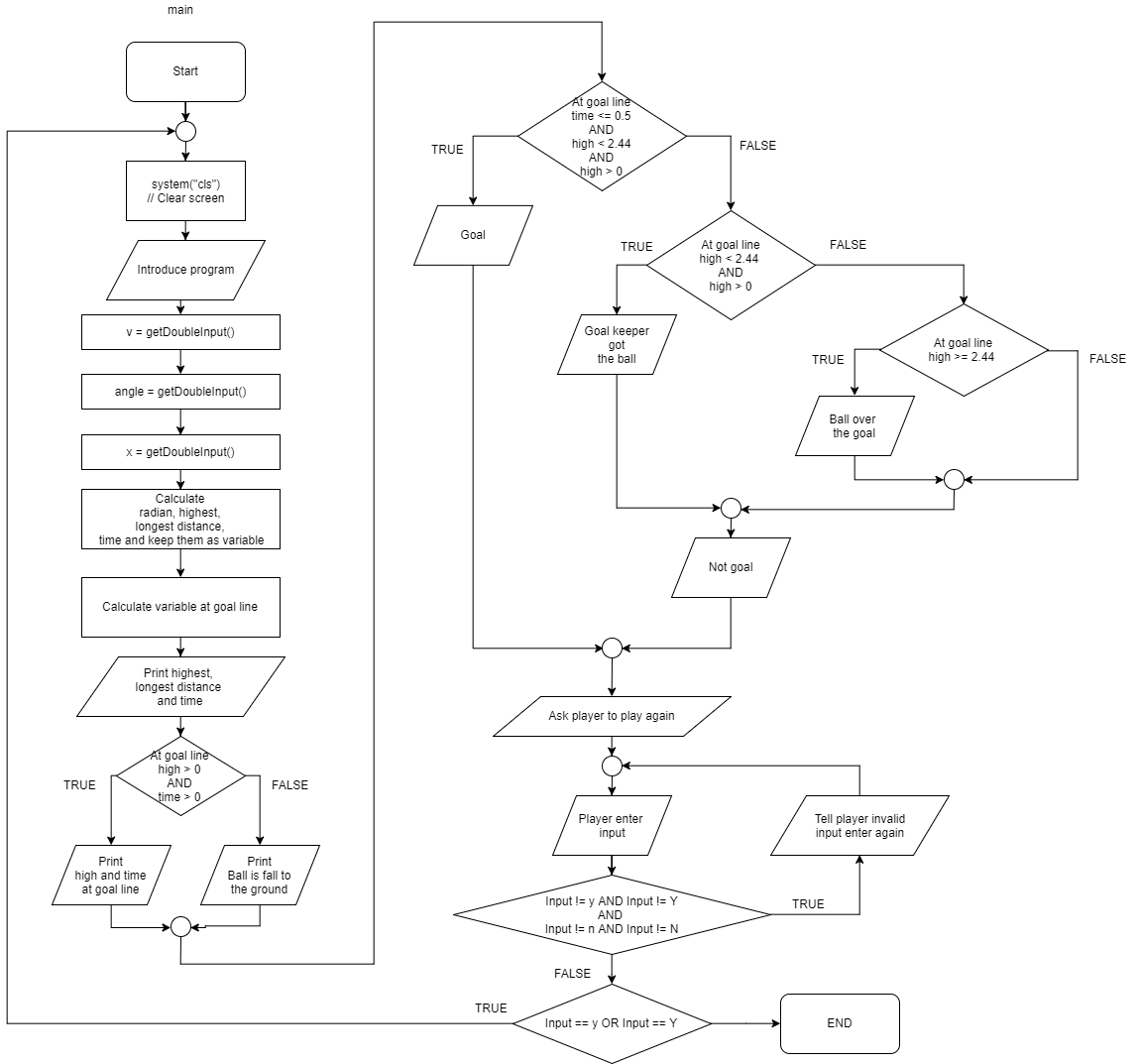
