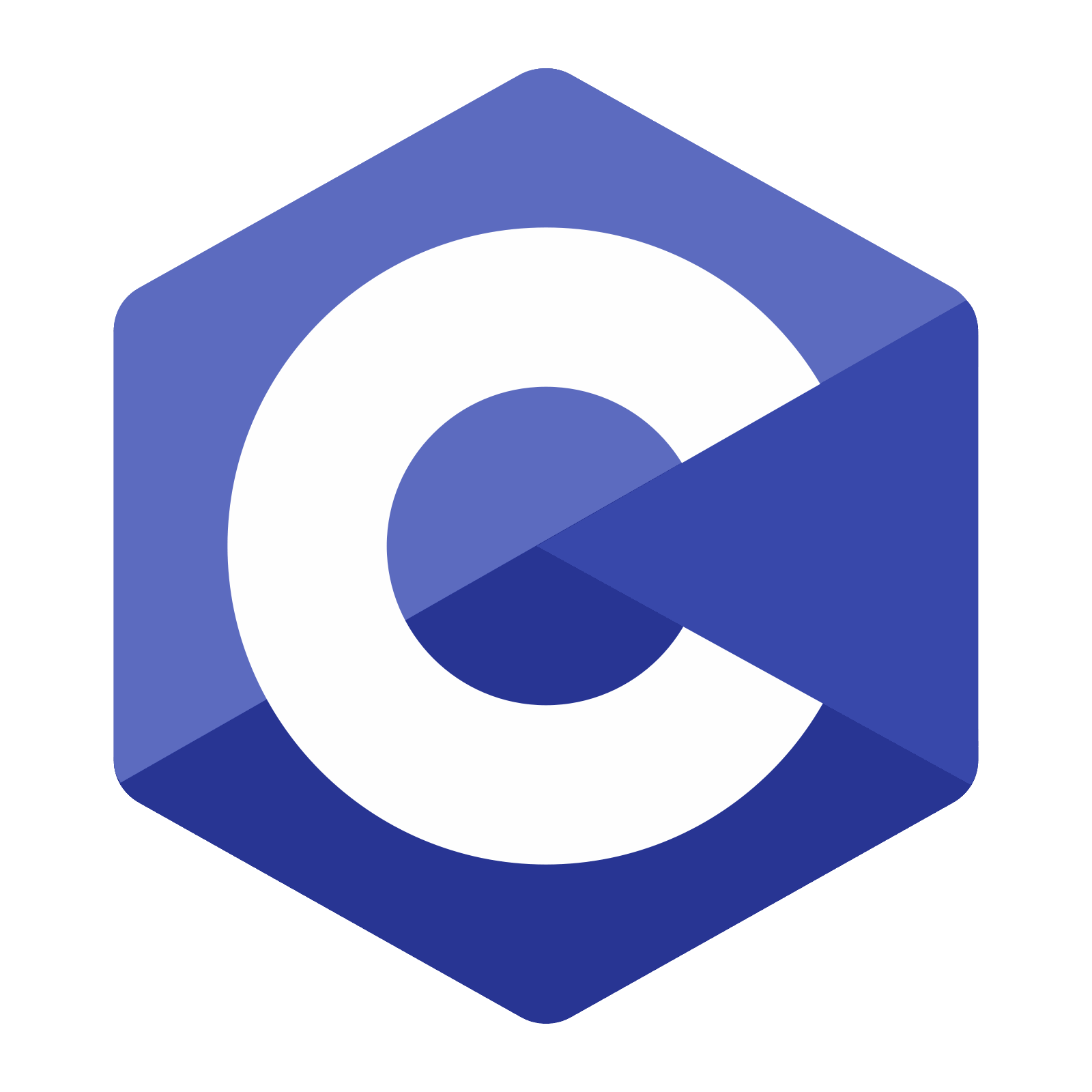
——————————————————————————————————————————

