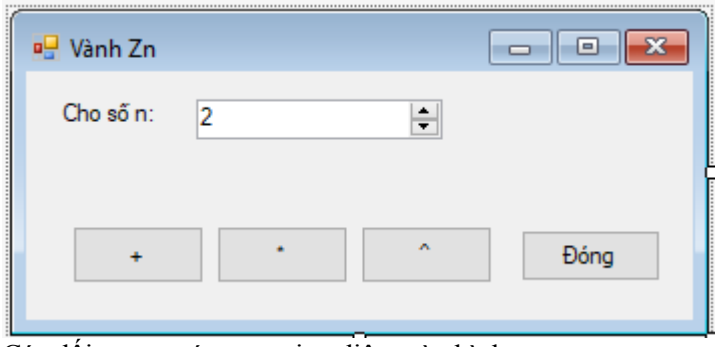
Bài thực hành 4.1 – Đại số hữu hạn

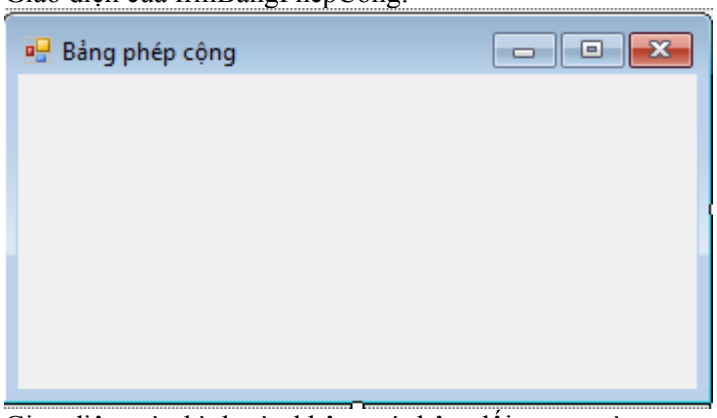
1. Anh/chị tạo một đề án mới bằng C#, có tên BaiDaiSoHuuHan
2. Anh/chị sửa ngay Form1.cs thành MainForm.cs, đồng thời sửa thành phần Text của nó thành "Đại số hữu hạn".
3. Anh.chị sửa lịa thuộc tính IsMdiContainer của MainForm thành True.
4. Trong MainForm có một đối tượng MenuStripBar, với tên MainMenu.
5. Trong MainMenu có 2 mục thực đơn chính:
   - Mục thực đơn chính thứ nhất, có tên mnuVanh, với thành phần Text là "Vành". Mục này có một mục thực đơn con:
     - Mục thực đơn con có tên: mnuVanhZn, với thành phần Text là "Vành Zn".
   - Mục thực đơn chính thứ hai, có tên mnuTruong, với thành phần Text là "Trường". Mục này có một thực đơn con:
     - Mục thực đơn com có tên mnuTruongQp, với thành phần Text là :Trường Qp".
6. Anh/chị tạo thêm 5 form con, lần lượt có tên:
   - frmVanhZn
   - frmTruongQp
   - frmBangPhepCong
   - frmBangPhepNhan
   - frmBangLuyThua
7. Giao diện của frmVanhZn như sau:



Các đối tượng có trong giao diện màn hình:

STT	Tên đối tượng	Loại đối tượng	Text	Các thuộc tính khác	Vị trí
1	lblN	Label	Cho số n		Dòng đầu tiên
2	nudN	NumericUpDown		Value=2 Minimun=2 Maximum=65535	Bên phải lblN
3	btnCong	Button	+		Ngay dưới lblN
4	btnTru	Button	-		Bên phải btnCong
5	btnLuyThua	Button	^		Bên phải btnNhan
6	btnDong	Button	Đóng		Bên phải btnLuyThua

8. Giao diện của frmBangPhepCong:



Giao diện màn hình này không có thêm đối tượng nào.

9. Trong frmBangPhep cong có:

- Khai báo biến: Có 6 biến riêng tư và 6 hằng riêng tư:
    - Biến riêng tư thứ nhất, có tên code, với kiểu ký tự và có trị ban đầu 'Z'.
    - Biến riêng tư thứ hai, có tên n, với kiểu số nguyên, và giá trị ban đầu 2.
    - Hằng riêng tư thứ nhất, có tên Row_Start, với kiểu số nguyên, và có giá trị 10.
    - Hằng riêng tư thứ hai, có tên Col_Start, với kiểu số nguyên, và có giá trị 50.
    - Hằng riêng tư thứ ba, có tên Cell_Width, với kiểu số nguyên, và có giá trị 30.
    - Hằng riêng tư thứ tư, có tên Cell_Vgap, với kiểu số nguyên, và có giá trị 5.
    - Hằng riêng tư thứ năm, có tên Row_Height, với kiểu số nguyên, và có giá trị 20.
    - Hằng riêng tư thứ sáu, có tên Cell_Hgap, với kiểu số nguyên, và có giá trị 2.
    - Biến riêng tư thứ ba, có tên txtPhepToan, với kiểu TextBox.
    - Biến riêng tư thứ tư, có tên txtCot, với kiểu mãng một chiều các phần tử TextBox.
    - Biến riêng tư thứ năm, có tên txtDong, với kiểu mãng một chiều các phần tử TextBox.
    - Biến riêng tư thứ sáu, có tên txtCell, với kiểu mảng hai chiều các phần tử TextBox.
- Các hàm tạo: Có hai hàm tạo:
    - Hàm tạo không tham biến. Hàm tạo này có sẵn một lệnh InitisalizeComponent(). Anh/chị thêm sau lệnh này một lệnh mới:

```
TaoBangPhepToan(n);
```

    - Hàm tạo hai tham biến: Tham biến hình thức thứ nhất, có tên c, với kiểu ký tự. Tham biến hình thức thứ hai có tên n0, với kiểu số nguyên. Nhiệm vụ của hàm tạo này là:

```
code = c;
n = n0;
InitializeComponent();
if (n>20)
   TaoLuoi(n);
else
   TaoBangPhepToan(n);
```

- Các phương thức: Có 3 phương thức riêng tư.
    - Phương thức riêng tư thứ nhất, có tên TaoBangPhepToan. Phương thức này có một tham biến hình thức, có tên n, với kiểu số nguyên. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Text = "Bảng phép cộng "+ MainForm.Ten(code) + n.ToString();
TextBox Cell;
this.SuspendLayout();

// Phép toán
Cell = new System.Windows.Forms.TextBox();
Cell.Location = new System.Drawing.Point(Col_Start, Row_Start);
Cell.Name = "Cell";
Cell.Size = new System.Drawing.Size(Cell_Width, 20);
Cell.ReadOnly = true;
Cell.TextAlign = HorizontalAlignment.Center;
this.Controls.Add(Cell);
txtPhepToan = Cell;

// Các ô hàng đầu tiên
txtCot = new TextBox[n];
for (int j = 0; j < n; j++)
{
   Cell = new System.Windows.Forms.TextBox();
   Cell.Location = new System.Drawing.Point(Col_Start + (Cell_Width + Cell_VGap) * (j + 1), Row_Start);
   Cell.Name = "Cell";
   Cell.Size = new System.Drawing.Size(Cell_Width, 20);
   Cell.ReadOnly = true;
   Cell.TextAlign = HorizontalAlignment.Center;
   this.Controls.Add(Cell);
   txtCot[j] = Cell;
}
```

```
// Các hàng chi tiết
txtDong = new TextBox[n];
txtCell = new TextBox[n, n];

for (int i = 0; i < n; i++)
{
   // Ô trên cột đầu tiên
   Cell = new System.Windows.Forms.TextBox();
   Cell.Location = new System.Drawing.Point(Col_Start, Row_Start + (Row_Height + Cell_HGap) * (i + 1));
   Cell.Name = "Cell";
   Cell.Size = new System.Drawing.Size(Cell_Width, 20);
   Cell.ReadOnly = true;
   Cell.TextAlign = HorizontalAlignment.Center;
   this.Controls.Add(Cell);
   txtDong[i] = Cell;

   // Các ô chi tiết
   for (int j = 0; j < n; j++)
   {
      Cell = new System.Windows.Forms.TextBox();
      Cell.Location = new System.Drawing.Point(Col_Start + (Cell_Width + Cell_VGap) * (j + 1),
                  Row_Start + (Row_Height + Cell_HGap) * (i + 1));
      Cell.Name = "Cell";
      Cell.Size = new System.Drawing.Size(Cell_Width, 20);
      Cell.ReadOnly = true;
      Cell.TextAlign = HorizontalAlignment.Center;
      this.Controls.Add(Cell);
      txtCell[i, j] = Cell;
   }
}
this.AutoScroll = true;

this.ResumeLayout();
TinhToan();
```

- Phương thức riêng tư thứ hai, có tên TinhToan. Phương thức này không có tham biến hình thức. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

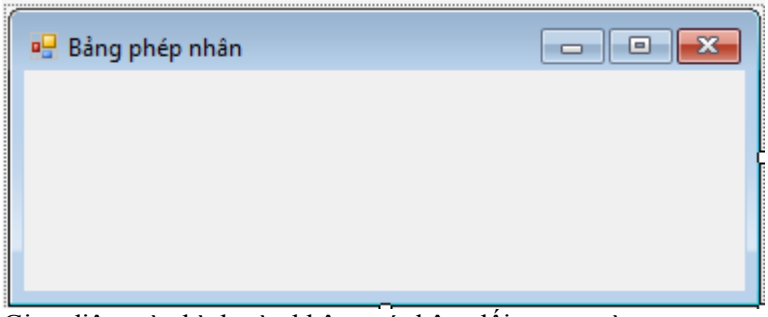
```
txtPhepToan.Text = "+";

// Điền hàng đầu tiên
for (int j = 0; j < n; j++)
   txtCot[j].Text = j.ToString();

// Điền các hàng chi tiết
for(int i=0; i<n; i++)
{
   // Điền cột đầu tiên của dòng chi tiết
   txtDong[i].Text = i.ToString();

   // Điền các ô chi tiết
   for (int j = 0; j < n; j++)
      txtCell[i, j].Text = ((i + j) % n).ToString();
}
```

- Phương thức riêng tư thứ ba, có tên TaoLuoi. Phương thức này có một tham biến hình thức, có tên n, với kiểu số nguyên. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Text = "Bảng phép cộng " + MainForm.Ten(code) + n.ToString();
```

```
DataGridView dgv = new DataGridView();
dgv.Location = new System.Drawing.Point(Col_Start, Row_Start);
dgv.Name = "dgv";
//dgv.Size = new System.Drawing.Size(Cell_Width, 20);
dgv.Dock = DockStyle.Fill;
this.Controls.Add(dgv);

dgv.AllowUserToAddRows = false;
dgv.RowHeadersVisible = false;

DataGridViewTextBoxColumn dgvCol;

dgvCol = new DataGridViewTextBoxColumn();
dgvCol.Name = "operator";
dgvCol.HeaderText = "+";
dgvCol.Width = 50;
dgv.Columns.Add(dgvCol);

for (int j = 0; j < n; j++)
{
   dgvCol = new DataGridViewTextBoxColumn();
   dgvCol.Name = "Col" + j.ToString("0000");
   dgvCol.HeaderText = j.ToString();
   dgvCol.Width = 50;
   dgvCol.FillWeight = 0.01f;
   dgv.Columns.Add(dgvCol);
}

for (int i = 0; i < n; i++)
{
   DataGridViewRow dgvRow = new DataGridViewRow();

   DataGridViewTextBoxCell dgvCell = new DataGridViewTextBoxCell();
   dgvCell.Value = i.ToString();
   dgvRow.Cells.Add(dgvCell);

   for (int j = 0; j < n; j++)
   {
      dgvCell = new DataGridViewTextBoxCell();
      dgvCell.Value = ((i + j) % n).ToString();
      dgvRow.Cells.Add(dgvCell);
   }
   dgv.Rows.Add(dgvRow);
}
```

10. Giao diện của frmBangPhepCong như sau:



Giao diện màn hình này không có thêm đối tượng nào.