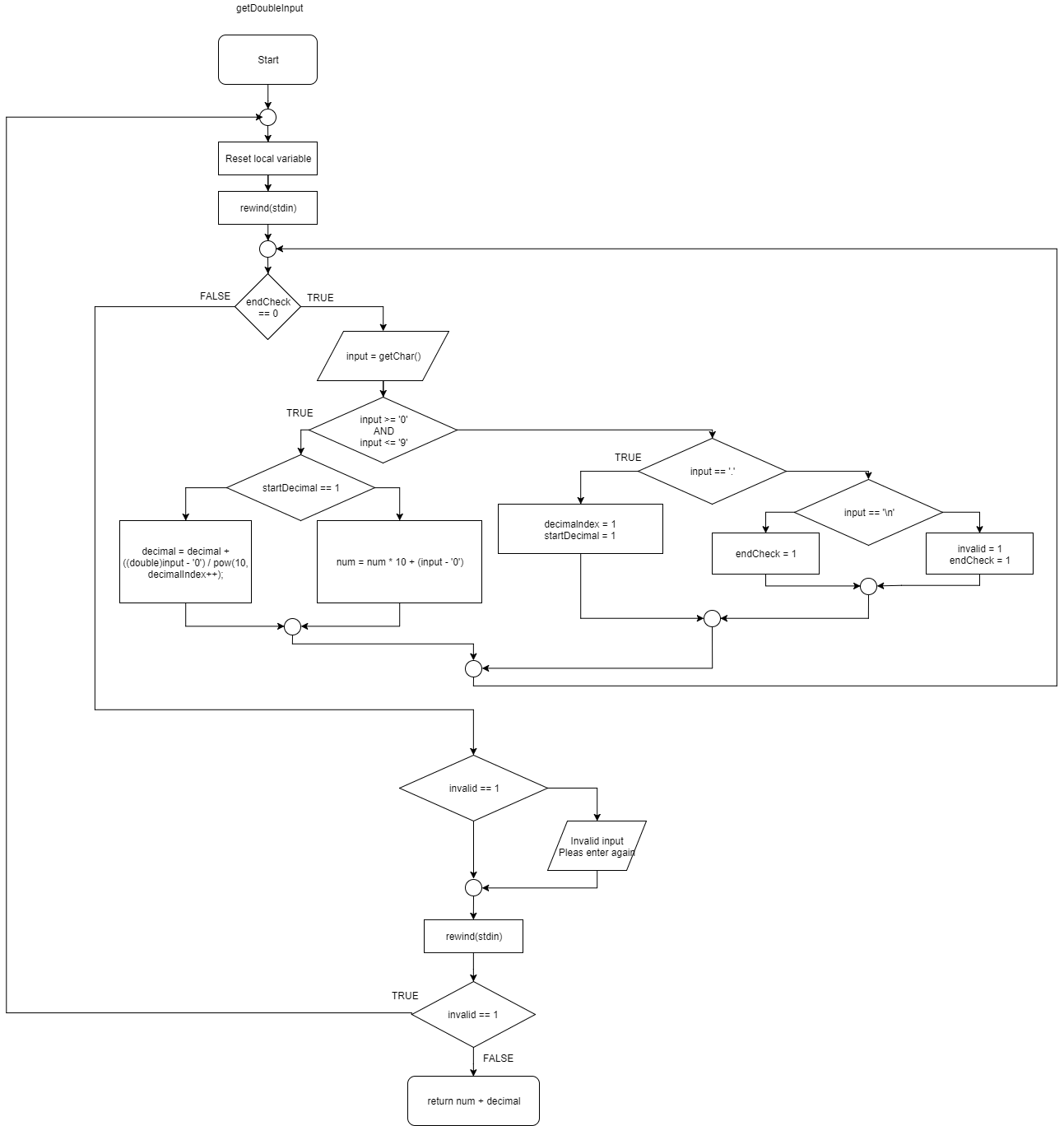
**Assignment 4: Project tile**