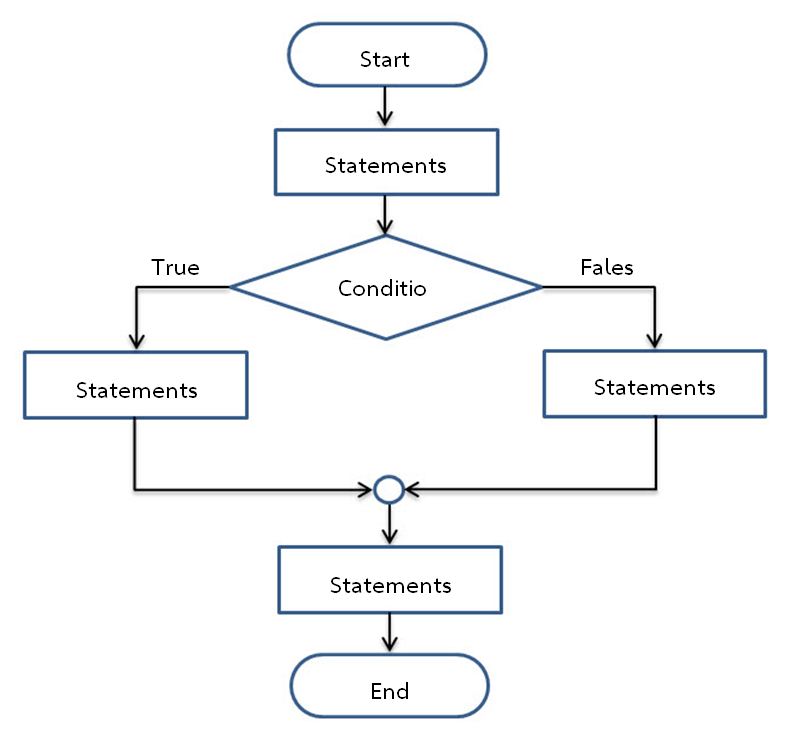
**ตัวอย่างที่ 2 การเขียนและอธิบายการทำงานของ Flowchart แบบตามลำดับดังนี้**



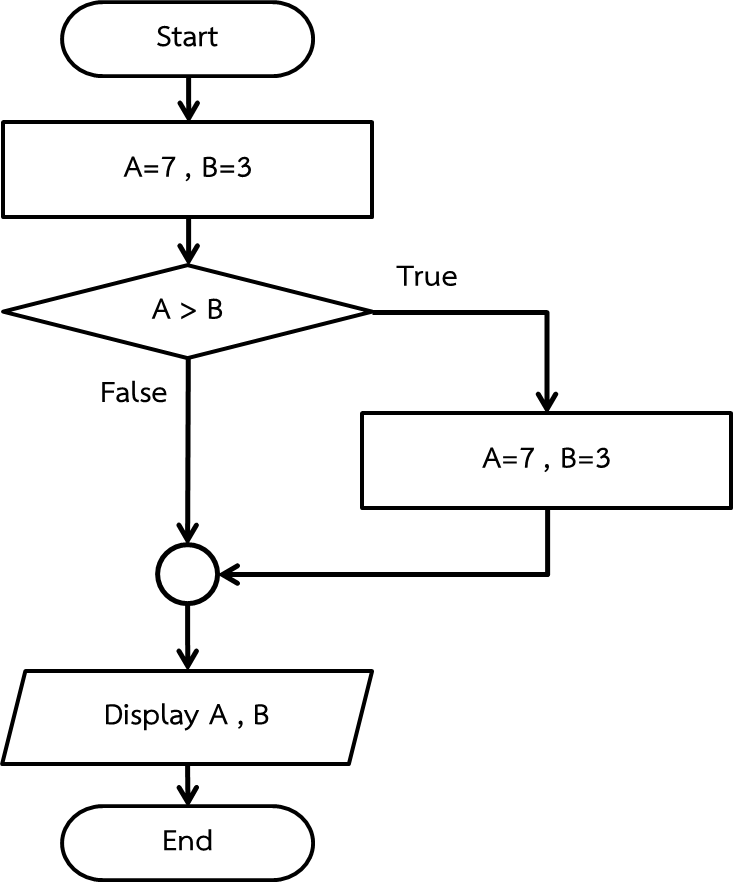
**อธิบายการทำงานของ Flowchart ได้ดังนี้**  
1. เริ่มการทำงานของโปรแกรม โดยกำหนดค่าตัวแปร A และ B มีค่าเท่ากับ 10 และ 2 ตามลำดับ   
2. กำหนดให้ค่าตัวแปร A มีค่าเท่ากับ A-B  
3. แสดงค่าตัวแปร A ทางจอภาพ จบการทำงานของโปรแกรม  
  