11. Trong frmBangPhep có:

- Khai báo biến: Có 6 biến riêng tư và 6 hằng riêng tư:
    - Biến riêng tư thứ nhất, có tên code, với kiểu ký tự và có trị ban đầu 'Z'.
    - Biến riêng tư thứ hai, có tên n, với kiểu số nguyên, và giá trị ban đầu 2.
    - Hằng riêng tư thứ nhất, có tên Row_Start, với kiểu số nguyên, và có giá trị 10.
    - Hằng riêng tư thứ hai, có tên Col_Start, với kiểu số nguyên, và có giá trị 50.
    - Hằng riêng tư thứ ba, có tên Cell_Width, với kiểu số nguyên, và có giá trị 30.
    - Hằng riêng tư thứ tư, có tên Cell_Vgap, với kiểu số nguyên, và có giá trị 5.
    - Hằng riêng tư thứ năm, có tên Row_Height, với kiểu số nguyên, và có giá trị 20.
    - Hằng riêng tư thứ sáu, có tên Cell_Hgap, với kiểu số nguyên, và có giá trị 2.
    - Biến riêng tư thứ ba, có tên txtPhepToan, với kiểu TextBox.
    - Biến riêng tư thứ tư, có tên txtCot, với kiểu mãng một chiều các phần tử TextBox.
    - Biến riêng tư thứ năm, có tên txtDong, với kiểu mãng một chiều các phần tử TextBox.
    - Biến riêng tư thứ sáu, có tên txtCell, với kiểu mảng hai chiều các phần tử TextBox.
- Các hàm tạo: Có hai hàm tạo:
    - Hàm tạo không tham biến. Hàm tạo này có sẵn một lệnh InitisalizeComponent(). Anh/chị thêm sau lệnh này một lệnh mới:

```
TaoBangPhepToan(n);
```

    - Hàm tạo hai tham biến: Tham biến hình thức thứ nhất, có tên c, với kiểu ký tự. Tham biến hình thức thứ hai có tên n0, với kiểu số nguyên. Nhiệm vụ của hàm tạo này là:

```
code = c;
n = n0;
InitializeComponent();
if (n > 20)
   TaoLuoi(n);
else
   TaoBangPhepToan(n); Các phương thức:
```

- Các phương thức: Có 3 phương thức riêng tư.
    - Phương thức riêng tư thứ nhất có tên TaoBangPhepToan. Phương thức này có một tham biến hình thức, có tên n, với kiểu số nguyên. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Text = "Bảng phép nhân "+ MainForm.Ten(code) + n.ToString();
TextBox Cell;
this.SuspendLayout();

// Phép toán
Cell = new System.Windows.Forms.TextBox();
Cell.Location = new System.Drawing.Point(Col_Start, Row_Start);
Cell.Name = "Cell";
Cell.Size = new System.Drawing.Size(Cell_Width, 20);
Cell.ReadOnly = true;
Cell.TextAlign = HorizontalAlignment.Center;
this.Controls.Add(Cell);
txtPhepToan = Cell;

// Các ô hàng đầu tiên
txtCot = new TextBox[n];
for (int j = 0; j < n; j++)
{
   Cell = new System.Windows.Forms.TextBox();
   Cell.Location = new System.Drawing.Point(Col_Start + (Cell_Width + Cell_VGap) * (j + 1), Row_Start);
   Cell.Name = "Cell";
   Cell.Size = new System.Drawing.Size(Cell_Width, 20);
   Cell.ReadOnly = true;
   Cell.TextAlign = HorizontalAlignment.Center;
   this.Controls.Add(Cell);
   txtCot[j] = Cell;
}
```

```
// Các hàng chi tiết
txtDong = new TextBox[n];
txtCell = new TextBox[n, n];

for (int i = 0; i < n; i++)
{
   // Ô trên cột đầu tiên
   Cell = new System.Windows.Forms.TextBox();
   Cell.Location = new System.Drawing.Point(Col_Start, Row_Start + (Row_Height + Cell_HGap) * (i + 1));
   Cell.Name = "Cell";
   Cell.Size = new System.Drawing.Size(Cell_Width, 20);
   Cell.ReadOnly = true;
   Cell.TextAlign = HorizontalAlignment.Center;
   this.Controls.Add(Cell);
   txtDong[i] = Cell;

   // Các ô chi tiết
   for (int j = 0; j < n; j++)
   {
      Cell = new System.Windows.Forms.TextBox();
      Cell.Location = new System.Drawing.Point(Col_Start + (Cell_Width + Cell_VGap) * (j + 1),
                  Row_Start + (Row_Height + Cell_HGap) * (i + 1));
      Cell.Name = "Cell";
      Cell.Size = new System.Drawing.Size(Cell_Width, 20);
      Cell.ReadOnly = true;
      Cell.TextAlign = HorizontalAlignment.Center;
      this.Controls.Add(Cell);
      txtCell[i, j] = Cell;
   }
}
//  this.AutoScroll = true;

this.ResumeLayout();
TinhToan();
```

- Phương thức riêng tư thứ hai có tên TinhToan. Phương thức này không có tham biến hình thức. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

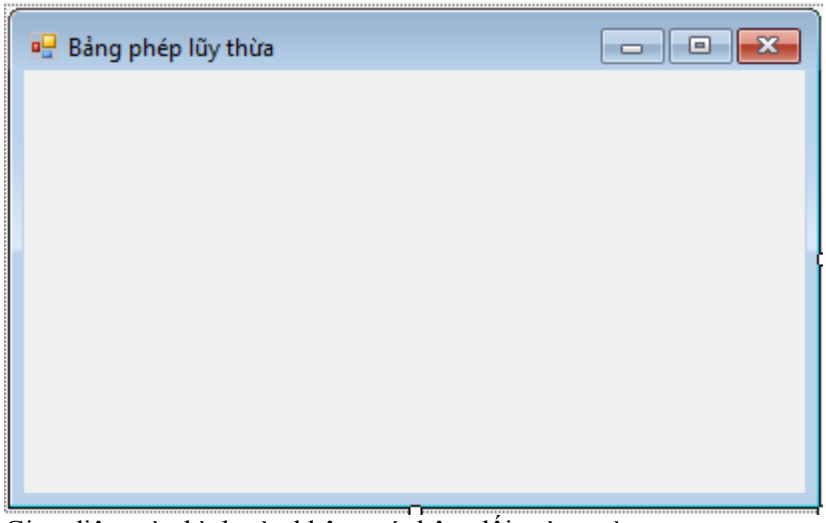
```
txtPhepToan.Text = "*";

// Điền hàng đầu tiên
for (int j = 0; j < n; j++)
   txtCot[j].Text = j.ToString();

// Điền các hàng chi tiết
for (int i = 0; i < n; i++)
{
   // Điền cột đầu tiên của dòng chi tiết
   txtDong[i].Text = i.ToString();

   // Điền các ô chi tiết
   for (int j = 0; j < n; j++)
      txtCell[i, j].Text = ((i * j) % n).ToString();
}
```

- Phương thức riêng tư thứ ba, có tên TaoLuoi. Phương thức này có một tham biến hình thức, có tên n, với kiểu số nguyên. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Text = "Bảng phép nhân "+ MainForm.Ten(code) + n.ToString();
```

```
DataGridView dgv = new DataGridView();
dgv.Location = new System.Drawing.Point(Col_Start, Row_Start);
dgv.Name = "dgv";
//dgv.Size = new System.Drawing.Size(Cell_Width, 20);
dgv.Dock = DockStyle.Fill;
this.Controls.Add(dgv);

dgv.AllowUserToAddRows = false;
dgv.RowHeadersVisible = false;

DataGridViewTextBoxColumn dgvCol;

dgvCol = new DataGridViewTextBoxColumn();
dgvCol.Name = "operator";
dgvCol.HeaderText = "*";
dgvCol.Width = 50;
dgv.Columns.Add(dgvCol);

for (int j = 0; j < n; j++)
{
   dgvCol = new DataGridViewTextBoxColumn();
   dgvCol.Name = "Col" + j.ToString("0000");
   dgvCol.HeaderText = j.ToString();
   dgvCol.Width = 50;
   dgvCol.FillWeight = 0.01f;
   dgv.Columns.Add(dgvCol);
}

for (int i = 0; i < n; i++)
{
   DataGridViewRow dgvRow = new DataGridViewRow();

   DataGridViewTextBoxCell dgvCell = new DataGridViewTextBoxCell();
   dgvCell.Value = i.ToString();
   dgvRow.Cells.Add(dgvCell);

   for (int j = 0; j < n; j++)
   {
      dgvCell = new DataGridViewTextBoxCell();
      dgvCell.Value = ((i * j) % n).ToString();
      dgvRow.Cells.Add(dgvCell);
   }
   dgv.Rows.Add(dgvRow);
}
```

12. Giao diện của frmBangPhepNhan như sau:



Giao diện màn hình này không có thêm đối tường nào.

13. Trong frmBangLuyThua có:
    - Khai báo biến: Có 6 biến riêng tư và 6 hằng riêng tư:
        - Biến riêng tư thứ nhất, có tên code, với kiểu ký tự và có trị ban đầu 'Z'.
        - Biến riêng tư thứ hai, có tên n, với kiểu số nguyên, và giá trị ban đầu 2.
        - Hằng riêng tư thứ nhất, có tên Row_Start, với kiểu số nguyên, và có giá trị 10.
        - Hằng riêng tư thứ hai, có tên Col_Start, với kiểu số nguyên, và có giá trị 50.
        - Hằng riêng tư thứ ba, có tên Cell_Width, với kiểu số nguyên, và có giá trị 30.
        - Hằng riêng tư thứ tư, có tên Cell_Vgap, với kiểu số nguyên, và có giá trị 5.
        - Hằng riêng tư thứ năm, có tên Row_Height, với kiểu số nguyên, và có giá trị 20.
        - Hằng riêng tư thứ sáu, có tên Cell_Hgap, với kiểu số nguyên, và có giá trị 2.
        - Biến riêng tư thứ ba, có tên txtPhepToan, với kiểu TextBox.
        - Biến riêng tư thứ tư, có tên txtCot, với kiểu mãng một chiều các phần tử TextBox.
        - Biến riêng tư thứ năm, có tên txtDong, với kiểu mãng một chiều các phần tử TextBox.
        - Biến riêng tư thứ sáu, có tên txtCell, với kiểu mảng hai chiều các phần tử TextBox.
    - Các hàm tạo: Có hai hàm tạo:
        - Hàm tạo không tham biến. Hàm tạo này có sẵn một lệnh InitisalizeComponent(). Anh/chị thêm sau lệnh này một lệnh mới:

```
TaoBangPhepToan(n);
```

        - Hàm tạo hai tham biến: Tham biến hình thức thứ nhất, có tên c, với kiểu ký tự. Tham biến hình thức thứ hai có tên n0, với kiểu số nguyên. Nhiệm vụ của hàm tạo này là:

```
code = c;
n = n0;
InitializeComponent();
if (n > 20)
   TaoLuoi(n);
else
   TaoBangPhepToan(n);
```

    - Các phương thức: Có 3 phương thức riêng tư.
        - Phương thức riêng tư thứ nhất có tên TaoBangPhepToan. Phương thức này có một tham biến hình thức, có tên n, với kiểu số nguyên. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Text = "Bảng phép lũy thừa " + MainForm.Ten(code) + n.ToString();
TextBox Cell;
this.SuspendLayout();

// Phép toán
Cell = new System.Windows.Forms.TextBox();
Cell.Location = new System.Drawing.Point(Col_Start, Row_Start);
Cell.Name = "Cell";
Cell.Size = new System.Drawing.Size(Cell_Width, 20);
Cell.ReadOnly = true;
```

```
Cell.TextAlign = HorizontalAlignment.Center;
this.Controls.Add(Cell);
txtPhepToan = Cell;

// Các ô hàng đầu tiên
txtCot = new TextBox[n];
for (int j = 0; j < n; j++)
{
   Cell = new System.Windows.Forms.TextBox();
   Cell.Location = new System.Drawing.Point(Col_Start + (Cell_Width + Cell_VGap) * (j + 1), Row_Start);
   Cell.Name = "Cell";
   Cell.Size = new System.Drawing.Size(Cell_Width, 20);
   Cell.ReadOnly = true;
   Cell.TextAlign = HorizontalAlignment.Center;
   this.Controls.Add(Cell);
   txtCot[j] = Cell;
}

// Các hàng chi tiết
txtDong = new TextBox[n];
txtCell = new TextBox[n, n];

for (int i = 0; i < n; i++)
{
   // Ô trên cột đầu tiên
   Cell = new System.Windows.Forms.TextBox();
   Cell.Location = new System.Drawing.Point(Col_Start, Row_Start + (Row_Height + Cell_HGap) * (i + 1));
   Cell.Name = "Cell";
   Cell.Size = new System.Drawing.Size(Cell_Width, 20);
   Cell.ReadOnly = true;
   Cell.TextAlign = HorizontalAlignment.Center;
   this.Controls.Add(Cell);
   txtDong[i] = Cell;

   // Các ô chi tiết
   for (int j = 0; j < n; j++)
   {
      Cell = new System.Windows.Forms.TextBox();
      Cell.Location = new System.Drawing.Point(Col_Start + (Cell_Width + Cell_VGap) * (j + 1),
                        Row_Start + (Row_Height + Cell_HGap) * (i + 1));
      Cell.Name = "Cell";
      Cell.Size = new System.Drawing.Size(Cell_Width, 20);
      Cell.ReadOnly = true;
      Cell.TextAlign = HorizontalAlignment.Center;
      this.Controls.Add(Cell);
      txtCell[i, j] = Cell;
   }
}
this.AutoScroll = true;

this.ResumeLayout();
TinhToan();
```

- Phương thức riêng tư thứ hai có tên TinhToan. Phương thức này không có tham biến hình thức. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là :

```
txtPhepToan.Text = "^";

// Điền hàng đầu tiên
```

```
for (int j = 0; j < n; j++)
   txtCot[j].Text = j.ToString();

// Điền các hàng chi tiết
for (int i = 0; i < n; i++)
{
   // Điền cột đầu tiên của dòng chi tiết
   txtDong[i].Text = i.ToString();