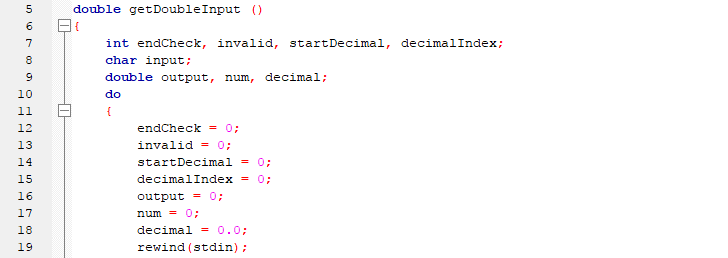
**Flow chart: Main function**

**Flowchart: getDoubleInput function**

**อธิบายการทำงานของโค้ด**



**บรรทัดที่ 1 – 3:** include library ที่จำเป็นต้องใช้



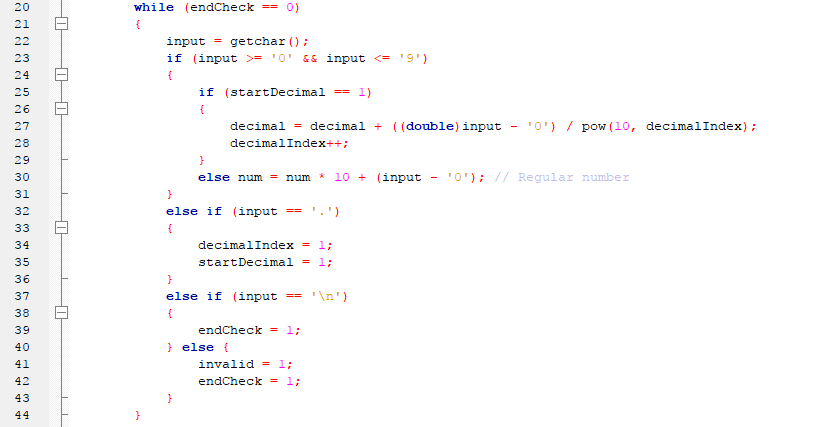
**บรรทัดที่ 5**: ประกาศฟังก์ชัน getDoubleInput สำหรับเช็คความถูกต้องของ input

**บรรทัดที่ 7 – 9:** ประกาศตัวแปรเตรียมไว้สำหรับเช็ค input

**บรรทัดที่ 10**: เริ่มลูป do while

**บรรทัดที่ 12 - 18:** รีเซ็ตค่าตัวแปรสำหรับลูป

**บรรทัดที่ 19:** ลบค่าที่ค้างอยู่ใน buffer



**บรรทัดที่ 20:** เริ่มลูป do while ทำงานเมื่อ endCheck == 0 หรือก็คือยังเช็คค่า input ไม่เสร็จ

**บรรทัดที่ 22:** รับค่า input

**บรรทัดที่ 23:** เช็ครหัส ASCII ของ input ว่าอยู่ระหว่าง 0 – 9 หรือไม่

**บรรทัดที่ 25 – 30:** เช็คว่าเริ่มต้นเก็บค่าในตำแหน่งทศนิยมหรือยัง

**ถ้าเริ่มแล้ว** ให้แปลงค่า input แล้วเก็บเป็นทศนิยมตามตำแหน่งไว้ที่ decimal

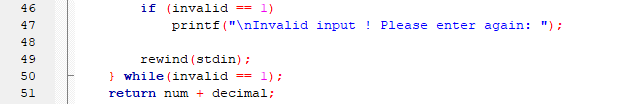
**ถ้ายังไม่ได้เริ่ม** ให้แปลงค่า input แล้วเพิ่มหน่วย num และเพิ่ม input ไปที่ num

