

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net.NetworkInformation;
using System.IO;

namespace CheckPing
{
    public partial class ICMP : Form
    {
        Ping p;
        int br=0;
        public ICMP()
        {
            InitializeComponent();
            txtInput.Focus();
        }
        void constructListview()
        {
            lvResult.Clear();
            lvResult.Columns.Add("Lần gửi", 36);
            lvResult.Columns.Add("Địa chỉ", 146);
            lvResult.Columns.Add("Kết quả", 212);
        }

        void check()
        {
            int success = 0;
            int count = 1;
            int numPacket = 4;
            List<int> roundtrip= new List<int>();
            int timeout = 4000;
            int size = 32;

            if (txtNumpacket.Text != String.Empty)
            {
                try
                {
                    numPacket = Convert.ToInt32(txtNumpacket.Text);
                }
                catch
                {
                    MessageBox.Show("Số gói tin là một số nguyên cụ thể!", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    return;
                }
            }

            if (txtSize.Text != String.Empty)
            {
                try
            
```

```

        {
            size = Convert.ToInt32(txtSize.Text);
        }
        catch
        {
            MessageBox.Show("Kích thước gói tin là một số nguyên cụ thể!", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        if (size < 0 || size > 65500)
        {
            MessageBox.Show("Kích thước gói tin không được nhỏ hơn 0 và lớn hơn 65500", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
    }

    byte[] buffer = new byte[size];

    string ip = txtInput.Text;

    if (ip == string.Empty)
    {
        MessageBox.Show("Nhập hostname hoặc địa chỉ ip để bắt đầu", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    while (count <= numPacket )
    {
        if (chbxContinous.Checked == true)
        {
            numPacket++;
        }

        Application.DoEvents();

        if (br == 1) break;
        p = new Ping();
        PingReply pr;
        try
        {
            pr = p.Send(ip, timeout, buffer);

            if (pr.Status == IPStatus.DestinationHostUnreachable)
            {
                string thongbao = "Destination Host Unreachable!";
                addtoLv(ip, thongbao, Color.Purple, count);
                success++;
            }

            if (pr.Status == IPStatus.TimedOut)
            {
                string thongbao = "Oops! Request Timeout!";
            }
        }
    }

```

```

        addtoLv(ip, thongbao, Color.Red, count);
    }

    if (pr.Status == IPStatus.Success)
    {
        string thongbao= "Ping success!";
        addtoLv(ip, thongbao, Color.Green, count);
        success++;
        if(pr.RoundtripTime != 0)
            roundtrip.Add(Convert.ToInt32(pr.RoundtripTime));
    }

    if (pr.Status == IPStatus.Unknown)
    {
        string thongbao = "Chưa biết nguyên nhân !";
        addtoLv(ip, thongbao, Color.Green, count);
    }

    lvDetail.Items.Add("Lần phản hồi: " + count);
    if (pr.Address != null)
    {
        lvDetail.Items.Add("Reply from: " + pr.Address.ToString());
    }
    if (pr.RoundtripTime != 0)
    {
        lvDetail.Items.Add("Thời gian hồi đáp: " +
pr.RoundtripTime.ToString() + " ms");
    }
    if (pr.Options != null)
    {
        lvDetail.Items.Add("Thời gian sống của gói tin: " +
pr.Options.Ttl.ToString());
    }
    if(pr.Status== IPStatus.Success)
    {
        lvDetail.Items.Add("Kích thước gói tin: " +
pr.Buffer.Length.ToString() + " byte");
    }
    if (pr.Status == IPStatus.Success || pr.Status ==
IPStatus.DestinationHostUnreachable)
        lvDetail.Items.Add("_____");

    }
    catch
    {
        MessageBox.Show("IP hoặc Host không tồn tại ! Mời nhập lại!", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    count++;
}

```

```

        count = count - 1;
        int lost = count - success;
        double phantramlost = Math.Round((((double)lost / (double)numPacket) * 100),
0);
        lvThongke.Items.Add("Gửi = " + (count).ToString() + ", Nhận = " +
success.ToString() + ", Mất = " + lost.ToString() + " ( " + phantramlost.ToString() + "%
Mất )");
        if (roundtrip.Count != 0)
        {
            lvThongke.Items.Add("Approximate round trip times in mili-second: ");
            int max = roundtrip.Max();
            int min = roundtrip.Min();
            int avegare = (min + max) / 2;
            lvThongke.Items.Add("Minimun = " + min.ToString() + "ms, Maximum = " +
max.ToString() + "ms, Average = " + avegare.ToString() + "ms");
        }

        br = 0;
    }

    void addtoLv(string ip, string thongbao, Color color, int count)
    {
        ListViewItem address = new ListViewItem(count.ToString());
        ListViewItem.ListViewSubItem result = new ListViewItem.ListViewSubItem(address,
ip);
        ListViewItem.ListViewSubItem result1 = new
ListViewItem.ListViewSubItem(address, thongbao);
        address.SubItems.Add(result);
        address.SubItems.Add(result1);
        lvResult.Items.Add(address);
        address.ForeColor = color;
    }

    private void btCheck_Click(object sender, EventArgs e)
    {
        btCheck.Enabled = false;
        lvDetail.Clear();
        constructListview();
        lvThongke.Clear();
        check();
        btCheck.Enabled = true;
        chbxContinous.Checked = false;
        txtSize.Clear();
        txtNumpacket.Clear();
        txtInput.Focus();
    }

    private void btReset_Click(object sender, EventArgs e)
    {
        lvDetail.Clear();
        constructListview();//thiết lập lvRessult
        lvThongke.Clear();// xóa trống lvThongke
        chbxContinous.Checked = false;
        txtSize.Clear();
        txtNumpacket.Clear();
        txtInput.Clear();
        txtInput.Focus();
    }

    //Xử lí đóng form

```

```

private void ICMP_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show("Thoát chương trình?", "Thông báo!",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
    {
        e.Cancel = true;
    }
}

private void ICMP_KeyPress(object sender, KeyPressEventArgs e)
{
    if(e.KeyChar == 27)// nhấn esc
        br = 1;
}
}
}

```

Ping IP or HostName

PING IP OR HOSTNAME

Kết quả:

Lần gửi	Địa chỉ	Kết quả

Chi tiết:

Nhập IP hoặc Hostname:

Tùy chọn

Kích thước tin (/l) bytes Số gói tin(/n)

Gửi liên tục(/t) ☐ (Nhấn ESC để dừng !)