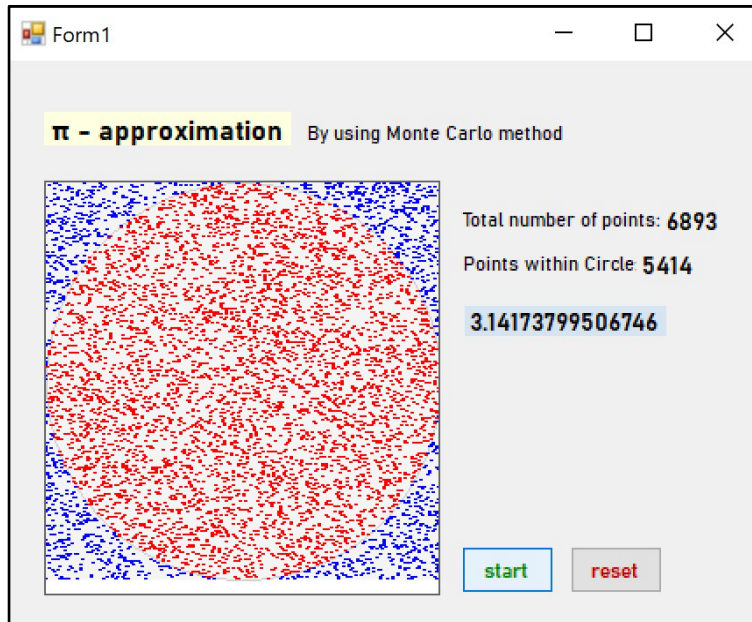


1. พัฒนาโปรแกรม เพื่อหาค่าประมาณของค่าคงที่ Pi



แนวคิด: ประมาณค่าคงที่ Pi ($\pi \approx 3.141592...$) โดยใช้ Monte Carlo Method

โดยจะสุ่มตัวเลขคู่อันดับ (x,y) มา n จำนวน เพื่อ plot จุดทั้งหมดลงบนพื้นที่สี่เหลี่ยมจัตุรัส

เทียบสัดส่วนของจำนวนจุดทั้งหมดที่อยู่ภายในวงกลม (จุดสีแดง) และ จำนวนจุดทั้งหมดในพื้นที่สี่เหลี่ยมจัตุรัส (n จุด)

กำหนดให้ ความยาวด้านของสี่เหลี่ยมจัตุรัสนี้ ยาว $d = x$ หน่วย

จะได้ว่า วงกลมที่อยู่ภายในสี่เหลี่ยมจัตุรัสนี้มีรัศมี $r = x/2$ หน่วย

จาก สูตรหาพื้นที่วงกลม: $A = \pi r^2$

จะได้ พื้นที่วงกลม = $\pi * (x/2) * (x/2) = \pi * (x^2/4)$ ---(1)

และ สูตรหาพื้นที่สี่เหลี่ยมจัตุรัส: $A = d^2$

จะได้ พื้นที่สี่เหลี่ยมจัตุรัส = x^2 ---(2)

นำสมการ (1) / (2) จะได้ พื้นที่วงกลม / พื้นที่สี่เหลี่ยมจัตุรัส = $\pi / 4$

$$\pi = 4 * (\text{พื้นที่วงกลม} / \text{พื้นที่สี่เหลี่ยมจัตุรัส})$$

เมื่อ n มีค่ามากๆ เราจะประมาณพื้นที่วงกลมและพื้นที่สี่เหลี่ยมได้จาก จำนวนจุดที่ plot

$$\pi \approx 4 * (\text{จำนวนจุดในวงกลม} / \text{จำนวนจุดทั้งหมด})$$

เราสามารถหาค่าประมาณค่าคงที่ π ได้จาก 4 เท่าของอัตราส่วนจำนวนจุดภายในวงกลมต่อจำนวนจุดทั้งหมด