**บรรทัดที่ 32 - 36:** ถ้า input เท่ากับ . ให้เริ่มเก็บค่าในตำแหน่งทศนิยม

**บรรทัดที่ 37 - 39:** ถ้าว่า input เท่ากับขึ้นบรรทัดใหม่ ให้ endCheck = 1 เพื่อจบการตรวจสอบ Input

**บรรทัดที่ 40 – 43**: เมื่อไม่เข้าเงื่อนไขใดๆ เลย แสดงว่าได้ input มาไม่ถูกต้อง กำหนด invalid = 1, endCheck = 1

เพื่อให้จบการตรวจสอบค่า input และจำไว้ว่าค่า input ไม่ถูกต้อง

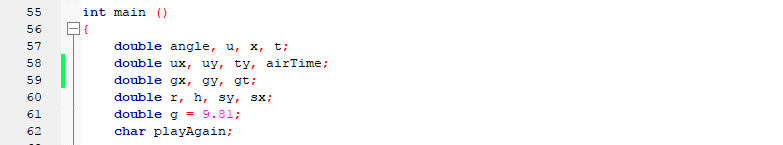


**บรรทัดที่ 46:** ถ้าค่า input ไม่ถูกต้อง ให้บอกผู้เล่นว่าค่า input ไม่ถูกต้องให้กรอกใหม่

**บรรทัดที่ 49:** ลบค่าที่ค้างใน buffer ออก

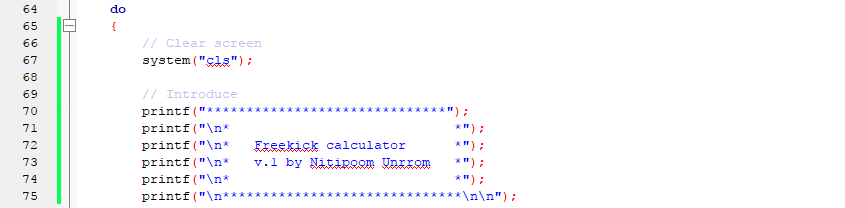
**บรรทัดที่ 50:** ถ้าค่า input ไม่ถูกต้องทำงานลูปใหม่

**บรรทัดที่ 51:** return ค่าตัวเลขที่ได้จากการตรวจสอบ input กลับไปแบบ double



**บรรทัดที่ 55:** เริ่มฟังก์ชัน main

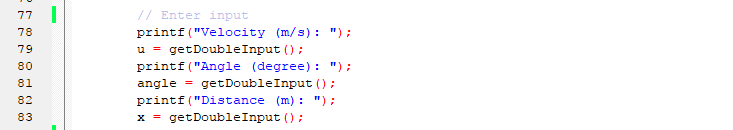
**บรรทัดที่ 57 – 62:** ประกาศตัวแปร



**บรรทัดที่ 64:** เริ่มลูป do while สำหรับเริ่มเล่นรอบใหม่

**บรรทัดที่ 67:** clear screen ล้างหน้าจอจากลูปครั้งที่แล้วออก เพื่อความสวยงามไม่รกหน้าจอ

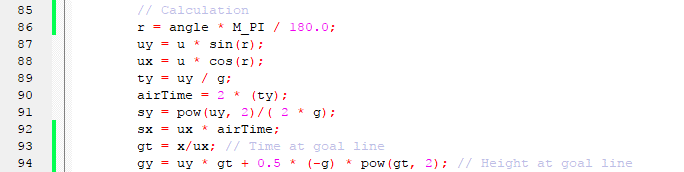
**บรรทัดที่ 70 – 75:** โปรแกรมแนะนำตนเอง เนื่องจากมีการล้างหน้าจอออกใหม่ทุกรอบที่เริ่มเล่น จึงต้องแนะนำโปรแกรมใหม่ทุกครั้งเพื่อความสวยงาน



**บรรทัดที่ 78 – 79**: ถามความเร็ว และให้ฟังก์ชัน getDoubleInput รับค่า Input จากผู้เล่นไปตรวจสอบและส่งค่ากลับมาเป็น double เก็บไว้ที่ตัวแปร u

**บรรทัดที่ 78 – 79:** ถามมุม และให้ฟังก์ชัน getDoubleInput รับค่า Input จากผู้เล่นไปตรวจสอบและส่งค่ากลับมาเป็น double เก็บไว้ที่ตัวแปร angle

**บรรทัดที่ 78 – 79:** ถามระยะทาง และให้ฟังก์ชัน getDoubleInput รับค่า Input จากผู้เล่นไปตรวจสอบและส่งค่ากลับมาเป็น double เก็บไว้ที่ตัวแปร x



**บรรทัดที่ 86:** คำนวณแปลงมุมองศาเป็นเรเดียน

**บรรทัดที่ 87 – 88:** คำนวณแรงในแนวนาบและแนวระดับ

**บรรทัดที่ 89:** คำนวณหาเวลาที่ตำแหน่งสูงสุด

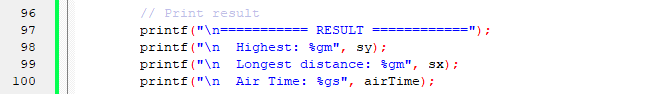
**บรรทัดที่ 90:** คำนวณหาเวลาที่ลอยอยู่ในอากาศ