   // Điền các ô chi tiết
   for (int j = 0; j < n; j++)
      if (i == 0) txtCell[i, j].Text = 1.ToString();
      else txtCell[i, j].Text = ((int.Parse(txtCell[i-1, j].Text) * j) % n).ToString();
}
```

- Phương thức riêng tư thứ ba, có tên TaoLuoi. Phương thức này có một tham biến hình thức, có tên n, với kiểu số nguyên. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Text = "Bảng phép lũy thừa " + MainForm.Ten(code) + n.ToString();

DataGridView dgv = new DataGridView();
dgv.Location = new System.Drawing.Point(Col_Start, Row_Start);
dgv.Name = "dgv";
//dgv.Size = new System.Drawing.Size(Cell_Width, 20);
dgv.Dock = DockStyle.Fill;
this.Controls.Add(dgv);

dgv.AllowUserToAddRows = false;
dgv.RowHeadersVisible = false;

DataGridViewTextBoxColumn dgvCol;

dgvCol = new DataGridViewTextBoxColumn();
dgvCol.Name = "operator";
dgvCol.HeaderText = "^";
dgvCol.Width = 50;
dgv.Columns.Add(dgvCol);

for (int j = 0; j < n; j++)
{
   dgvCol = new DataGridViewTextBoxColumn();
   dgvCol.Name = "Col" + j.ToString("0000");
   dgvCol.HeaderText = j.ToString();
   dgvCol.Width = 50;
   dgvCol.FillWeight = 0.01f;
   dgv.Columns.Add(dgvCol);
}

for (int i = 0; i < n; i++)
{
   DataGridViewRow dgvRow = new DataGridViewRow();

   DataGridViewTextBoxCell dgvCell = new DataGridViewTextBoxCell();
   dgvCell.Value = i.ToString();
   dgvRow.Cells.Add(dgvCell);

   for (int j = 0; j < n; j++)
   {
```

```
            dgvCell = new DataGridViewTextBoxCell();
            if (i == 0) dgvCell.Value = 1.ToString();
            else dgvCell.Value = ((int.Parse(dgv.Rows[i-1].Cells[j+1].Value.ToString()) * j) % n).ToString();
            dgvRow.Cells.Add(dgvCell);
        }
        dgv.Rows.Add(dgvRow);
    }
```

14. Trong frmVanhZn có:
    - Khai báo biến: Có một biến riêng tư duy nhất:
        - Biến riêng tư có tên code, với kiểu ký tự, và có giá trị ban đầu là 'Z'.
    - Các hàm tạo: Có một hàm tạo duy nhất.
        - Hàm tạo không tham biến: Nhiệm vụ của hàm tạo này chỉ có một câu lệnh có sẵn InitializeComponent().
    - Các đặc trưng: Không có.
    - Các phương thức: Không có.
    - Các xử lý: Có 4 phương thức riêng tư xử lý biến cố.
        - Phương thức riêng tư thứ nhất, xử lý biến cố Click của btnCong, có tên btnCong_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
int n = (int)(nudN.Value);
frmBangPhepCong frm = new frmBangPhepCong(code, n);
frm.MdiParent = this.MdiParent;
frm.Show();
```

        - Phương thức riêng tư thứ hai, xử lý biến cố Click của btnNhan, có tên btnNhan_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
int n = (int)(nudN.Value);
frmBangPhepNhan frm = new frmBangPhepNhan(code, n);
frm.MdiParent = this.MdiParent;
frm.Show();
```

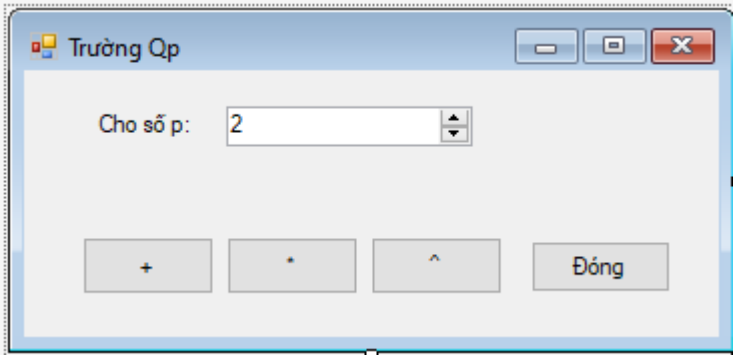
        - Phương thức riêng tư thứ ba, xử lý biến cố Click của btnLuyThua, có tên btnLuyThua_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
int n = (int)(nudN.Value);
frmBangPhepLuyThua frm = new frmBangPhepLuyThua(code, n);
frm.MdiParent = this.MdiParent;
frm.Show();
```

        - Phương thức thứ tư, xử lý biến cố Click của btnDong, có tên btnDong_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Close();
```

15. Giao diện của frmTruongQp như sau:



Các đối tượng có trong giao diện màn hình:

STT	Tên đối tượng	Loại đối tượng	Text	Các thuộc tính khác	Vị trí
1	lblP	Label	Cho số p		Dòng đầu tiên
2	nudP	NumericUpDown		Value=2 Minimun=2 Maximum=65535	Bên phải lblP
3	btnCong	Button	+		Ngay dưới lblP
4	btnTru	Button	-		Bên phải btnCong
5	btnLuyThua	Button	^		Bên phải btnNhan
6	btnDong	Button	Đóng		Bên phải btnLuyThua

16. Trong frmTruongQp có:
    - Khai báo biến: Có một biến riêng tư duy nhất:
        - Biến riêng tư có tên code, với kiểu ký tự, và có giá trị ban đầu là 'Q'.
    - Các hàm tạo: Có một hàm tạo duy nhất.
        - Hàm tạo không tham biến: Nhiệm vụ của hàm tạo này chỉ có một câu lệnh có sẵn InitializeComponent().
    - Các phương thức: Có một phương thức riêng tư duy nhất.
        - Phương thức riêng tư có tên IsPrime. Phương thức này có một tham biến hình thức, có tên n, với kiểu số nguyên. Phương thức này trả về kiểu luận lý. Nhiệm vụ của phương thức này là:

```
int a = 2;
while (a * a <= n)
{
   if (n % a == 0) return false;
   a++;
}
return true;
```

    - Các xử lý: Có 5 phương thức riêng tư xử lý biến cố:
        - Phương thức riêng tư thứ nhất, xử lý biến cố ValueChange của nudP, có tên nudP_ValueChanged. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
bool HopLe = IsPrime((int)nudP.Value);
btnCong.Enabled = btnNhan.Enabled = btnLuyThua.Enabled = HopLe;
```

        - Phương thức riêng tư thứ hai, xử lý biến cố Click của btnCong, có tên btnCong_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
int n = (int)(nudP.Value);
frmBangPhepCong frm = new frmBangPhepCong(code, n);
frm.MdiParent = this.MdiParent;
frm.Show();
```

        - Phương thức riêng tư thứ ba, xử lý biến cố Click của btnNhan, có tên btnNhan_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
int n = (int)(nudP.Value);
frmBangPhepNhan frm = new frmBangPhepNhan(code, n);
frm.MdiParent = this.MdiParent;
frm.Show();
```

        - Phương thức riêng tư thứ tư, xử lý biến cố Click của btnLuyThua, có tên btnLuyThua_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
int n = (int)(nudP.Value);
frmBangPhepLuyThua frm = new frmBangPhepLuyThua(code, n);
frm.MdiParent = this.MdiParent;
```

```
frm.Show();
```

- Phương thức riêng tư thứ năm, xử lý biến cố Click của btnDong, có tên btnDong_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
this.Close();
```

17. Trong MainForm có:
    - Khai báo biến: Không có.
    - Các hàm tạo: Có một hàm tạo duy nhất.
        - Hàm tạo không tham biến. Nhiệm vụ của hàm tạo này là một của hàm tạo này là một câu sẵn có sẵn InitializeComponent().
    - Các phương thức: Có một phươbg thức riêng tư duy nhất.
        - Phương thức riêng tư, có tên Ten. Phương thức này có một tham biến hình thức, có tên code, với kiểu ký tự. Phương thức này trả về kiểu chuỗi. Nhiệm vụ của phương thức này là:

```
return (code == 'Z' )? "Vành Z" : "Trường Q";
```

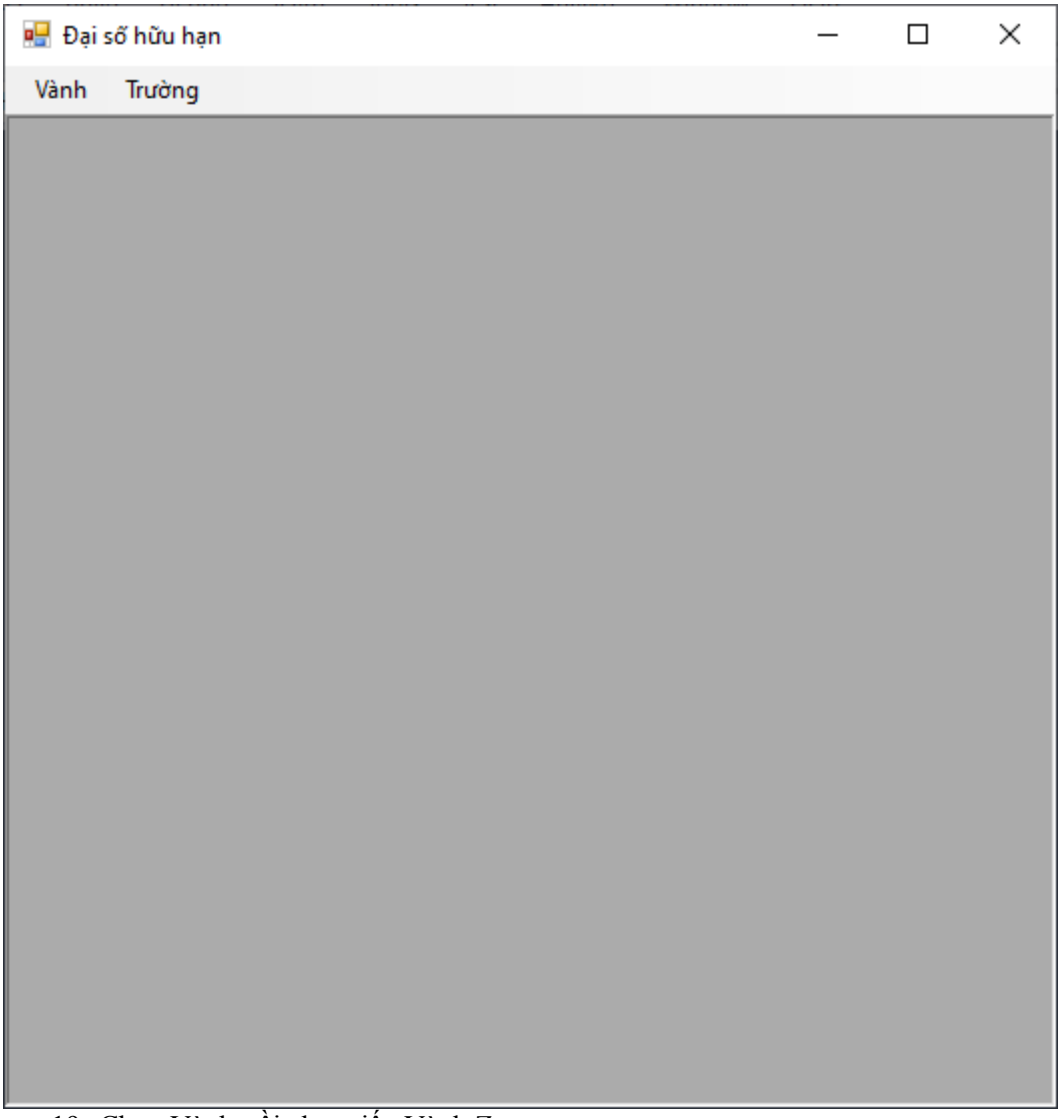
    - Các xử lý: Có 2 phương thức riêng tư xử lý biến cố.
        - Phương thức riêng tư thứ nhất, xử lý biến cố Click của mnuVanhZn, có tên mnuVanhZn_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
frmVanhZn frm = new frmVanhZn();
frm.MdiParent = this;
frm.Show();
```

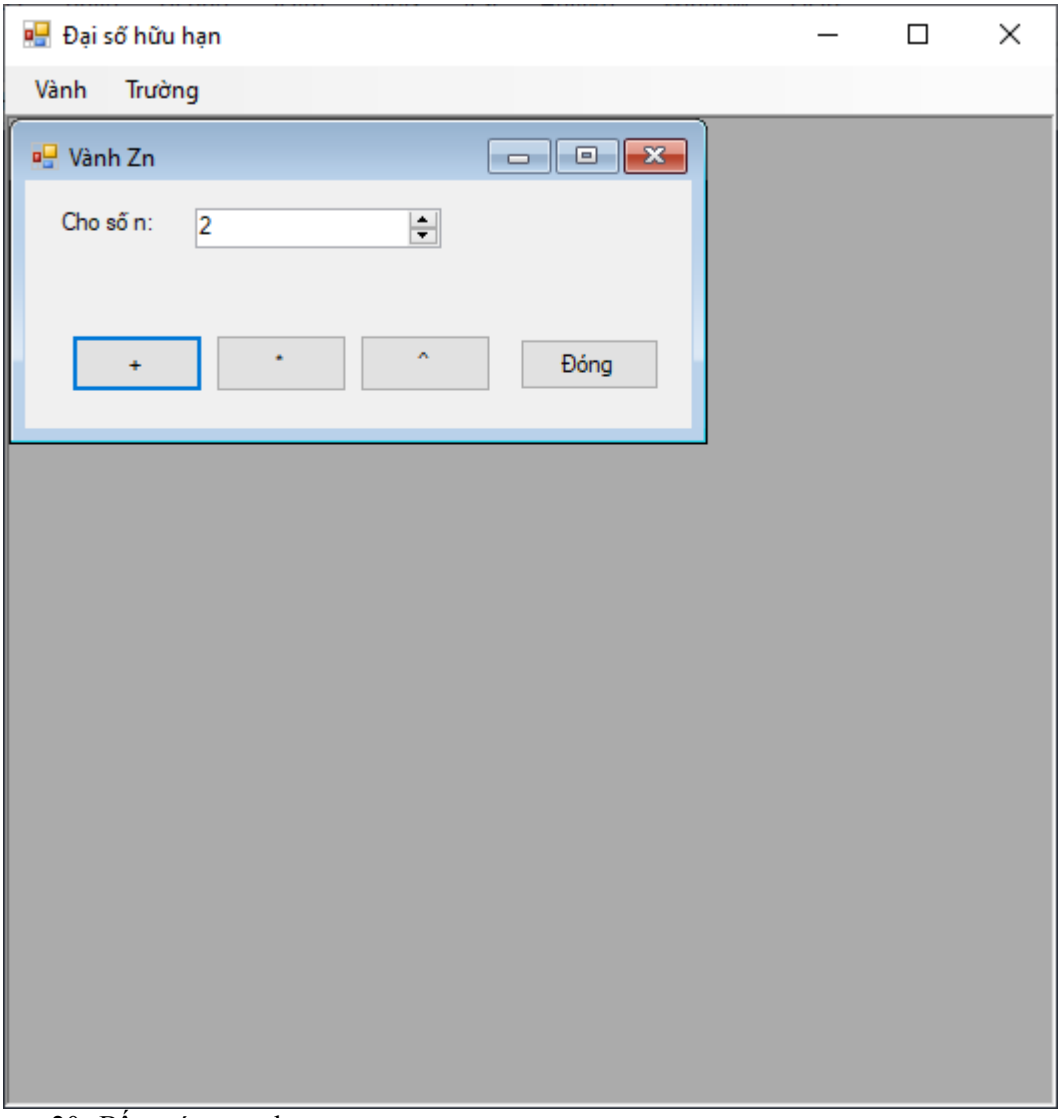
        - Phương thức riêng tư thứ hai, xử lý biến cố Click của mnuQp, có tên mnuQp_Click. Phương thức này có hai tham biến hình thức. Tham biến hình thức thứ nhất có tên sender, với kiểu object. Tham biến hình thức thứ hai có tên e, với kiểu EventArgs. Phương thức này không có kiểu dữ liệu trả về. Nhiệm vụ của phương thức này là:

```
frmTruongQp frm = new frmTruongQp();
frm.MdiParent = this;
frm.Show();
```