**จาก Flowchart สามารถเขียนโค๊ดโปรแกรมดังต่อไปนี้**

|  |  |
| --- | --- |
| บรรทัดที่ | โค้ดโปรแกรม |
| 1 2 3 4 5 6 7 8 | #include <stdio.h> void main() { int A, B; A=10; B=2; A=A-B; } |

**2. การเลือกกระทำตามเงื่อนไข (Decision) **การเลือกทำงานตามเงื่อนไข (Decision) เป็นรูปแบบการเขียนโปรแกรมที่มีทางเลือกเพื่อตัดสินใจซึ่งโปรแกรมจะตรวจสอบเงื่อนไข เพื่อเลือกทิศทางการทำงานของโปรแกรม โดยเลือกทางเลือกใดทางเลือกหนึ่ง จากสองทางเลือกเท่านั้นคือ  
> ทำงานในทางเลือกหนึ่งเมื่อเงื่อนไขเป็นจริง และ  
      > ทำงานในอีกทางเลือกหนึ่งเมื่อเขื่อนไขเป็นเท็จ เมื่อทำงานในแต่ละทางเลือกเสร็จแล้ว โปรแกรมก็จะทำงานในขั้นตอนต่อไป ซึ่งมีรูปแบบการทำงานดังรูปที่ 1

*****รูปที่ 1 การเลือกกระทำตามเงื่อนไข (Decision)*  
——————————————————————————————————————————

**ตัวอย่างที่ 1 การเขียนและอธิบายการทำงานของ Flowchart แบบเลือกการทำงานตามเงื่อนไขดังนี้**



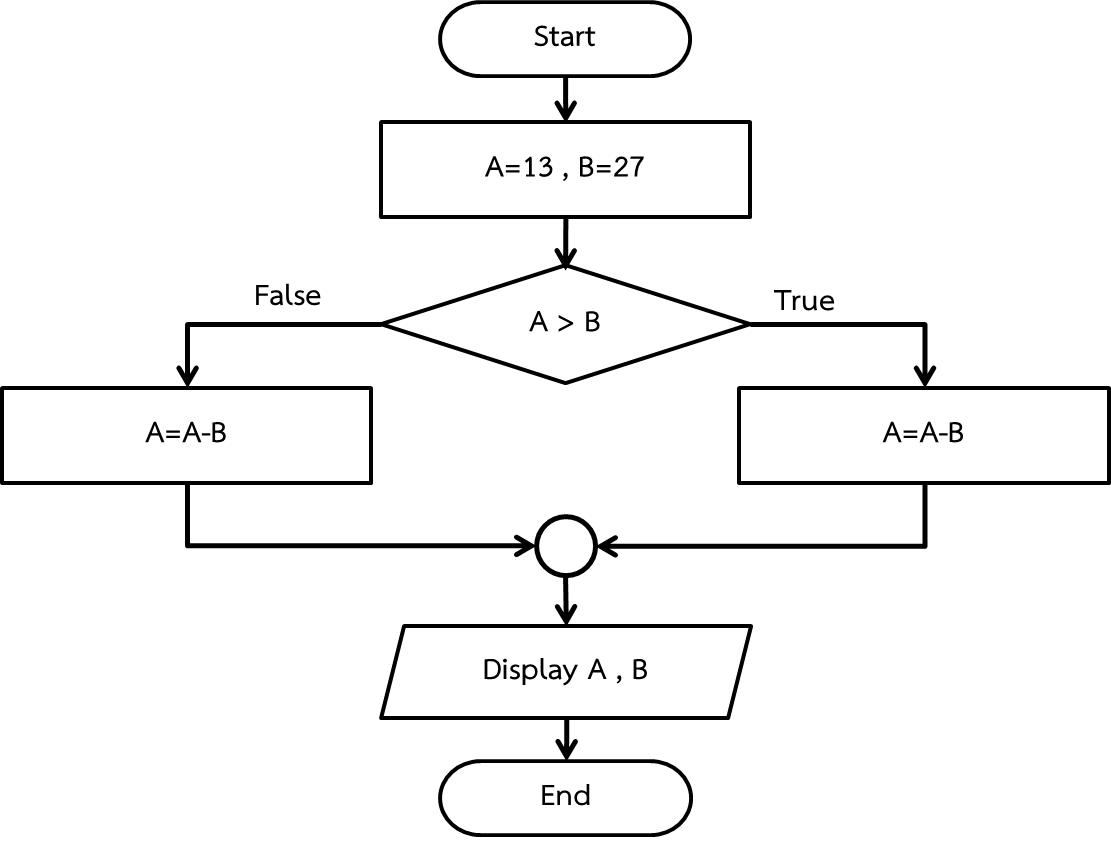
**อธิบายการทำงานของ Flowchart ได้ดังนี้**1. เริ่มการทำงานของโปรแกรม โดยกำหนดค่าตัวแปร A และ B มีค่าเท่ากับ 7 และ 3  
ตามลำดับ  
              2. ตรวจสอบเงื่อนไขว่าค่าตัวแปร A มากกว่าค่าตัวแปร B หรือไม่   
       3. ถ้าเงื่อนไงว่าค่าตัวแปร A มากกว่าค่าตัวแปร B เป็นจริง กำหนดให้ตัวแปร A มีค่า  
 เท่ากับ A+1  4. แสดงค่าตัวแปร A และ B ทางจอภาพ จบการทำงานของโปรแกรม  
  
