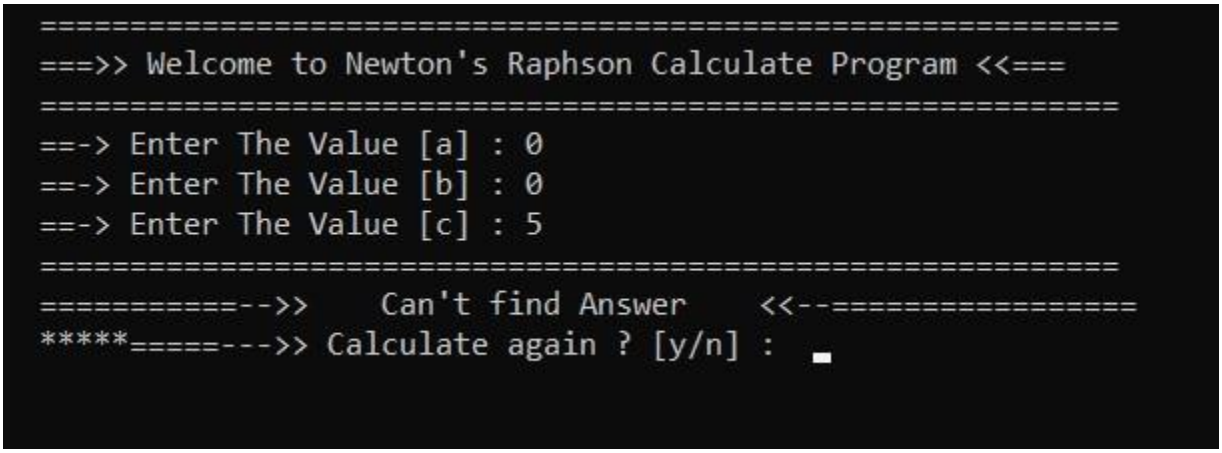
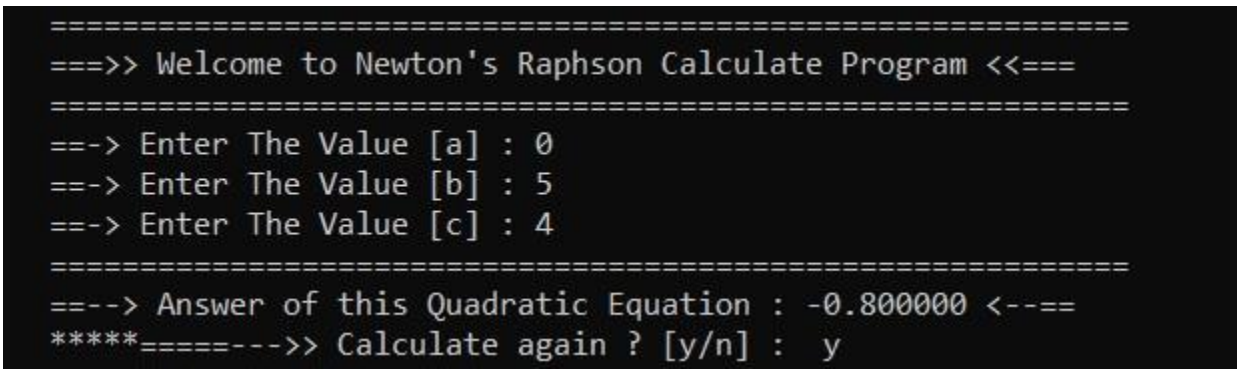
# Report : Assignment 6