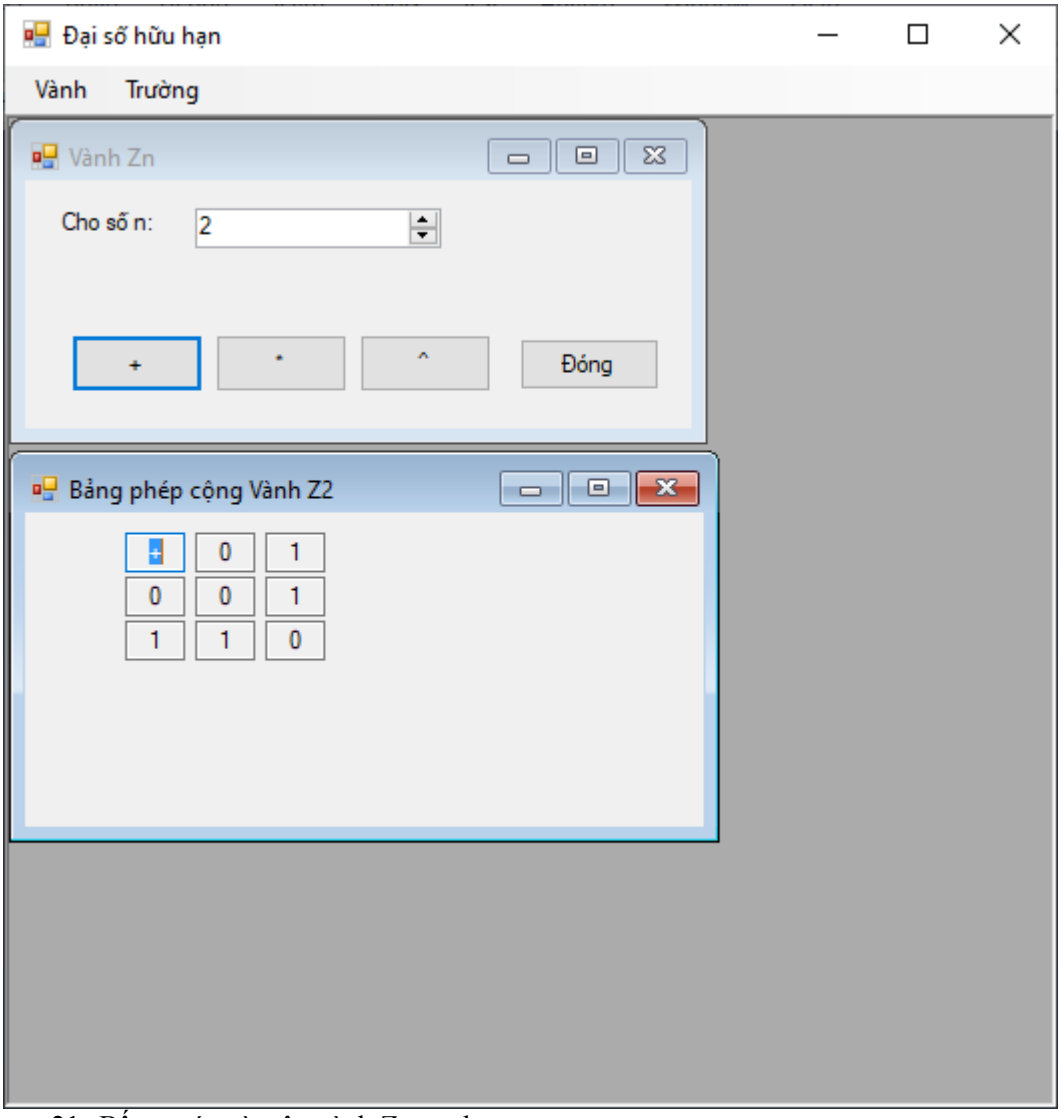
18. Chạy thử chương trình ứng dụng:



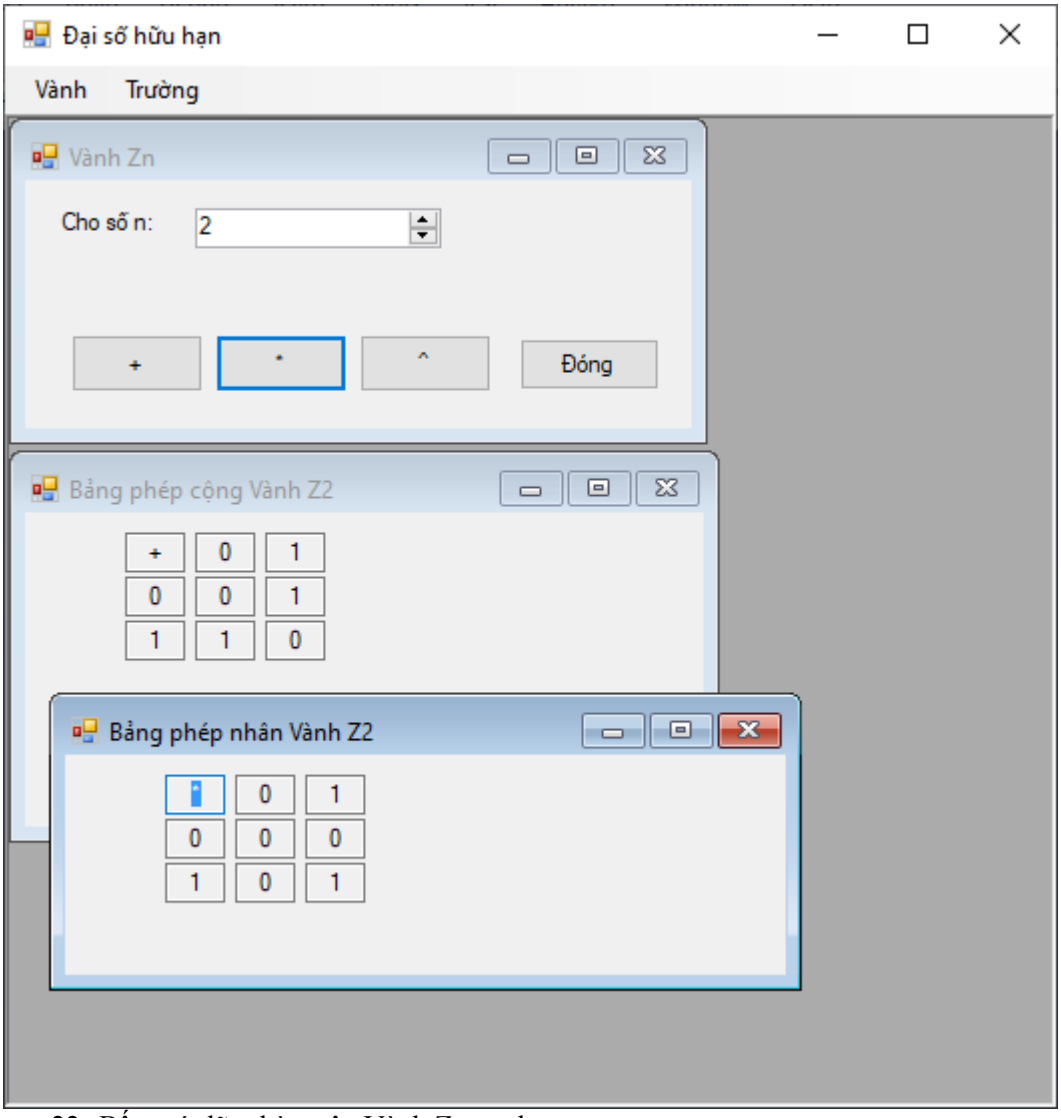
19. Chọn Vành, rồi chọn tiếp Vành Zn.



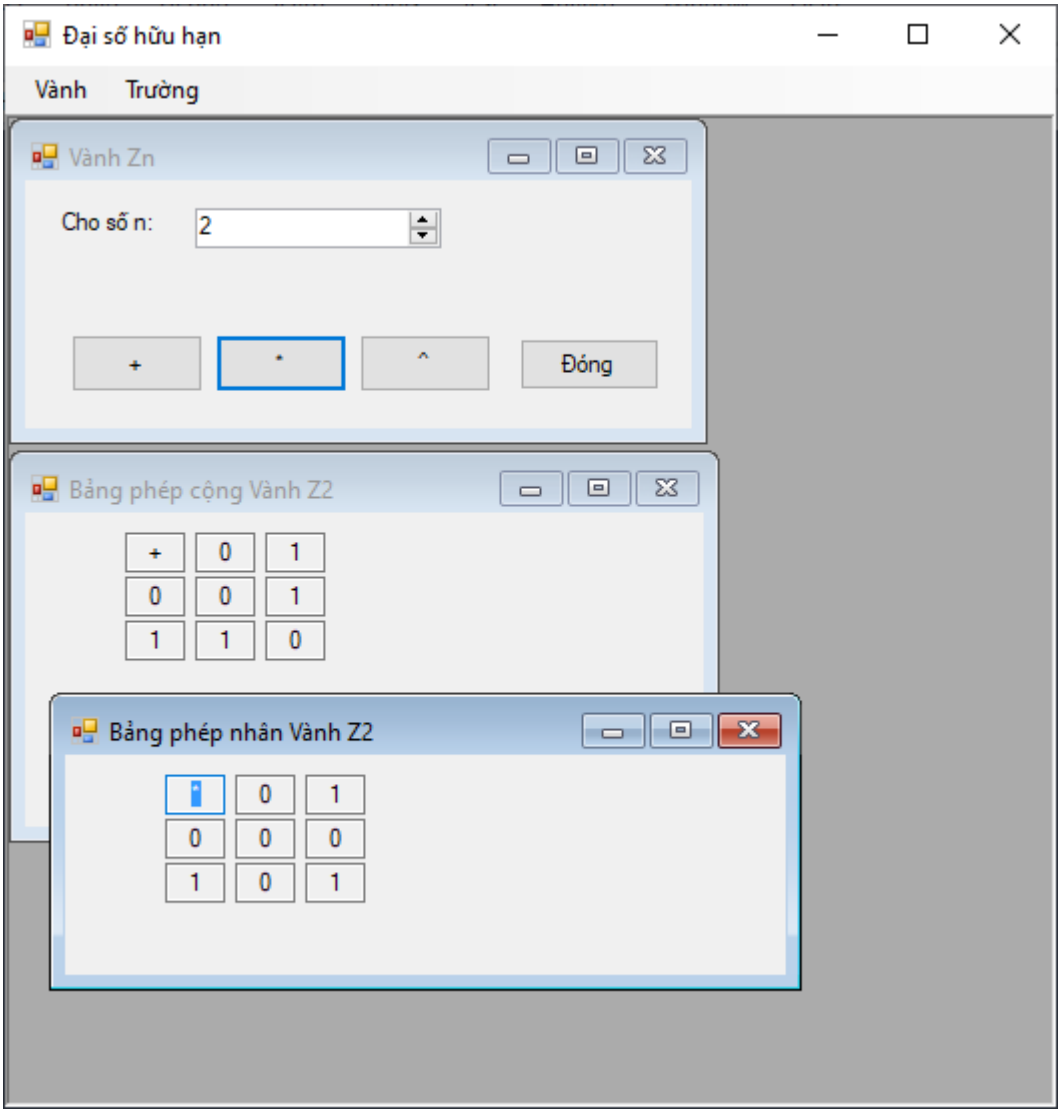
20. Bấm nút +, ta được:



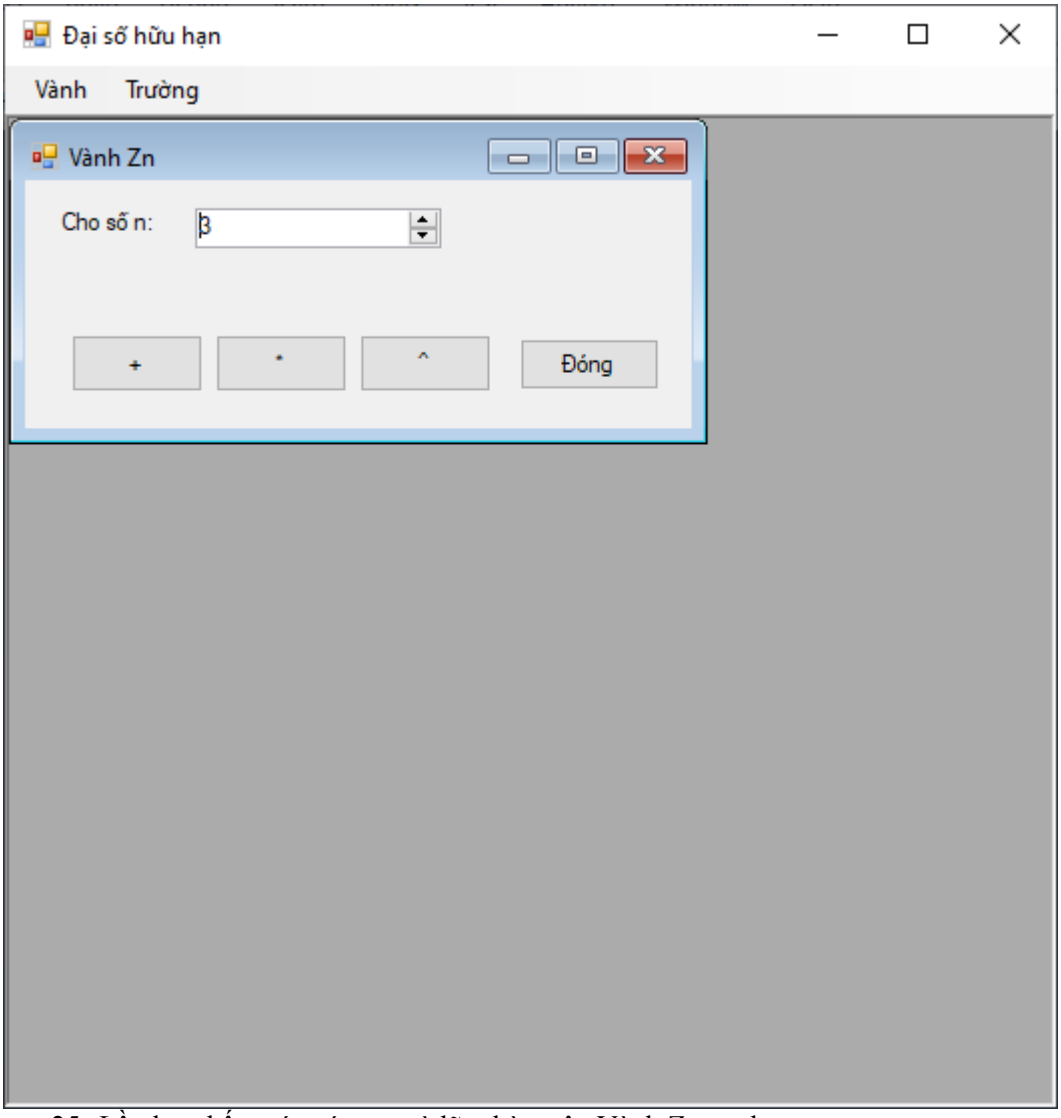
21. Bấnm nút trừ trên vành Zn, ta được:



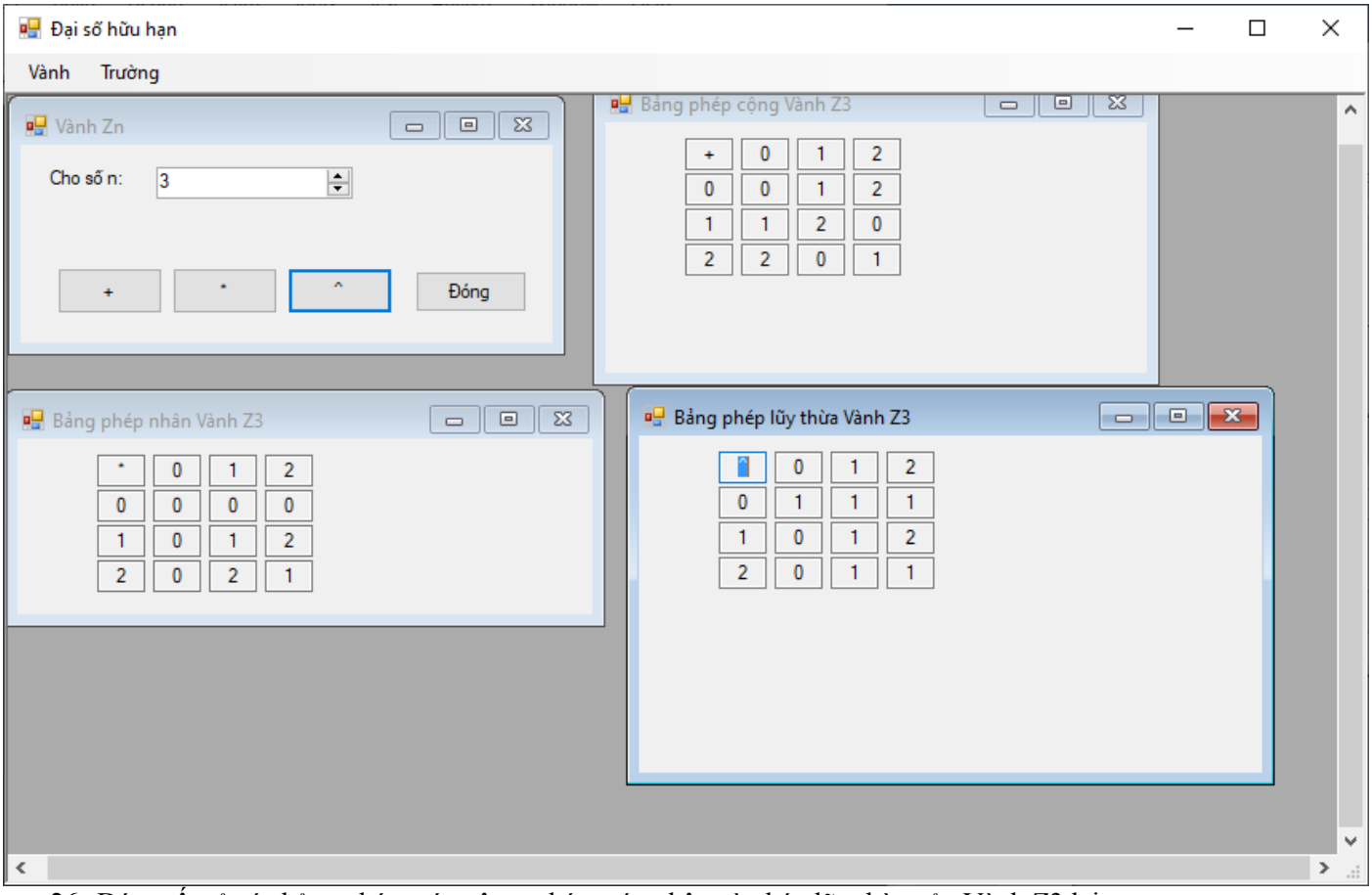
22. Bấm nút lũy thừa trên Vành Zn, ta được:



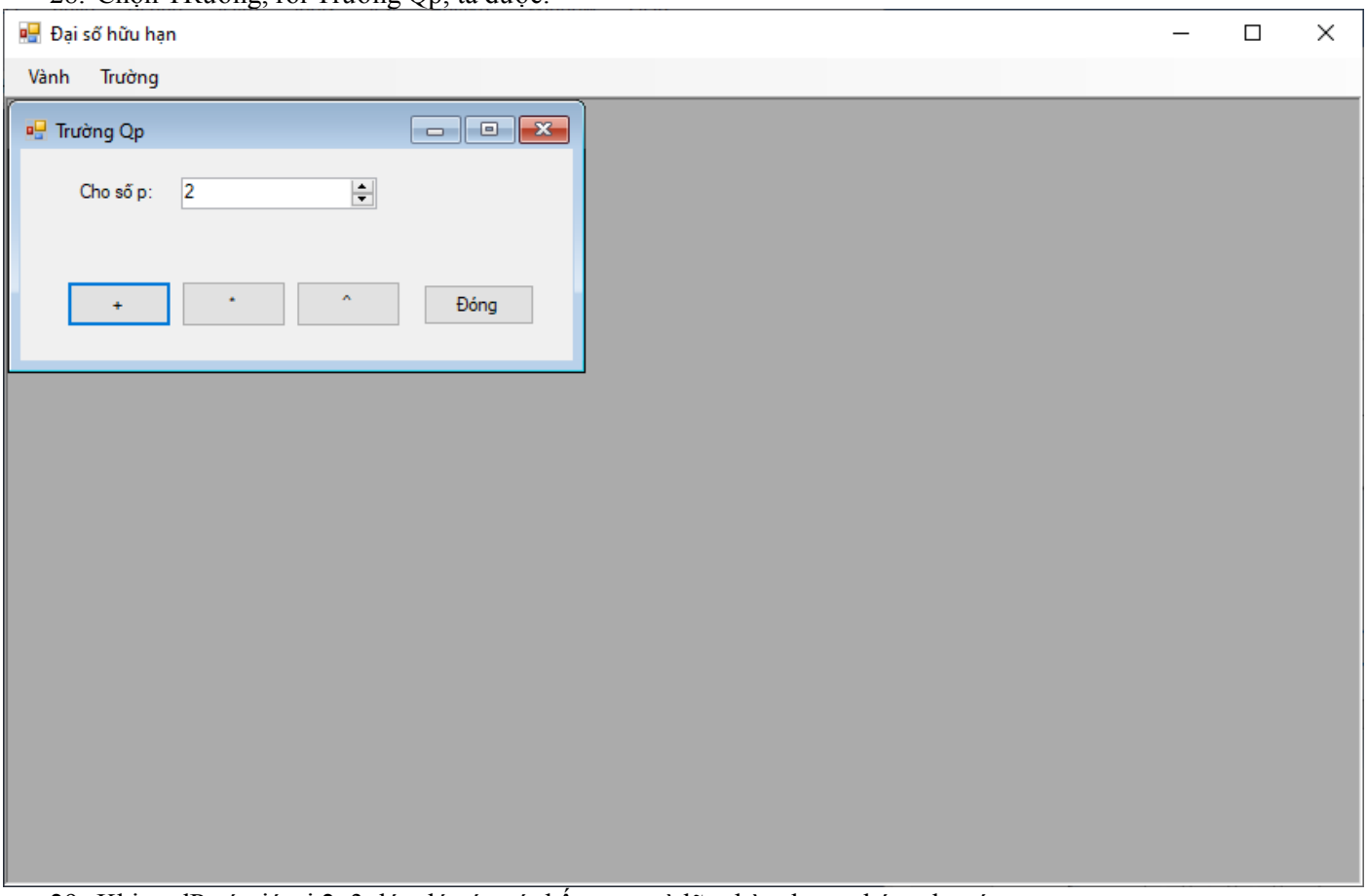
23. Đóng ba bảng phép cộng, phép trừ, và lũy thừa của Z2 lại.
24. Bấm nút lên trong nudN để được con số 3.



25. Lần lượt bấm các nút +, - và lũy thừa trên Vành Zn, ta được:

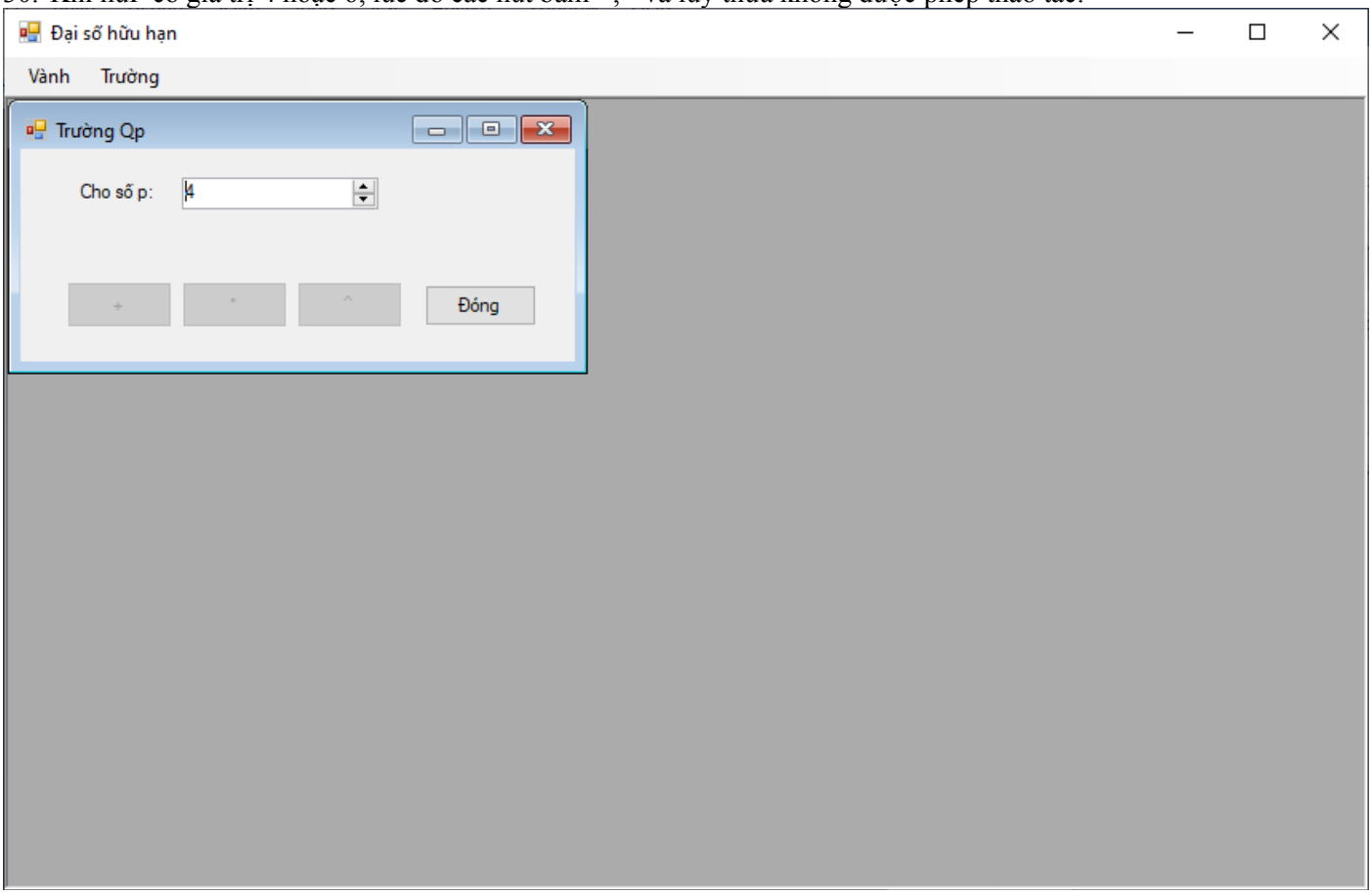


26. Đóng tất cả các bảng phép toán cộng, phép toán nhân và phép lũy thừa của Vành Z3 lại.
27. Đóng luôn frmVanhZn.
28. Chọn TRường, rồi Trường Qp, ta được:

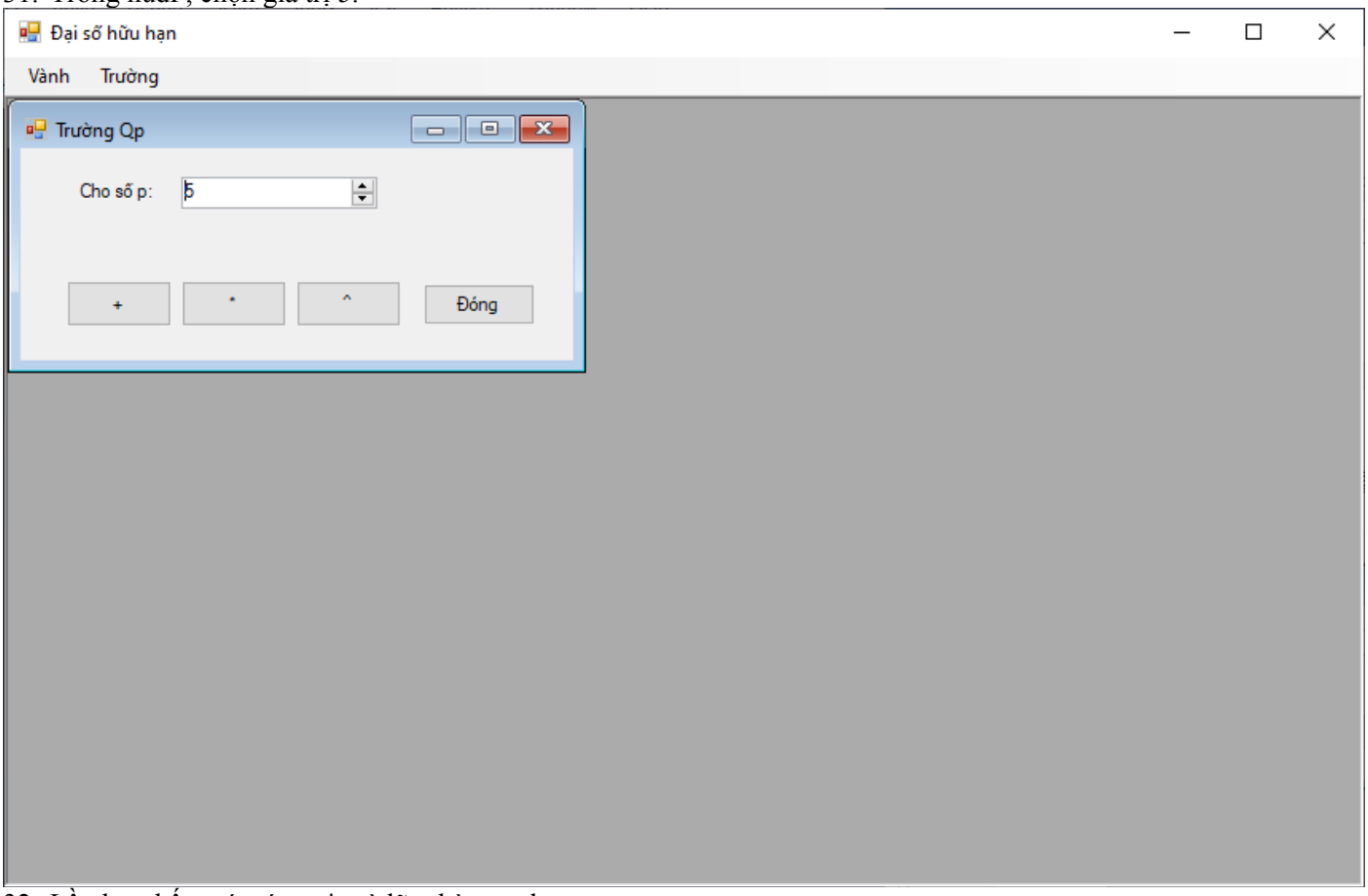


29. Khi nudP có giá trị 2, 3, lúc đó các nút bấm +, - và lũy thừa được phép tahy tác.

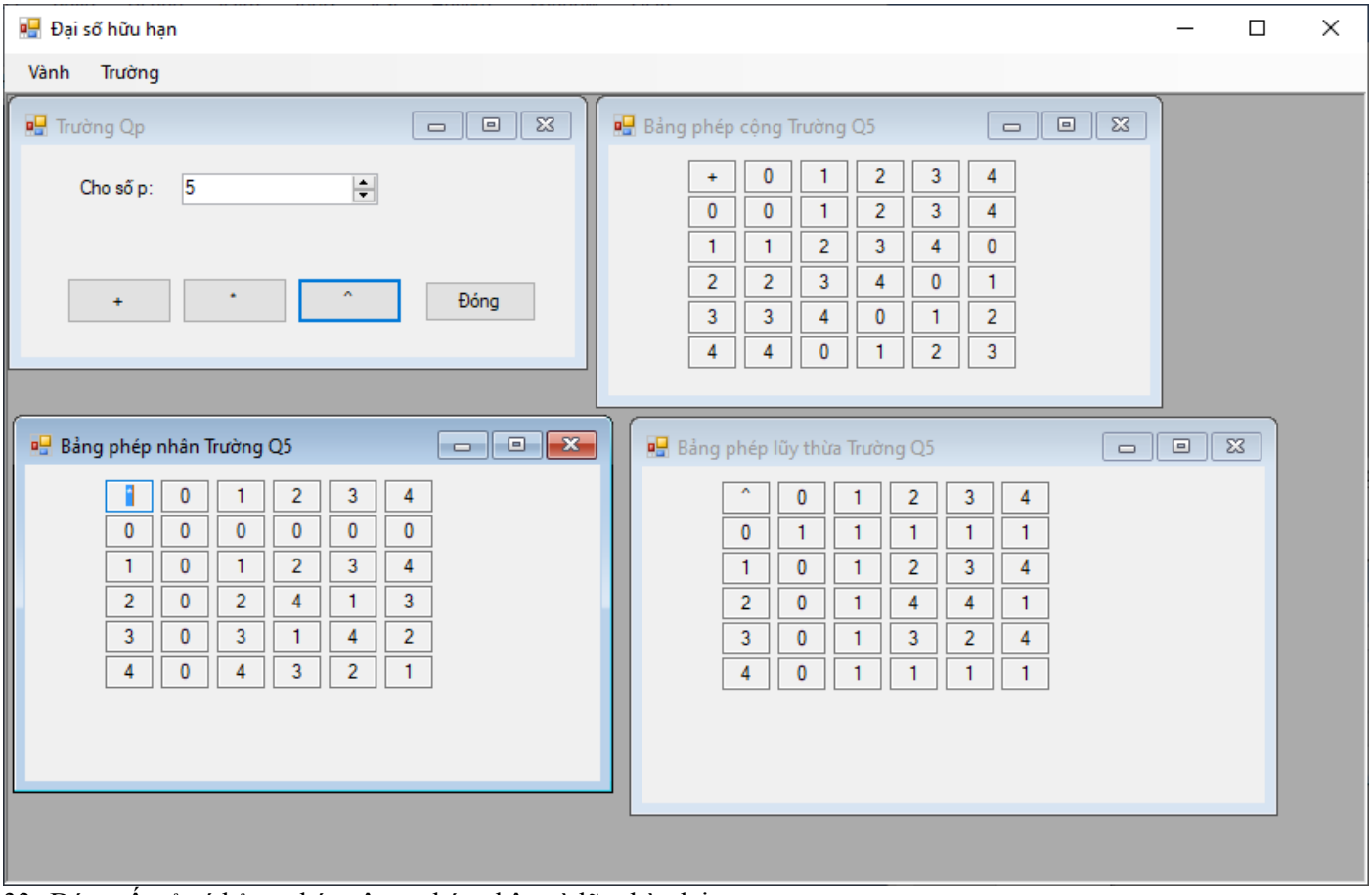
30. Khi nuP có giá trị 4 hoặc 6, lúc đó các nút bấm +, - và lũy thừa không được phép thao tác.



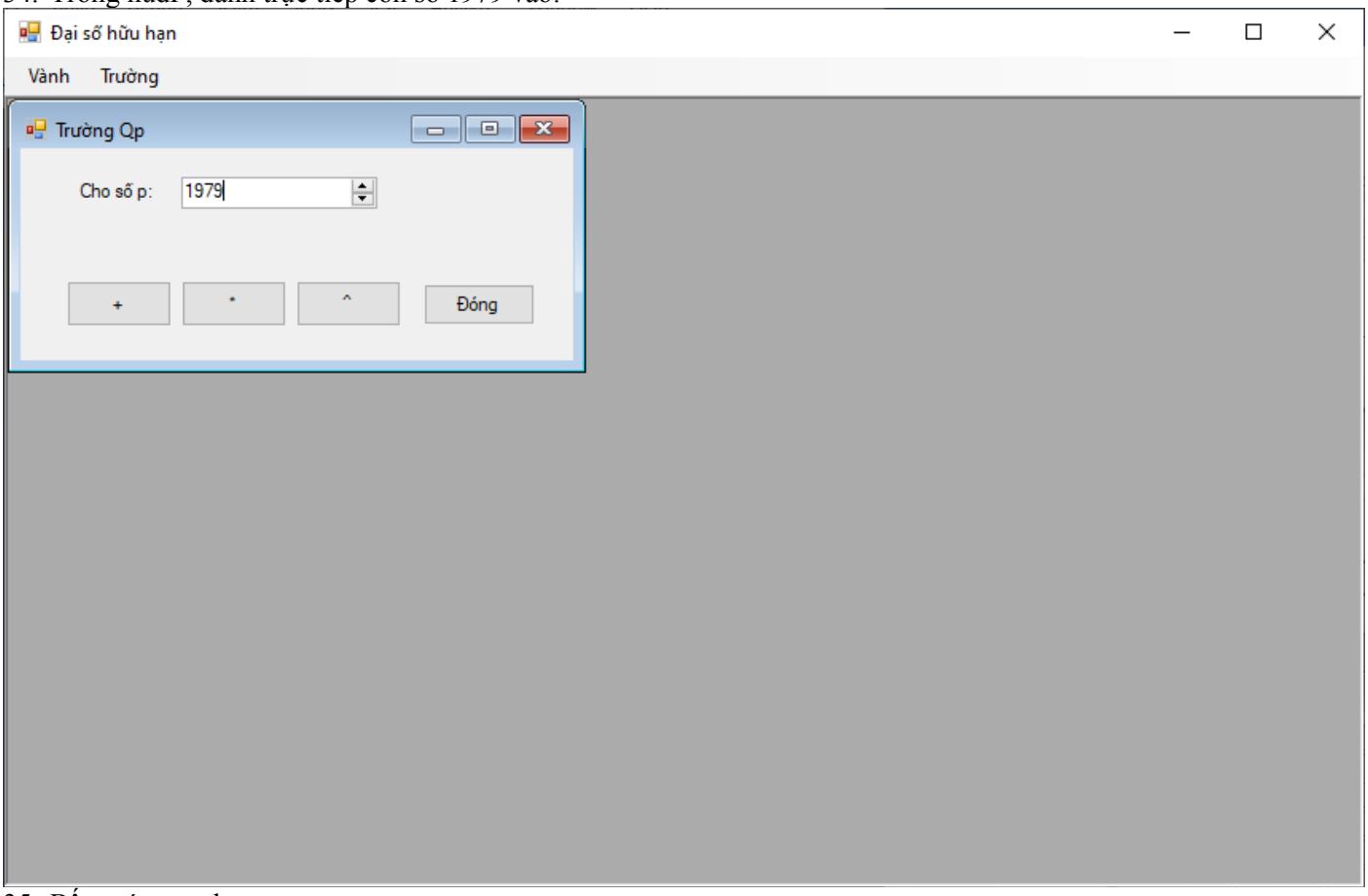
31. Trong nudP, chọn giá trị 5.