การทำงานของโปรแกรม:

```
1 reference
38 private void start_Click(object sender, EventArgs e)
39 {
40     Random rd = new Random();
41     while (plot)
42     {
43         int x = rd.Next(225);
44         int y = rd.Next(225);
45         Point p = new Point(x, y);
46         points.Add(p);
47         totalPointCount++;
48
49         double d = Math.Sqrt(Math.Pow(x - 112.5, 2) + Math.Pow(y - 112.5, 2));
50         if (d <= 112.5) pointInCircle++;
51
52         piEst = 4 * (double) pointInCircle / totalPointCount;
53
54         totalPoint.Text = totalPointCount.ToString();
55         inCircle.Text = pointInCircle.ToString();
56         label5.Text = " $\pi \approx$  " + piEst.ToString();
57
58         Refresh();
59     }
}
```

```
1 reference
69 private void pictureBox1_Paint(object sender, PaintEventArgs e)
70 {
71     Graphics g = e.Graphics;
72     g.FillRectangle(Brushes.WhiteSmoke, 0, 0, 225, 225);
73     g.DrawEllipse(Pens.LightGray, 0,0,225, 225);
74     foreach (Point pt in points){
75         double d = Math.Sqrt(Math.Pow(pt.X - 112.5, 2) + Math.Pow(pt.Y - 112.5, 2));
76         if (d <= 112.5)
77         {
78             g.FillEllipse(Brushes.Red, pt.X - 1, pt.Y - 1, 2, 2);
79         }
80         else
81         {
82             g.FillEllipse(Brushes.Blue, pt.X - 1, pt.Y - 1, 2, 2);
83         }
84     }
85 }
86 }
```

เมื่อผู้ใช้งานกด start โปรแกรมจะทำการสุ่มคู่อันดับ (x,y) n จุด เก็บไว้ใน List<Point>
นับจำนวนจุดทั้งหมด (totalPointCount) และ จุดที่อยู่ภายในวงกลม (pointInCircle) และแสดงผลเป็นค่าประมาณที่คำนวณได้จากสมการข้างต้น (piEst)

จะเห็นได้ว่าเมื่อเวลาผ่านไป n มีค่ามากๆ ค่าประมาณที่คำนวณได้จะมีค่าเข้าใกล้ค่าคงที่ $\pi \approx 3.141592...$

ในส่วนนี้ จะแสดงผลทาง pictureBox โดยทำการ plot จุดที่อยู่ภายในวงกลม และ จุดที่อยู่ภายนอกวงกลมด้วยสีแดงและน้ำเงินตามลำดับ

2. พัฒนาโปรแกรม เพื่อคำนวณหาเปอร์เซ็นต์ที่จะมีเพื่อนในชั้นเรียนเกิดวันเดียวกัน

แนวคิด: รับจำนวนเต็ม n (จำนวนนักเรียนในห้องเรียน) ทำการทดลองโดยการสุ่มตัวเลขจำนวนเต็มแทนวันเกิด (จำนวนเต็ม 0, 1, 2, 3, ... , 364, 365) ให้กับนักเรียน n คน และนับจำนวนห้องเรียนที่มีนักเรียนเกิดวันเดียวกันอย่างน้อยหนึ่งคน

The three screenshots show the 'Birthday' application for different numbers of students: 34, 40, and 50. Each window has a title bar 'Form1', a 'Birthday' icon, a '#students' input field, a 'submit' button, a 'clear' button, and an 'output' label showing a percentage.

#students	Output Percentage
34	78.50 %
40	89.50 %
50	96.90 %

การทำงานของโปรแกรม:

เมื่อผู้ใช้งาน submit จำนวนนักเรียนในห้องเรียน จะทำการวนลูปเพื่อสุ่มวันเกิดให้กับนักเรียนทุกคนในห้องเรียน และตรวจสอบว่าในห้องเรียนนั้นมีคนเกิดวันเดียวกันหรือไม่ ทำซ้ำ 1000 รอบ นับจำนวนแบบ ที่ในห้องเรียนมีอย่างน้อยหนึ่งคู่ที่มีนักเรียนเกิดวันเดียวกัน คำนวณและแสดงผลในรูปแบบเปอร์เซ็นต์

```
20 private void textBox2_TextChanged(object sender, EventArgs e)
21 {
22 }
23
24 int cases = 1000;
25 int count;
26 int n;
27
28 private bool CheckRandBirth(int n)
29 {
30     HashSet<int> data = new HashSet<int>();
31     Random rd = new Random();
32     for (int i=0; i<n; i++)
33     {
34         data.Add(rd.Next(365));
35     }
36     return data.Count() != n;
37 }
```

```
1 reference
41 private void button1_Click(object sender, EventArgs e)
42 {
43     count = 0;
44     if (int.TryParse(n_students.Text, out n))
45     {
46         for(int i=0; i<cases; i++)
47         {
48             if (CheckRandBirth(n))
49             {
50                 textBox1.Text += "\r\n" + (i + 1).ToString() + " YES ";
51                 count++;
52             }
53             else
54             {
55                 textBox1.Text += "\r\n" + (i + 1).ToString() + " NO ";
56             }
57         }
58         label3.Text = (((double) count / cases) * 100).ToString("0.00") + " %";
59     }
60     else
61     {
62         MessageBox.Show("Invalid!");
63     }
64 }
65 }
```