**จาก Flowchart สามารถเขียนโค้ดโปรแกรมได้ดังนี้**

|  |  |
| --- | --- |
| บรรทัดที่ | โค๊ดโปรแกรม |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 | #include <stdio.h> void main() { int A, B; A=7; B=3; if (A>B) { A=A+1; }  printf("A=%d, B=%d",A,B); getch(); } |

ผลลัพธ์บนจอภาพ

|  |
| --- |
| A=8, B=3 |

**ตัวอย่างที่ 2 การเขียนและอธิบายการทำงานของ Flowchart แบบเลือกการทำงานตามเงื่อนไขดังนี้**

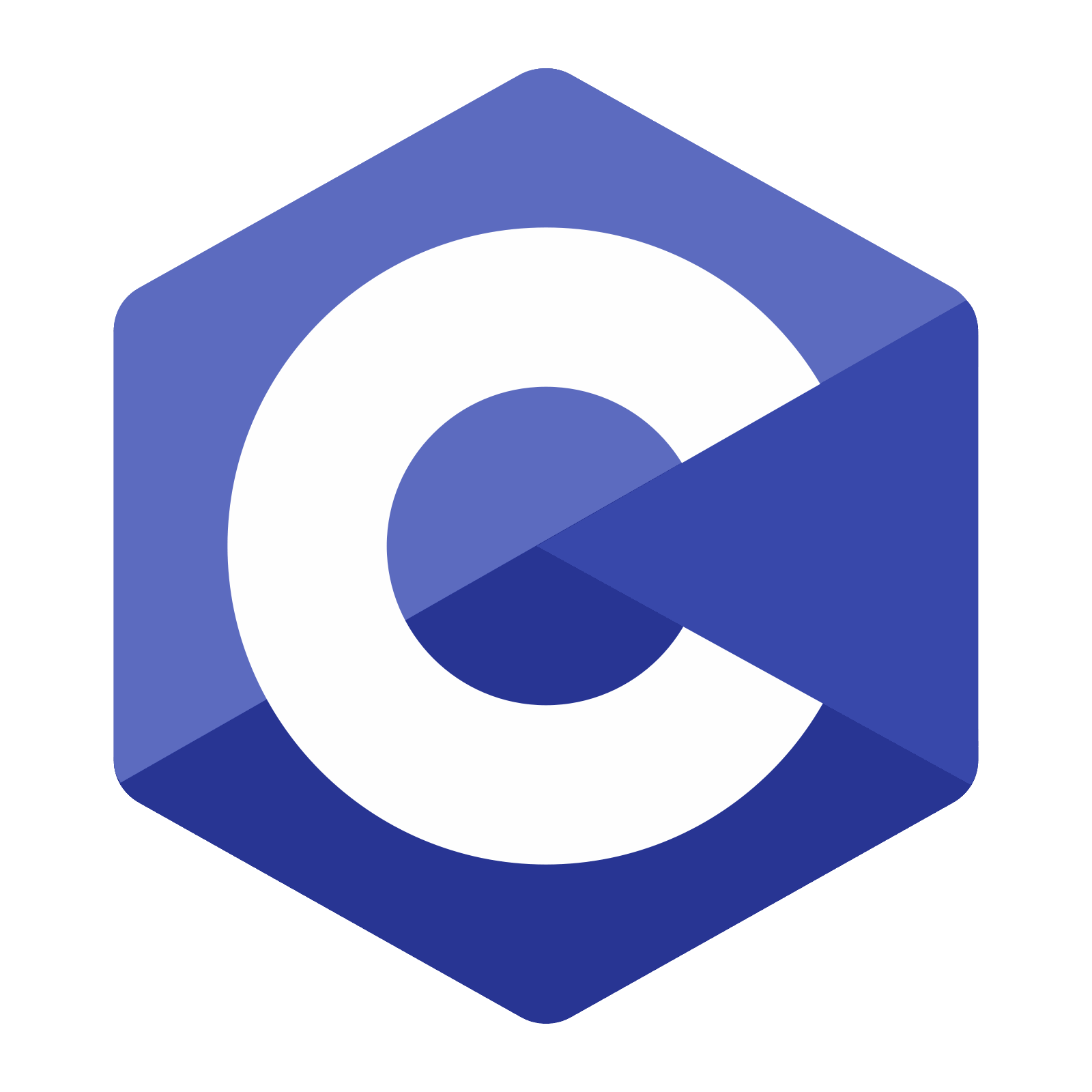


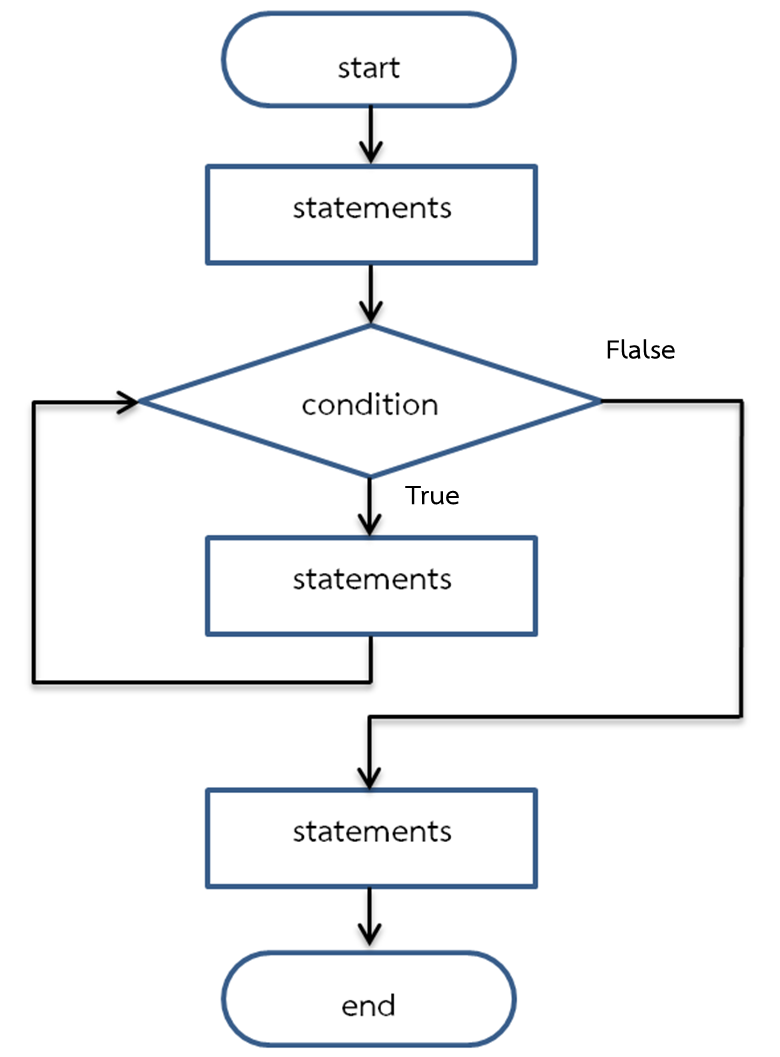
**อธิบายการทำงานของ Flowchart ได้ดังนี้**              1. เริ่มต้นการทำงานของโปรแกรม โดยกำหนดค่าตัวแปร A และ B มีค่าเท่ากับ 13   
และ 27 ตามลำดับ  
              2. ตรวจสอบเงื่อนไขว่าตัวแปร A มากกว่าตัวแปร B หรือไม่  
              3. ถ้าเงื่อนไขตัวแปร A มากกว่าค่าตัวแปร B เป็นจริง ให้กำหนดให้ตัวแปร A มีค่า  
เท่ากับ A+B  
              4. แต่ถ้าเงื่อนไขค่าตัวแปร A มากกว่าค่าตัวแปร B เป็นเท็จ ให้กำหนดตัวแปร A มีค่า  
เท่ากับ A-B  
              5. แสดงค่าตัวแปร A และ B ทางจอภาพ จบการทำงานของโปรแกรม  
  