Test case




Case 1 : รับค่าของสัมประสิทธิ์ของสมการกำลังสองมาคือ 0 0 5 ตามลำดับจากนั้นก็ส่งค่าไปคิดที่ฟังก์ NR ตามโปรแกรมโดยส่งค่า 0.00001 คือค่าผิดพลาดที่ยอมรับได้และ 123.123 คือค่า x ที่สุ่มตอนแรกให้ฟังก์ชั่นไปคิดหา x ตัวใหม่ที่ทำให้สมการเป็นจริงจากนั้นเมื่อส่งค่ามาที่ฟังก์ชั่น NR แล้วฟังก์ชั่นก็จะนำค่าที่ได้ไปคิดตามสมการที่เราเขียนไว้โดยอันดับแรกจะทำการเช็คก่อนว่าส่วนเป็น 0 หรือไม่ถ้าส่วนเป็น 0 ก็จะให้บวกค่าของ x ตัวแรกที่เราสุ่มมาไป 0.00001 เพื่อให้ส่วนไม่เป็น 0 แต่ในกรณีนี้ส่วนจะเป็น 0 เสมอจึงเพราะสปส.ของสมการเป็น 0 ถึง 2 ตัวซึ่งทำให้สมการความซันหรืออนุพันธ์ของสมการเป็น 0 เสมอและเมื่อเพิ่มค่าเสร็จก็จะทำการบวกค่าของตัวแปร count ที่เราใช้นับไป 1 และทำอย่างนี้ซ้ำไปเรื่อยๆเนื่องจากส่วนของสมการเป็น 0 เสมอจนตัวแปร count > 1000 ก็จะหลุดออกจาก loop หลังจากนั้นก็จะไปเข้าเงื่อนไข else และทำการรีเทิร์นค่า = 0 ให้กลับฟังก์ชั่นและนำค่าไปเก็บไว้ในตัวแปร count ฟังก์ชั่น  main พร้อมกับรีเทิร์นค่า x ที่ทำให้สมการเป็นจริงกลับไปด้วย(pointer)ต่อจากนั้นที่ฟังก์ชั่น main จะทำการเช็คว่าค่าของตัวแปร count เป็น 0 หรือไม่ซึ่งใน case ที่1 นี้ค่า count เป็น 0 ทำให้แสดงข้อความว่าไม่สามารถหาคำตอบของสมการได้และจะแสดงคำถามถามผู้ใช้ว่าจะคำนวณต่อหรือไม่ถ้าผู้ใช้ตอบ y คือคำนวณอีกรอบแต่ถ้าผู้ใช้ตอบ n คือออกจากโปรแกรม




Case 2 : การทำงานในขึ้นตอนแรกๆจะเหมือนกับ case 1 ทุกอย่างเพียงแต่รับค่ามาต่างกันแล้วก็ส่งไปคิดคำตอบเหมือน case 1 แต่แตกต่างตรงที่ case2 นั้นส่วนไม่มีทางเป็น 0 แล้วจะทำให้ทำงานในส่วนของ else หลังจากถูกเช็คว่าส่วนเป็น 0 ไหมด้วยเงื่อน if โดยการทำงานใน else คือจะให้ทำการคำนวณหาค่าผิดพลาดของ x ตัวใหม่กับตัวเก่าและหลังจากนั้นก็ไปคำนวณหาค่าของ x ตัวใหม่และกำหนดให้ x ตัวใหม่นั้นเท่ากับ x ตัวเก่า(เพื่อว่าถ้า x ที่เราคิดมานั้นไม่ใช่คำตอบก็จะนำค่า x ที่เราคิดมาไปหาค่าของ x ตัวใหม่ไปเรื่อยๆจนกว่าจะเจอ x ที่เป็นคำตอบ)หลังจากนั้นจะไปบวกค่า count ไป 1 และทำการเช็คเงื่อนไขว่าค่าพิดพลาดของ x ตัวเก่ากับใหม่นั้นมันมากกว่า 0.00001 หรือไม่ถ้ามากกว่าเงื่อนไขก็จะเป็นจริงและเช็คอีก 1 เงื่อนไขว่าเป็นจริงหรือไม่คือตัวแปร count (นับรอบ) นั้นน้อยกว่า 1000 หรือไม่ถ้าน้อยกว่าให้เป็นจริงโดยถ้าทั้งสองเงื่อนไขเป็นจริงจะทำไปเรื่อยจนกว่าจะมีอันใดอันหนึ่งเป็นเท็จ เมื่อเป็นเท็จก็จะหยุด loop และนำไปเช็คค่า count ว่า น้อยกว่า 1000 ให้รีเทิร์นค่าของ count กลับไปแต่ถ้าไม่ให้รีเทิร์นค่า 0 กลับไปโดย case นี้นั้นรีเทิร์นค่าของ count กลับมาทำให้ในฟังก์ชั่น main ทำให้ count ไม่เป็น 0 ถ้าไม่เป็น 0 จะให้แสดงค่าของคำตอบที่ 1 ออกมาหลังจากนั้นให้ส่งค่าให้ฟังก์ชั่น NR คิดอีกครั้งแต่เปลี่ยนค่า x ที่สุ่มให้ไปคิดตอนแรกเป็นติดลบบ้างเพราะอาจจะทำให้ได้คำตอบอีก 1 คำตอบ(อีก 1 คำตอบอาจอยู่ด้านซ้ายเพราะกราฟเป็นพาราโบลาซึ่งอาจจะมี 2 จุดที่ตัดกับแกน x) เมื่อส่งไปคิดฟังก์ชั่น NR ก็จะรีเทิร์นค่ากลับมาพร้อมกับคำตอบตัวที่ 2 ที่เรากำลังจะคิดจากนั้นเราทำการเช็คว่าคำตอบที่ได้กลับมา(ผ่าน pointer)นั้นมีค่าใกล้เคียงกับคำตอบแรกหรือไม่ ถ้ามีค่าไม่ใกล้เคียงกับคำตอบตัวแรกก็ให้แสดงค่าของคำตอบตัวที่ 2 ด้วย แต่ถ้าใกล้เคียงก็ไม่ต้องแสดงค่าของคำตอบตัวที่ 2 ซึ่งทำให้ใน case ที่ 2 นี้จึงมีคำตอบเดียวเพราะค่าของคำตอบ 2 คำตอบนั้นใกล้เคียงกัน(ใกล้เคียงกันก็คือแทบจะเป็นค่าเดียวกันเลย)