**บรรทัดที่ 91:** คำนวณหาค่าความสูงจากพื้นของลูกบอลที่ลอยขึ้นไปได้สูงที่สุด

**บรรทัดที่ 92:** คำนวณหาระยะที่ไกลที่สุดที่ลูกบอลไปได้

**บรรทัดที่ 93:** คำนวณหาเวลาที่ลูกบอลลอยไปถึงที่ตำแหน่งเส้นประตู

**บรรทัดที่ 94:** คำนวณหาค่าความสูงจากพื้นของลูกบอลที่ลอยอยู่ที่ตำแหน่งเส้นประตู



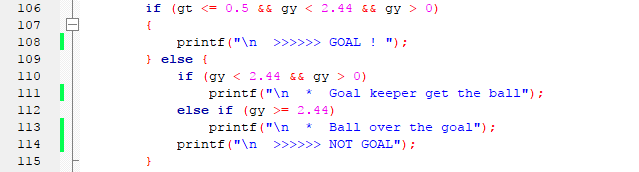
**บรรทัดที่ 97 – 100:** บอกค่าที่ได้จากการคำนวณกับผู้เล่น ว่าลูบอลลอยได้สูงสุดเท่าไร ไปได้ไกลเท่าใด และเวลาที่ลอยอยู่ในอากาศนานเท่าไร



**บรรทัดที่ 102 – 104:** ตรวจสอบว่าบอลลอยเหนือพื้นที่ตำแหน่งเส้นประตูหรือไม่

**ถ้าใช่** ให้แสดงค่าความสูงที่ตำแหน่งเส้นประตูและเวลาที่ลูกบอลไปถึง

**ถ้าไม่ใช่** ให้บอกผู้เล่นว่าบอลตกถึงพื้นก่อนถึงเส้นประตู



**บรรทัดที่ 106 - 109:** ตรวจว่าที่ตำแหน่งเส้นประตูบอลใช่เวลาเดินทางมาถึงน้อยกว่าหรือเท่ากับ 0.5 วินาที รวมทั้งยังไม่ตกถึงพื้นและลอยต่ำกว่า 2.44 เมตร

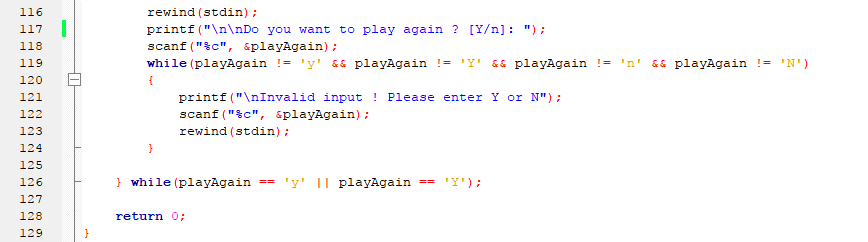
**ถ้าใช่** แสดงว่าบอลเข้าประตู ให้บอกผู้เล่นว่าลูกบอลเข้าประตู

**ถ้าไม่ใช่** แสดงว่าบอลไม่เข้าประตูให้ทำคำสั่งในบรรทัดที่ 110 – 114 เพื่อหาสาเหตุว่าทำไมบอลถึงไม่เข้าประตู

**บรรทัดที่ 110 – 111:** ถ้าบอลลอยอยู่ต่ำกว่า 2.44 เมตร และลอยเหนือพื้น แสดงว่าเวลาที่บอลเดินทางมาถึงโกลด์เกิน 0.5 บอกผู้เล่นว่า ผู้รักษาประตูสามารถรักษาประตูเอาไว้ได้

