



# Calculate-of-epicenter



## ที่มาและความสำคัญ



จากการศึกษาในวิชาธรณีถึงวิธีการคำนวณหาจุด epicenter ศูนย์กลางการเกิดแผ่นดินไหว ทำให้เราเกิดความสนใจในการจัดทำโปรแกรมสำหรับการคำนวณ จุด epicenter เพื่อความสะดวก รวดเร็วและแม่นยำยิ่งกว่าการใช้ กระดาษ วงเวียน ดินสอ ในการคำนวณทาง ธรณีวิทยา

## ประโยชน์ที่คาดว่าจะได้รับ



- 1.สามารถคำนวณหาจุด epicenter ได้แม่นยำมากกว่าการคำนวณด้วยบุคคล
- 2.สามารถคำนวณหาจุด epicenter ได้อย่างรวดเร็ว3.สามารถคำนวณหาจุด epicenter ได้อย่างสะดวก มีขั้นตอนในการได้มาซึ่งคำตอบไม่ยุ่งยาก

## วัตถุประสงค์ของโครงการ



- สามารถระบุจุดเหนือศูนย์กลางเกิดแผ่นดินไหว (Epicenter) ได้อย่างถูกต้อง และใกล้เคียงกับธรรมชาติมากที่สุด
- สามารถคำนวณหาจุดเหนือศูนย์กลางเกิดแผ่นดินไหว (Epicenter) ได้อย่างสะดวก มีขั้นตอนการ ได้มาซึ่งคำตอบที่แม่นยำ
- สามารถใช้เป็นข้อมูลการเฝ้าระวังและป้องกันภัยที่ อาจเกิดขึ้นภายหลังเหตุการณ์แผ่นดินไหวได้

## คำอธิบายโปรแกรม



- 1.เปิดโปรแกรม และเลือกเมนูตามที่ต้องการในแต่ละ หมายเลข 1-4
- 2.ใส่ข้อมูลให้ถูกต้องตามคำสั่งที่โปรแกรมต้องการ
- 3.โปรแกรมจะรับข้อมูลออกมาและแสดงผลทางหน้าจอ

```
int getEasting()
{
    double easting = 500000.0 + (K4 * p + K5 * pow(p, 3.0));
    //easting=(int)easting;
    return easting;
}

double Calculate (double h1,double k1,double r1,double h2,double k2,double r2,double h3,double k3,double r3)
{
    double epicenter_[50];
    epicenter_[0]=(((pow(r1,2)-pow(r2,2)-pow(h1,2)+pow(h2,2)-pow(k1,2)+pow(k2,2))* (2*(k3-k2)))-((pow(r1,2)-pow(r2,2)-pow(h1,2)+pow(h2,2)-pow(k1,2)+pow(k2,2))- (4*(h2-h1)*(k3-k2))* (k3-k2)));
    return epicenter_[0],epicenter_[1];
}

void CheckEpicenter(int x, int y, int h1, int k1, int r1)
{
    if((pow ((x-h1),2)+ pow ((y-k1),2)) == pow (r1,2))
    {
        printf("The epicenter is %d m. from Prime Meridian and %d m. from the Equator\n",x,y);
    }
    else
    {
        printf("The stations haven't joint epicenter\n");
    }
}
```