และจะแสดงคำถามถามผู้ใช้ว่าจะคำนวณต่อหรือไม่ถ้าผู้ใช้ตอบ y คือคำนวณอีกรอบแต่ถ้าผู้ใช้ตอบ n คือออกจากโปรแกรม

```
==-> Enter The Value [a] : 2
==-> Enter The Value [b] : 5
==-> Enter The Value [c] : 2
============================================================
==--> Answer of this Quadratic Equation : -0.500000 <--==
==--> And : -2.000000 <--==
*****======--->> Calculate again ? [y/n] :  y
```

Case 3 : case นี้เหมือนกันกับ case 2 แตกต่างที่ค่าที่นำไปคิดที่ฟังก์ชั่น NR โดย case นี้เมื่อนำไปคิดในฟังก์ชั่นแล้วได้คำตอบตัวที่ 1 มาแล้วก็ส่งค่าไปใหม่เพื่อคำนวณหาค่าคำตอบที่ 2 และได้คำตอบที่ 2 ออกมาซึ่งคำตอบ 1 กับ 2 นั้นไม่ใกล้เคียงกันก็คือต่างกันมากกว่าค่าผิดพลาดที่เรากำหนดไว้ (0.00001) ทำให้ case นี้มี 2 คำตอบ จากนั้นให้แสดงค่าของคำตอบตัวที่ 2 ออกมาด้วยและจะแสดงคำถามถามผู้ใช้ว่าจะคำนวณต่อหรือไม่ถ้าผู้ใช้ตอบ y คือคำนวณอีกรอบแต่ถ้าผู้ใช้ตอบ n คือออกจากโปรแกรม

```
==-> Enter The Value [a] : 2
==-> Enter The Value [b] : 2
==-> Enter The Value [c] : 5
============================================================
==========-->>    Can't find Answer    <<--=================
*****======--->> Calculate again ? [y/n] :  y
```

Case 4 : ในกรณีนี้นั้นจะทำงานคล้ายกับกรณีที่ 2 ต่างกันตรงที่ค่าที่ส่งไปคิดจึงทำให้มีการทำงานแค่บางส่วนที่ทำแตกต่างกันโดยกรณีนี้เมื่อส่งค่าไปคิดในฟังก์ชั่น NR แล้วก็จะเช็คเงื่อนไขส่วนเป็น 0 หรือไม่และก็ส่งไปคิดใน else ต่อแต่ case นี้ค่าของคำตอบที่ทำให้สมการเป็นจริงนั้นไม่ใกล้เคียงกัน คือค่าผิดพลาดของ x ตัวใหม่และ x ตัวเก่านั้นต่างกันมากกว่า 0.00001 ทำให้ทำงานวนรอบไปเรื่อยๆ จบเกิน 1000 รอบและทำให้ไม่มีคำตอบของสมการหลังจากนั้นจึงรีเทิร์นค่า 0 กลับมาทำให้ฟังก์ชั่น