**บรรทัดที่ 112 - 113:** ถ้าบอลสูงกว่าหรือเท่ากับ 2.44 เมตร ให้บอกผู้เล่นว่า บอลลอยข้ามประตู

**บรรทัดที่ 114:** บอกผู้เล่นว่าบอลไม่เข้าประตู



**บรรทัดที่ 116:** ลบค่า input ที่ค้างอยู่ใน buffer เพื่อรอรับค่า input ใหม่จากผู้เล่น

**บรรทัดที่ 117:** ถามผู้เล่นว่าต้องการเล่นใหม่หรือไม่

**บรรทัดที่ 118:** รับ input จากผู้เล่น

**บรรทัดที่ 119:** เริ่มลูป while ทำงานเมื่อค่า input ที่ได้รับมาไม่ใช่ y, Y, n, N

**บรรทัดที่ 121:** บอกผู้เล่นว่าใส่ค่า input ไม่ถูกต้องให้ใส่มาใหม่

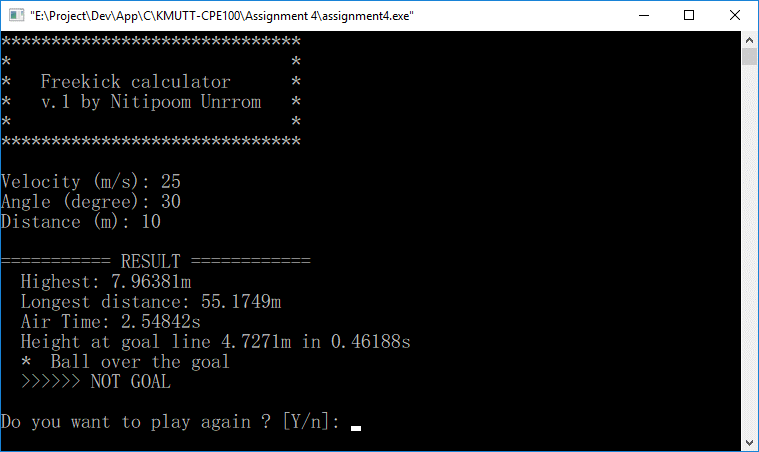
**บรรทัดที่ 122:** รอรับค่าที่ผู้เล่นจะใส่ input มาใหม่

**บรรทัดที่ 123:** ลบค่า input ที่ค้างอยู่ใน buffer

**บรรทัดที่ 126:** ถ้าค่า input เท่ากับ y หรือ Y ให้วนกลับไปทำงานใน do เพื่อเล่นใหม่อีกครั้ง

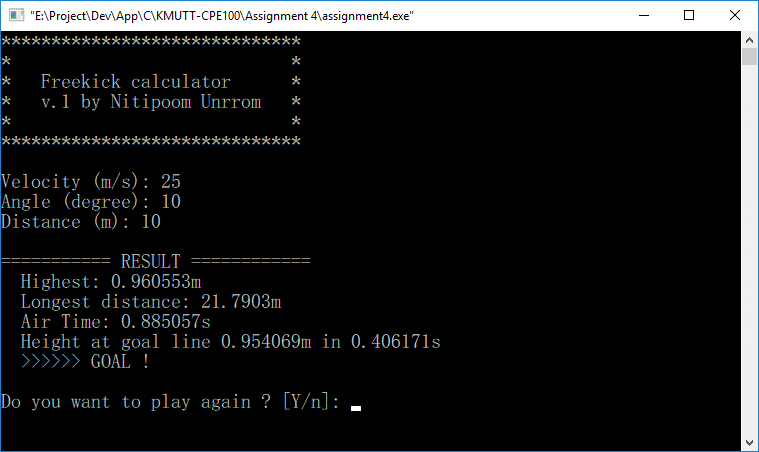
**ผลการทดสอบการทำงานของโค้ด**

1. ทดสอบที่ **ความเร็ว** 25 **มุม** 30 **ระยะทาง** 10   
ผลลัพธ์ บอลลอยข้ามประตู บอลไม่เข้าประตู



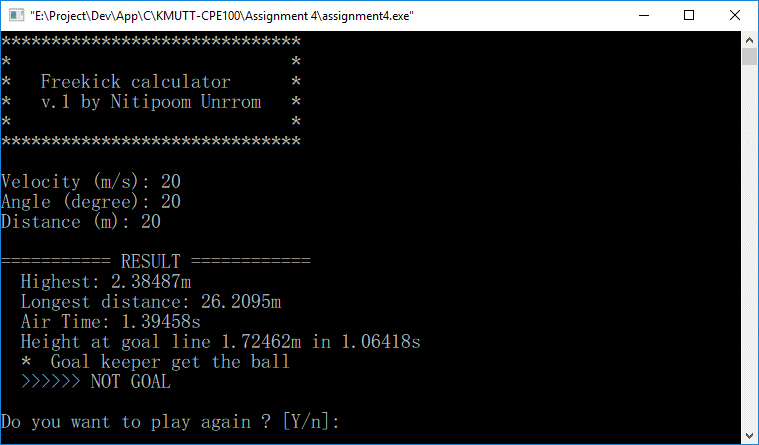
2. ทดสอบที่ **ความเร็ว** 25 **มุม** 10 **ระยะทาง** 10

