

แผนการสอนปฏิบัติการครั้งที่ 6

- **วัตถุประสงค์** เข้าใจการส่งค่าผ่านทางพารามิเตอร์ และ pointer
- **โจทย์ปัญหา** สร้างฟังก์ชันที่คำนวณทางพารามิเตอร์ และมีการตรวจสอบความสำเร็จ
- **ผลลัพธ์การเรียนรู้** เข้าใจการทำงานของโปรแกรม และแก้ไขความผิดพลาด
- **ขั้นตอน**

- พิจารณาหาความผิดพลาดของโปรแกรมหาคำตอบของรากสมการสอง ที่ใช้เทคนิคของ newton-raphson ที่กำหนดให้ เพื่อเขียนใหม่
 - กรณีเรียกฟังก์ชัน diff แล้วตัวหารเป็นศูนย์ ($2ax+b=0$)
 - กรณีที่สมการไม่มีคำตอบ จะวนรอบไม่รู้จบ
 - กรณีสมการคำตอบมี 2 ค่า แต่โปรแกรมหาคำตอบได้เพียงค่าเดียว

- ทดสอบการเรียกใช้ฟังก์ชัน

- อ่านค่าพารามิเตอร์ a, b, c แล้วส่งไปให้ฟังก์ชัน เพื่อหาคำตอบ
- 0,0,5 : 0,5,4 : 2,5,2 : 2,2,5 : 1,-2,1 : 1,-1,1
- อธิบายผิดพลาดที่เกิดขึ้นจากเทสเคส

- ปรับปรุงฟังก์ชันให้ return ความสำเร็จในการหาคำตอบและส่งคำตอบกลับทาง *Xn

int Newton_Raphson (double a, double b, double c, double err, double Xp, double *Xn)

- ตรวจสอบถูกต้องของคำตอบที่หาได้ไม่ให้เป็น $\pm err$
- return 0 เมื่อหาคำตอบไม่ได้ และ return > 0 เมื่อหาคำตอบได้เป็น Xn
- เพิ่มเงื่อนไขตรวจสอบ ไม่ให้หารด้วย 0 ก่อนส่งไปหา diff ถ้าเท่ากับศูนย์ให้ปรับค่า Xp ใหม่ (บวกเพิ่ม)
- เพิ่มการนับจำนวนรอบที่ทำงาน(count) และปรับปรุงเงื่อนไขการหาคำตอบ ไม่ให้มากเกินไป (ไม่เกิน 10000 รอบ)

if (fabs(X1-X2) <= 0.00001)

- ถ้าหาคำตอบได้ (count < 1000) ให้ return จำนวนรอบ count
- ถ้าวนรอบถึง 1000 รอบแล้วยังไม่ได้คำตอบ ให้ return 0

- ให้ทดสอบหาคำตอบ 2 ครั้ง โดยตั้งค่า Xp เป็น -100.123 และ +100.123
 - ถ้าคำตอบเดียวกัน(ใกล้กัน)ให้แสดงค่าเดียว ถ้าคนละคำตอบให้แสดงทั้ง 2 ค่า

- งานที่ส่งตามหลัง

- ตัวอย่างหน้าจอ และคำอธิบายเทสเคสทั้ง 6 กรณี

ค่าเริ่มต้นทำงาน

ค่าผิดพลาด

คำตอบที่รับกลับ

if (2*a*Xp+b==0)
Xp = Xp+0.00001;

ผิดพลาดไม่เกิน 0.00001

count = Newton_Raphson (a, b, c, 0.00001, -100.123, &X1);
count = Newton_Raphson (a, b, c, 0.00001, 100.123, &X2);