main ได้ค่าของ count = 0 และแสดงผลว่าไม่สามารถหาคำตอบของสมการนั้นได้ หลังจากนั้นจะแสดงคำถามถามผู้ใช้ว่าจะคำนวณต่อหรือไม่ถ้าผู้ใช้ตอบ y คือคำนวณอีกรอบแต่ถ้าผู้ใช้ตอบ n คือออกจากโปรแกรม

```
==-> Enter The Value [a] : 1
==-> Enter The Value [b] : -2
==-> Enter The Value [c] : 1
============================================================
==--> Answer of this Quadratic Equation : 1.000007 <--==
==--> And : 0.999993 <--==
*****======--->> Calculate again ? [y/n] :  y
```

Case 5 : case นี้เหมือนกันกับ case 2 และ 3 แตกต่างที่ค่าที่นำไปคิดที่ฟังก์ชั่น NR โดย case นี้เมื่อนำไปคิดในฟังก์ชั่นแล้วได้คำตอบตัวที่ 1 มาแล้วก็ส่งค่าไปใหม่เพื่อคำนวณหาค่าคำตอบที่ 2 และได้คำตอบ ที่ 2 ออกมาซึ่งคำตอบ 1 กับ 2 นั้นไม่ใกล้เคียงกันก็คือต่างกันมากกว่าค่าผิดพลาดที่เรากำหนดไว้ (0.00001) ทำให้ case นี้มี 2 คำตอบ จากนั้นให้แสดงค่าของคำตอบตัวที่ 2 ออกมาด้วยและจะแสดงคำถามถามผู้ใช้ว่าจะคำนวณต่อหรือไม่ถ้าผู้ใช้ตอบ y คือคำนวณอีกรอบแต่ถ้าผู้ใช้ตอบ n คือออกจากโปรแกรม

```
==-> Enter The Value [a] : 1
==-> Enter The Value [b] : -1
==-> Enter The Value [c] : 1
============================================================
-------------->>    Can't find Answer    <<--------------------
*****======--->> Calculate again ? [y/n] :
```

Case 6 : ในกรณีนี้นั้นจะทำงานคล้ายกับกรณีที่ 2 กับ 4 ต่างกันตรงที่ค่าที่ส่งไปคิดจึงทำให้มีการทำงานแค่บางส่วนที่ทำแตกต่างกันโดยกรณีนี้เมื่อส่งค่าไปคิดในฟังก์ชั่น NR แล้วก็จะเช็คเงื่อนไขส่วนเป็น 0 หรือไม่และก็ส่งไปคิดใน else ต่อแต่ case นี้ค่าของคำตอบที่ทำให้สมการเป็นจริงนั้นไม่ใกล้เคียงกันคือค่าผิดพลาดของ x ตัวใหม่และ x ตัวเก่านั้นต่างกันมากกว่า 0.00001 ทำให้ทำงานวนรอบไปเรื่อยๆจบเกิน 1000 รอบและทำให้ไม่มีคำตอบของสมการหลังจากนั้นจึงรีเทิร์นค่า 0 กลับมาทำให้ฟังก์ชั่น main ได้ค่าของ count = 0 และแสดงผลว่าไม่สามารถหาคำตอบของสมการนั้น

ได้ หลังจากนั้นจะแสดงคำถามถามผู้ใช้ว่าจะคำนวณต่อหรือไม่ถ้าผู้ใช้ตอบ y คือคำนวณอีกรอบแต่ถ้าผู้ใช้ตอบ n คือออกจากโปรแกรม

## ประเมินตัวเอง


ให้ตัวเอง 80 เพราะเข้าใจและสามารถทำโจทย์ได้ด้วยตัวเองแตต่อาจมีปัญหานิดหน่อยระหว่างการทำ Assignment แต่ก็สามารถแก้ไขได้ด้วยตัวเองและรู้ที่มาของปัญหาอย่างชัดเจน