ผลลัพธ์ บอลเข้าประตู



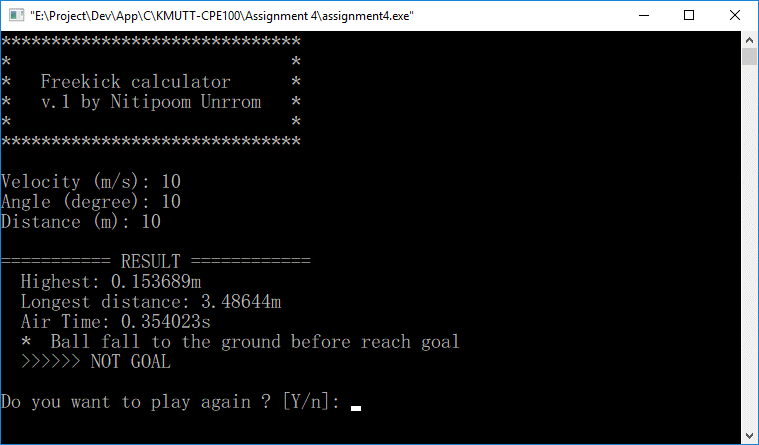
3. ทดสอบที่ **ความเร็ว** 20 **มุม** 20 **ระยะทาง** 20

ผลลัพธ์ ผู้รักษาประตูรักษาประตูเอาไว้ได้ บอลไม่เข้าประตู

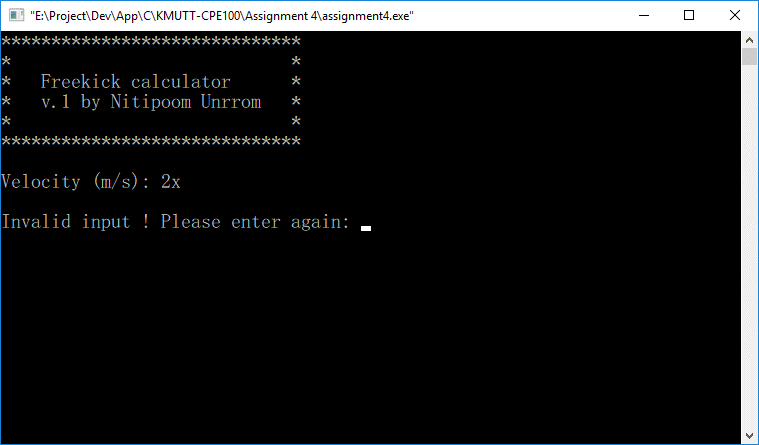


4. ทดสอบที่ **ความเร็ว** 10 **มุม** 10 **ระยะทาง** 10

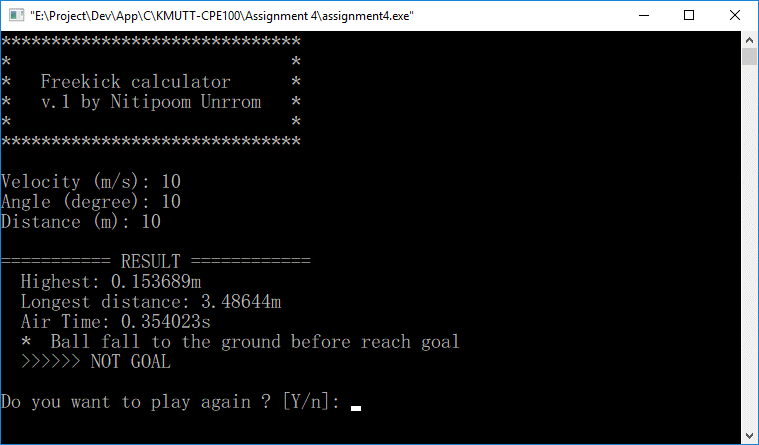
ผลลัพธ์ บอลตกถึงพื้นก่อนถึงเส้นประตู บอลไม่เข้าประตู



5. ทดสอบการป้องกันการหยุดทำงาน เมื่อป้อนค่าผิด



6. คำถามเพื่อถามผู้เล่นว่าต้องการเล่นใหม่หรือไม่



หน้าจอเมื่อเริ่มใหม่