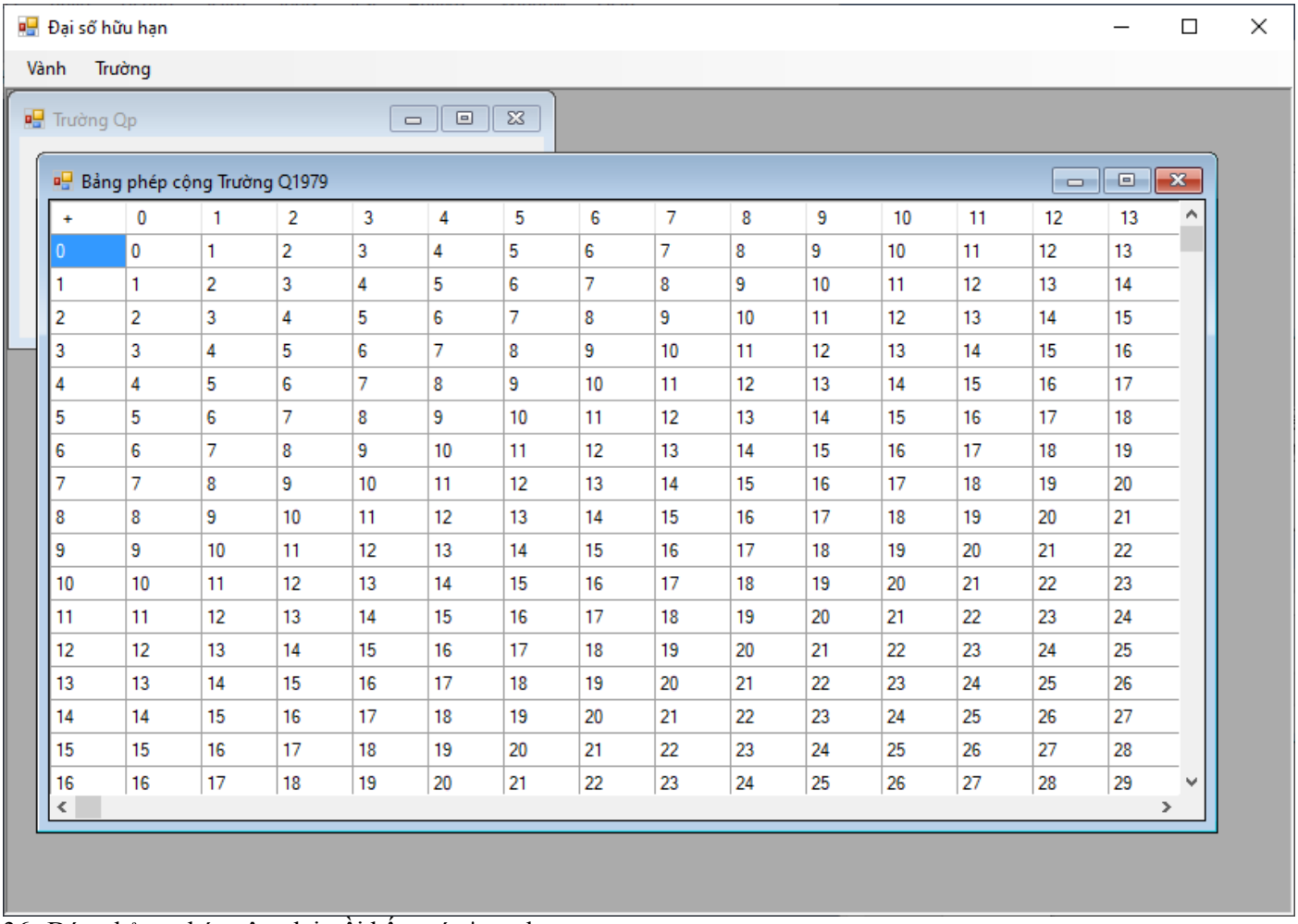
32. Lần lượt bấm cácnút +, *, và lũy thừa, ta được:



33. Đóng tất cả cá bảng phép cộng, phép nhân và lũy thừa lại.
34. Trong nudP, đánh trực tiếp con số 1979 vào:

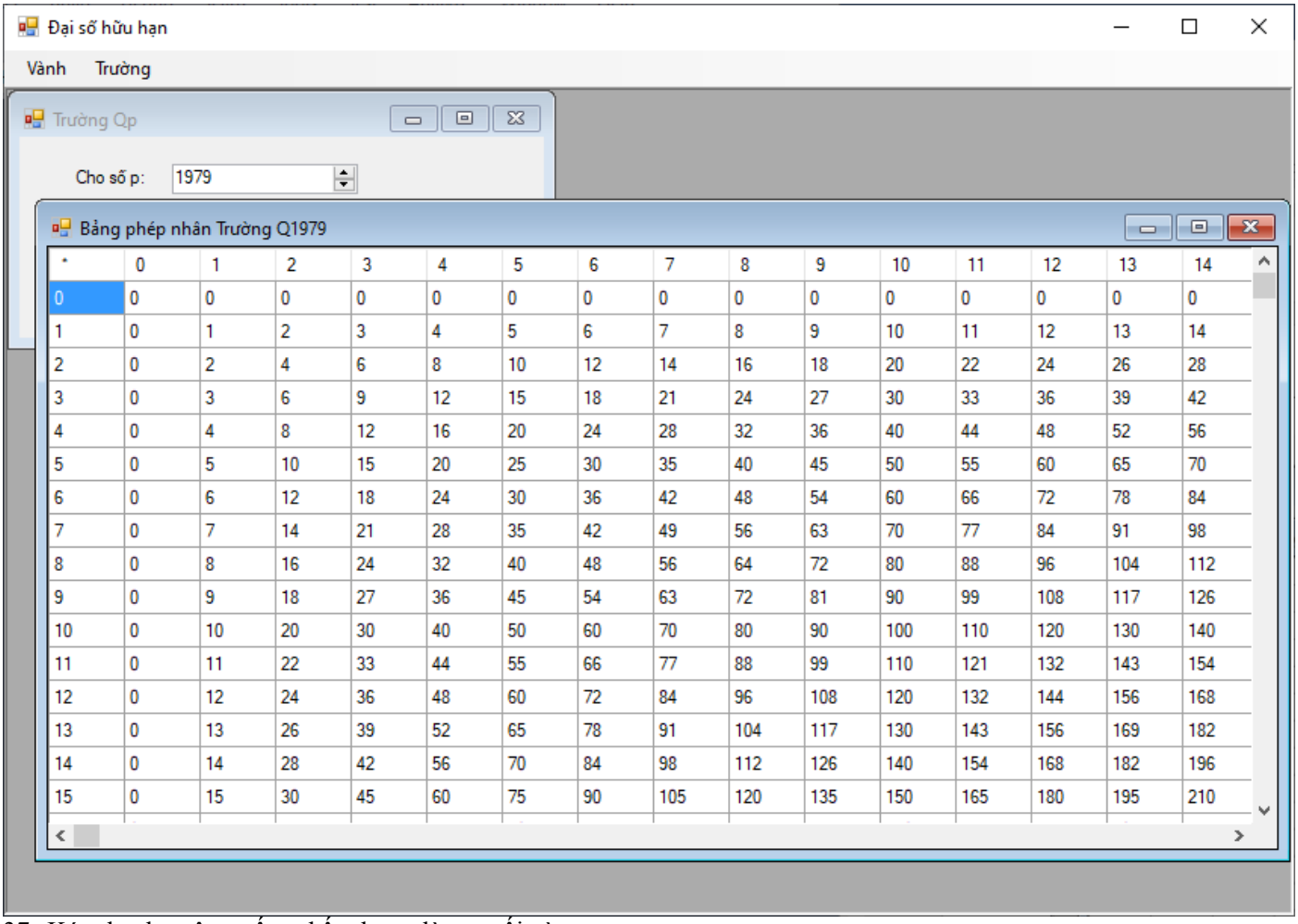


35. Bấm nút + ta được.



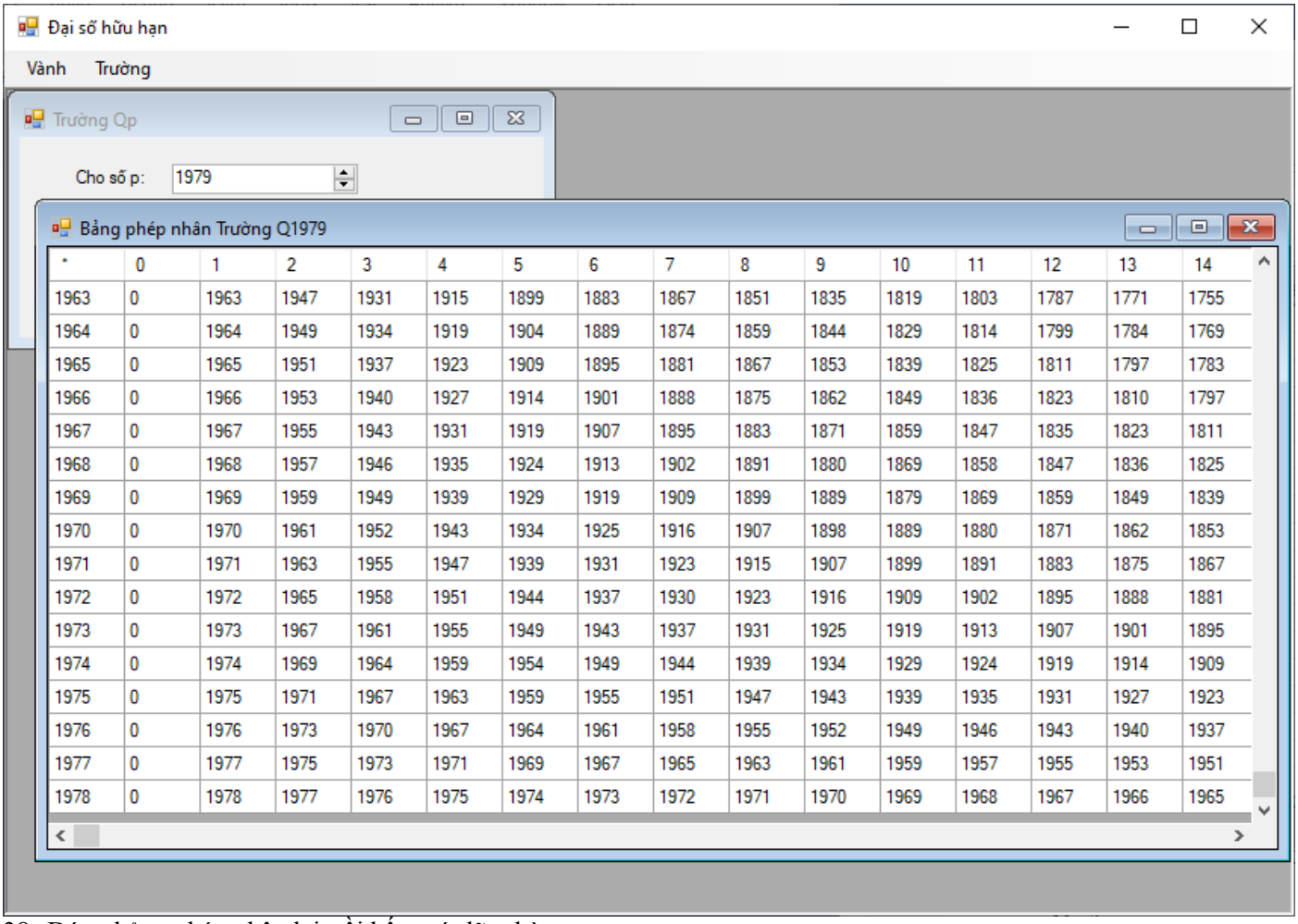
+	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	3	4	5	6	7	8	9	10	11	12	13	14	15	16
4	4	5	6	7	8	9	10	11	12	13	14	15	16	17
5	5	6	7	8	9	10	11	12	13	14	15	16	17	18
6	6	7	8	9	10	11	12	13	14	15	16	17	18	19
7	7	8	9	10	11	12	13	14	15	16	17	18	19	20
8	8	9	10	11	12	13	14	15	16	17	18	19	20	21
9	9	10	11	12	13	14	15	16	17	18	19	20	21	22
10	10	11	12	13	14	15	16	17	18	19	20	21	22	23
11	11	12	13	14	15	16	17	18	19	20	21	22	23	24
12	12	13	14	15	16	17	18	19	20	21	22	23	24	25
13	13	14	15	16	17	18	19	20	21	22	23	24	25	26
14	14	15	16	17	18	19	20	21	22	23	24	25	26	27
15	15	16	17	18	19	20	21	22	23	24	25	26	27	28
16	16	17	18	19	20	21	22	23	24	25	26	27	28	29

36. Đóng bảng phép cộng lại, rồi bấm nút *, ta được:



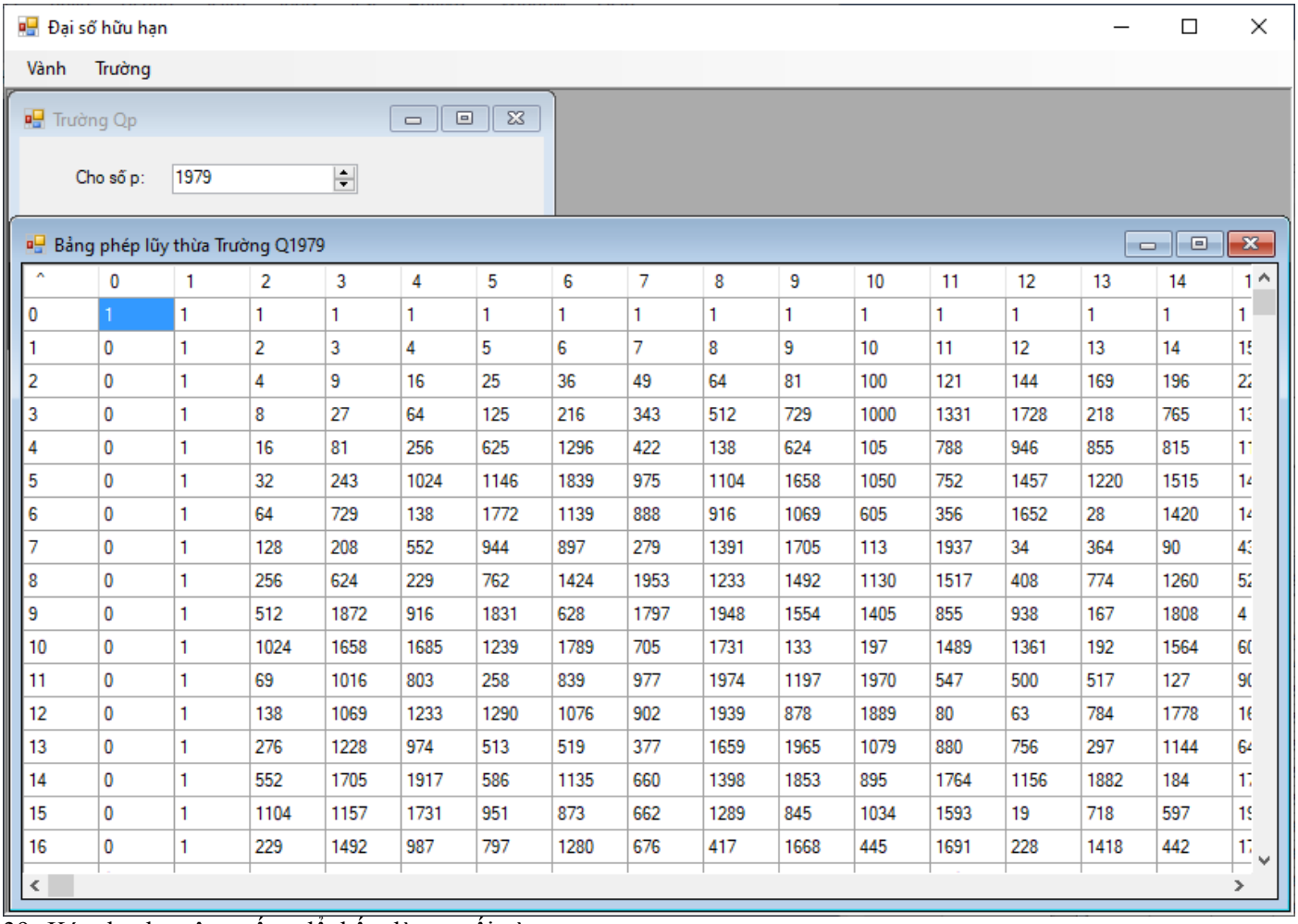
*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28
3	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42
4	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56
5	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70
6	0	6	12	18	24	30	36	42	48	54	60	66	72	78	84
7	0	7	14	21	28	35	42	49	56	63	70	77	84	91	98
8	0	8	16	24	32	40	48	56	64	72	80	88	96	104	112
9	0	9	18	27	36	45	54	63	72	81	90	99	108	117	126
10	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140
11	0	11	22	33	44	55	66	77	88	99	110	121	132	143	154
12	0	12	24	36	48	60	72	84	96	108	120	132	144	156	168
13	0	13	26	39	52	65	78	91	104	117	130	143	156	169	182
14	0	14	28	42	56	70	84	98	112	126	140	154	168	182	196
15	0	15	30	45	60	75	90	105	120	135	150	165	180	195	210

37. Kéo thanh cuộn xuống thấy được dòng cuối cùng



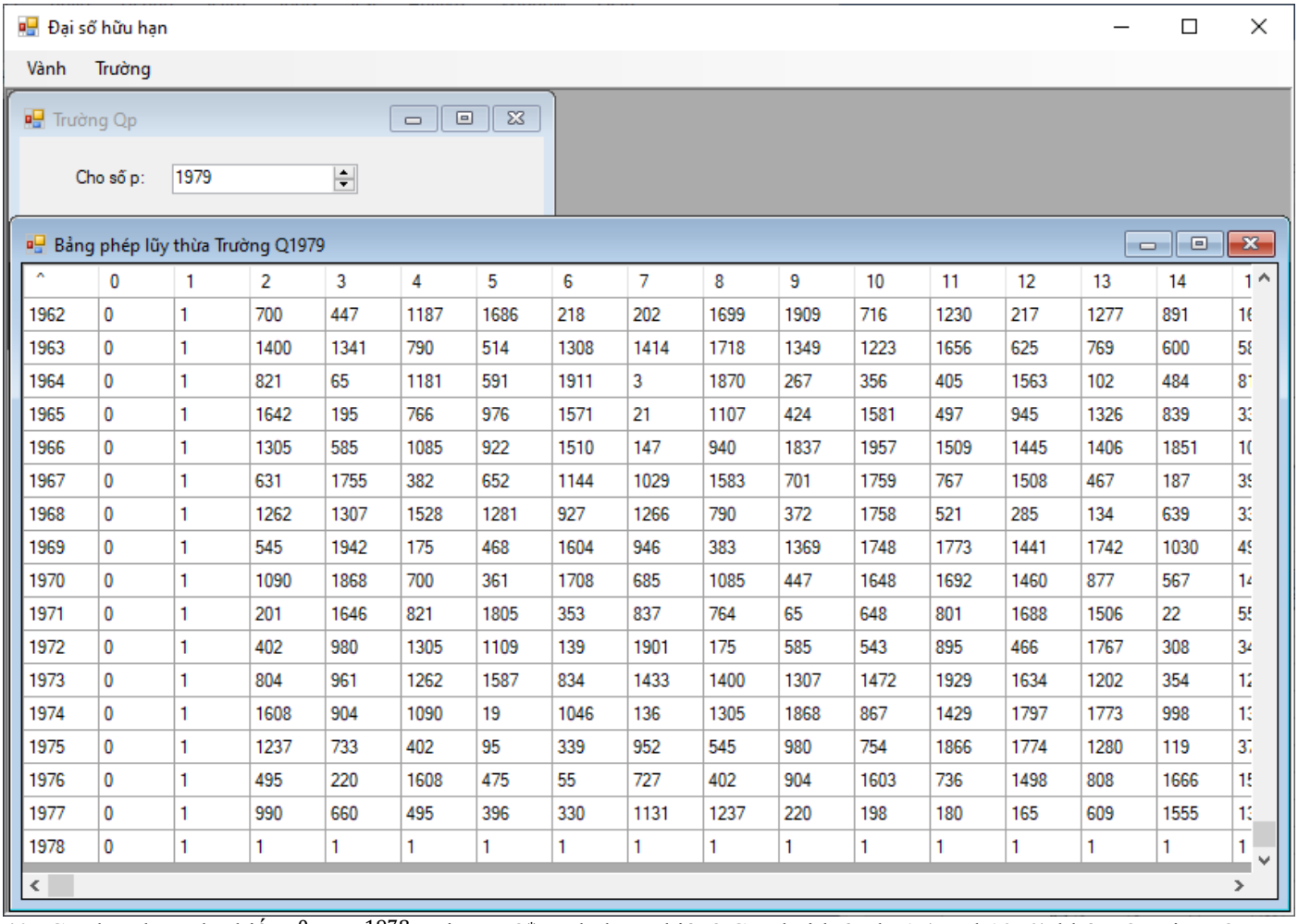
*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1963	0	1963	1947	1931	1915	1899	1883	1867	1851	1835	1819	1803	1787	1771	1755
1964	0	1964	1949	1934	1919	1904	1889	1874	1859	1844	1829	1814	1799	1784	1769
1965	0	1965	1951	1937	1923	1909	1895	1881	1867	1853	1839	1825	1811	1797	1783
1966	0	1966	1953	1940	1927	1914	1901	1888	1875	1862	1849	1836	1823	1810	1797
1967	0	1967	1955	1943	1931	1919	1907	1895	1883	1871	1859	1847	1835	1823	1811
1968	0	1968	1957	1946	1935	1924	1913	1902	1891	1880	1869	1858	1847	1836	1825
1969	0	1969	1959	1949	1939	1929	1919	1909	1899	1889	1879	1869	1859	1849	1839
1970	0	1970	1961	1952	1943	1934	1925	1916	1907	1898	1889	1880	1871	1862	1853
1971	0	1971	1963	1955	1947	1939	1931	1923	1915	1907	1899	1891	1883	1875	1867
1972	0	1972	1965	1958	1951	1944	1937	1930	1923	1916	1909	1902	1895	1888	1881
1973	0	1973	1967	1961	1955	1949	1943	1937	1931	1925	1919	1913	1907	1901	1895
1974	0	1974	1969	1964	1959	1954	1949	1944	1939	1934	1929	1924	1919	1914	1909
1975	0	1975	1971	1967	1963	1959	1955	1951	1947	1943	1939	1935	1931	1927	1923
1976	0	1976	1973	1970	1967	1964	1961	1958	1955	1952	1949	1946	1943	1940	1937
1977	0	1977	1975	1973	1971	1969	1967	1965	1963	1961	1959	1957	1955	1953	1951
1978	0	1978	1977	1976	1975	1974	1973	1972	1971	1970	1969	1968	1967	1966	1965

38. Đóng bảng phép nhân lại, rồi bấm nút lũy thừa.



^	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	4	9	16	25	36	49	64	81	100	121	144	169	196	22
3	0	1	8	27	64	125	216	343	512	729	1000	1331	1728	218	765	13
4	0	1	16	81	256	625	1296	422	138	624	105	788	946	855	815	11
5	0	1	32	243	1024	1146	1839	975	1104	1658	1050	752	1457	1220	1515	14
6	0	1	64	729	138	1772	1139	888	916	1069	605	356	1652	28	1420	14
7	0	1	128	208	552	944	897	279	1391	1705	113	1937	34	364	90	43
8	0	1	256	624	229	762	1424	1953	1233	1492	1130	1517	408	774	1260	52
9	0	1	512	1872	916	1831	628	1797	1948	1554	1405	855	938	167	1808	4
10	0	1	1024	1658	1685	1239	1789	705	1731	133	197	1489	1361	192	1564	60
11	0	1	69	1016	803	258	839	977	1974	1197	1970	547	500	517	127	90
12	0	1	138	1069	1233	1290	1076	902	1939	878	1889	80	63	784	1778	16
13	0	1	276	1228	974	513	519	377	1659	1965	1079	880	756	297	1144	64
14	0	1	552	1705	1917	586	1135	660	1398	1853	895	1764	1156	1882	184	17
15	0	1	1104	1157	1731	951	873	662	1289	845	1034	1593	19	718	597	19
16	0	1	229	1492	987	797	1280	676	417	1668	445	1691	228	1418	442	17

39. Kéo thanh cuộn xuống để thấy dòng cuối cùng.



^	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1
1962	0	1	700	447	1187	1686	218	202	1699	1909	716	1230	217	1277	891	1
1963	0	1	1400	1341	790	514	1308	1414	1718	1349	1223	1656	625	769	600	5
1964	0	1	821	65	1181	591	1911	3	1870	267	356	405	1563	102	484	8
1965	0	1	1642	195	766	976	1571	21	1107	424	1581	497	945	1326	839	3
1966	0	1	1305	585	1085	922	1510	147	940	1837	1957	1509	1445	1406	1851	1
1967	0	1	631	1755	382	652	1144	1029	1583	701	1759	767	1508	467	187	3
1968	0	1	1262	1307	1528	1281	927	1266	790	372	1758	521	285	134	639	3
1969	0	1	545	1942	175	468	1604	946	383	1369	1748	1773	1441	1742	1030	4
1970	0	1	1090	1868	700	361	1708	685	1085	447	1648	1692	1460	877	567	1
1971	0	1	201	1646	821	1805	353	837	764	65	648	801	1688	1506	22	5
1972	0	1	402	980	1305	1109	139	1901	175	585	543	895	466	1767	308	3
1973	0	1	804	961	1262	1587	834	1433	1400	1307	1472	1929	1634	1202	354	1
1974	0	1	1608	904	1090	19	1046	136	1305	1868	867	1429	1797	1773	998	1
1975	0	1	1237	733	402	95	339	952	545	980	754	1866	1774	1280	119	3
1976	0	1	495	220	1608	475	55	727	402	904	1603	736	1498	808	1666	1
1977	0	1	990	660	495	396	330	1131	1237	220	198	180	165	609	1555	1
1978	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

40. Các bạn hãy cho biết a^0 và a^{1978} với $a \in Q^*_{1979}$ là bao nhiêu? Có phải luôn là 1 (mod 1979) không? Tại sao? Có phải vì 1979 là số nguyên tố và 1978 = 1979 – 1 không?
41. Theo các bạn số 1997 có là số nguyên tố không? Như vậy dòng a^{1996} với $a \in Q^*_{1997}$ là bao nhiêu?
42. Các bạn thử thêm với các số nguyên tố 1993, 1999, 2003, 2011, 2017, 2027
43. Chúc các bạn thành công.

---oOo---