**จาก Flowchart สามารถเขียนโค้ดโปรแกรมได้ดังนี้**

|  |  |
| --- | --- |
| บรรทัดที่ | โค๊ดโปรแกรม |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 | #include <stdio.h> void main() { int A, B; A=13; B=27; if (A>B) { A=A+B; } else {       A=A-B;  } printf("A=%d, B=%d,A,B"); getch(); } |

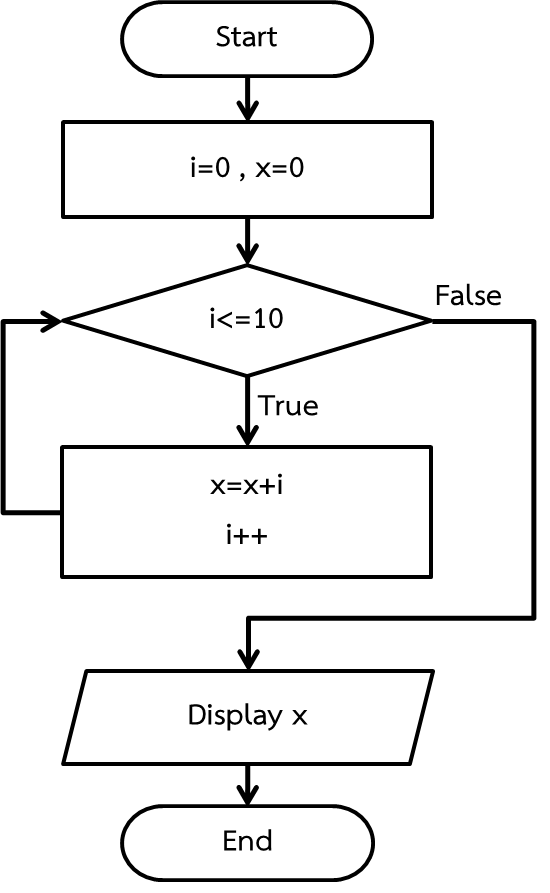
**ผลลัพธ์บนจอภาพ**

|  |
| --- |
| A=-14, B=27 |

3. การทำซ้ำ (Loop)   
การทำซ้ำ (Loop) เป็นรูปแบบการเขียนโปรแกรมที่มีการทำงานในขั้นตอนเดิมซ้ำ ๆ กัน หลาย ๆ รอบ ซึ่งการทำงานของโปรแกรมจะมีการตรวจสอบเงื่อนไข เพื่อกำหนดให้เข้าทำงานในลูป หรือออกจากลูป  
การทำงาน ซึ่งมีรูปแบบการทำงานดังรูปที่ 1

  
รูปที่ 1 การทำซ้ำ(Loop)   
——————————————————————————————————————————

**ตัวอย่างที่ 1 การเขียนและอธิบายการทำงานของ Flowchart แบบทำซ้ำดังนี้**



**อธิบายการทำงานของ Flowchart ได้ดังนี้**  
1. เริ่มการทำงานของโปรแกรม โดยกำหนดค่าตัวแปร i และ x มีค่าเท่ากับ 0   
และ 0 ตามลำดับ  
2. ตรวจสอบเงื่อนไขว่าค่าตัวแปร i น้อยกว่าหรือเท่ากับ 10 หรือไม่  
3. ถ้าเงื่อนไขค่าตัวแปร i น้อยกว่าหรือเท่ากับ 10 เป็นจริง กำหนดให้ตัวแปร x มีค่า  
เท่ากับ x+i และตัวแปร i เพิ่มค่า 1 ค่า คือ i++   
4. เมื่อตรวจสอบเงื่อนไขว่าค่าตัวแปร i น้อยกว่าหรือเท่ากับ 10 เป็นเท็จ ให้ออก  
จากลูปการทำงาน  
5. แสดงค่าตัวแปร x ทางจอบภาพ จบการทำงานของโปรแกรม  
  
**จาก Flowchart สามารถเขียนโค้ดโปรแกรมได้ดังนี้**

|  |  |
| --- | --- |
| บรรทัดที่ | โค้ดโปรแกรม |
| 1 2 3 4 5 6 7 8 9 10 11 12 | #include <stdio.h> int main() { int i, x; x=0; for (i=0; i<=10;i++) { x=x+i;  } printf("x = %d",x); getch(); } |

**ผลลัพธ์บนจอภาพ**

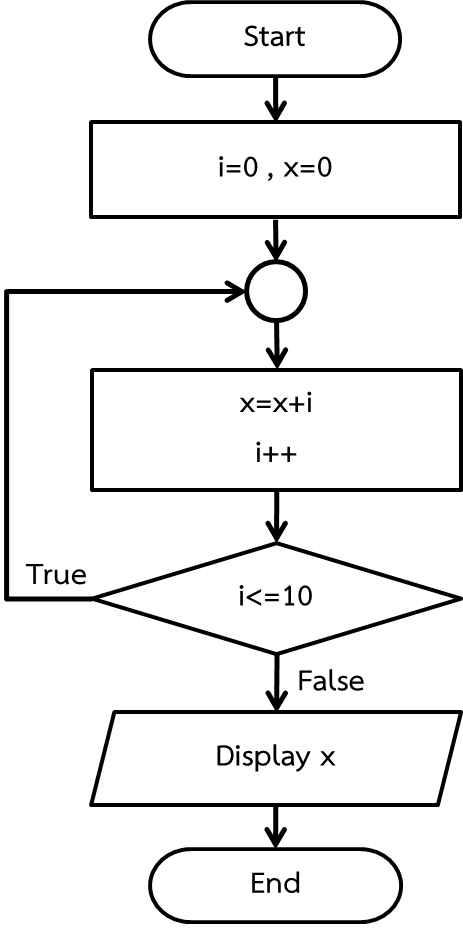
|  |
| --- |
| x = 55 |

**จาก Flowchart สามารถเขียนโค้ดโปรแกรมได้อีกรูปแบบหนึ่งดังนี้**

|  |  |
| --- | --- |
| บรรทัดที่ | โค้ดโปรแกรม |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 | #include <stdio.h> int main() { int i, x; i=0; x=0; while (i<=10)  { x=x+i;  } printf("x = %d",x); getch(); } |

——————————————————————————————————————————

**ตัวอย่างที่ 2 การเขียนและอธิบายการทำงานของ Flowchart แบบทำซ้ำดังนี้**

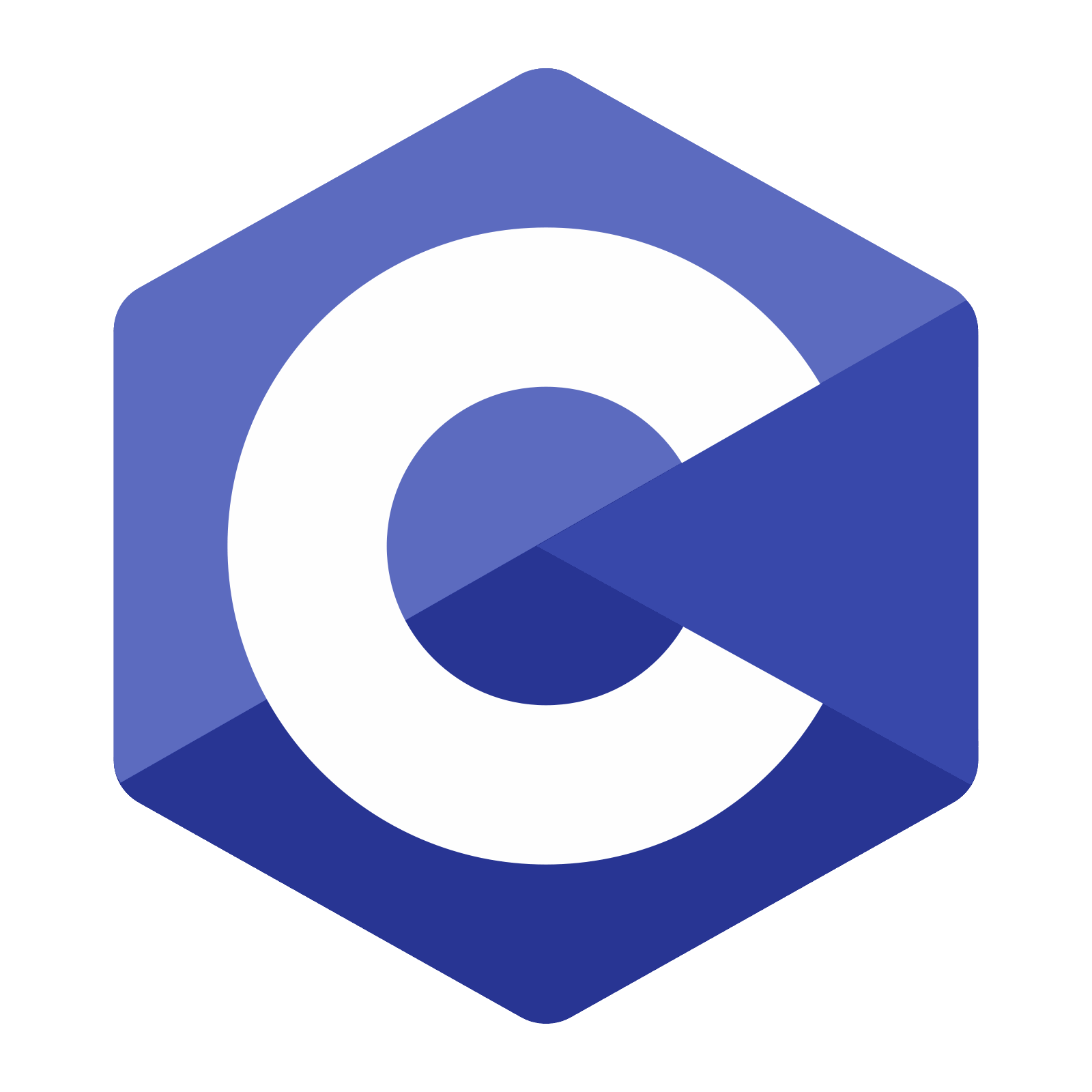


**อธิบายการทำงานของ Flowchart ได้ดังนี้**  
1. เริ่มการทำงานของโปรแกรม โดยกำหนดค่าตัวแปร i และ x มีค่าเท่ากับ 0   
 และ 0 ตามลำดับ  
2. กำหนดให้ตัวแปร x มีค่าเท่ากับ x + i   
3. ตัวแปร i เพิ่มค่า 1 ค่า คือ i++   
4. ตรวจสอบเงื่อนไขว่าค่าตัวแปร i น้อยกว่า 10 หรือไม่  
5. ถ้าเงื่อนไขค่าตัวแปร i น้อยกว่า 10 เป็นจริง ให้กลับไปทำที่ข้อ 2  
6. แต่ถ้าเงื่อนไขค่าตัวแปร i น้อยกว่า 10 เป็นเท็จ ให้ออกจากลูปการทำงาน  
7. แสดงค่าตัวแปร x ทางจอภาพ จบการทำงานของโปรแกรม   
  
**จาก Flowchart สามารถเขียนโค้ดโปรแกรมได้อีกรูปแบบหนึ่งดังนี้**

|  |  |
| --- | --- |
| บรรทัดที่ | โค้ดโปรแกรม |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | #include <stdio.h> int main() { int i, x; i=0; x=0; do { x=x+i;  i++; } while (i<=10); printf("x = %d",x); getch(); } |

**ผลลัพธ์บนจอภาพ**

|  |
| --- |
| x = 55 |

ประโยชน์ของผังงาน   
     1. ทำให้มองเห็นภาพของระบบและโปรแกรมได้ทั้งหมด และใช้เวลาในการเข้าใจการทำงานได้เร็วขึ้น   
     2. การเขียนผังงานใช้สัญลักษณ์เป็นสากล สามารถนำไปเขียนและสื่อความหมาย ความเข้าใจได้ทุกภาษา   
     3. ทำให้ตรวจพบข้อผิดพลาดได้ง่าย และสามารถแก้ไขจุดผิดพลาดได้อย่างรวดเร็ว   
     4. หากมีการขยายงานต่อจากผังงานเดิมก็สามารถเพิ่มเติมได้ โดยอาศัยได้ดูผังงานเดิมประกอบ   
จะทำให้เข้าใจได้รวดเร็วกว่าการดูรายละเอียดจากโปรแกรม หรือระบบงานเดิม