Sinh viên:Bùi Ngọc Huy

STT: 34

MSSV:1150080137

Lớp: 11CNPM2

LAB2

WINDOWS FORMS APPLICATION ĐƠN GIẢN

A. Mục tiêu

− Hướng dẫn sinh viên xây dựng chương trình Windows Form đơn giản.

− Hướng dẫn làm quen môi trường lập trình C# với phần mềm Microsoft Visual Studio 2022.

− Hướng dẫn sinh viên tập làm quen mới soạn thảo, chỉnh sửa mã nguồn, biên dịch, gỡ lỗi (debug), thực thi chương trình.

− Làm quen với các control cơ bản trên Winform như:

+ Label: hiển thị thông tin.

+ TextBox: nhập thông tin dữ liệu.

+ Button: cho phép người dùng click để thực hiện chức năng nào đó.

+ CheckBox: cho phép người dùng chọn một hoặc nhiều tùy chọn.

+ RadioButton: cho phép người dùng chọn một tùy chọn.

+ MessageBox: hiển thị thông tin đến người dùng theo dạng hộp thoại.

+ NumericUpDown: Hộp chọn tăng giảm giá trị số.

+ Combobox: Hộp chọn 1 giá trị trong danh sách giá trị.

+ ListBox: Danh sách các mục chọn, cho phép chọn 1 hoặc nhiều mục.

+ GroupBox: Nhóm các đối tượng về cùng nhóm.

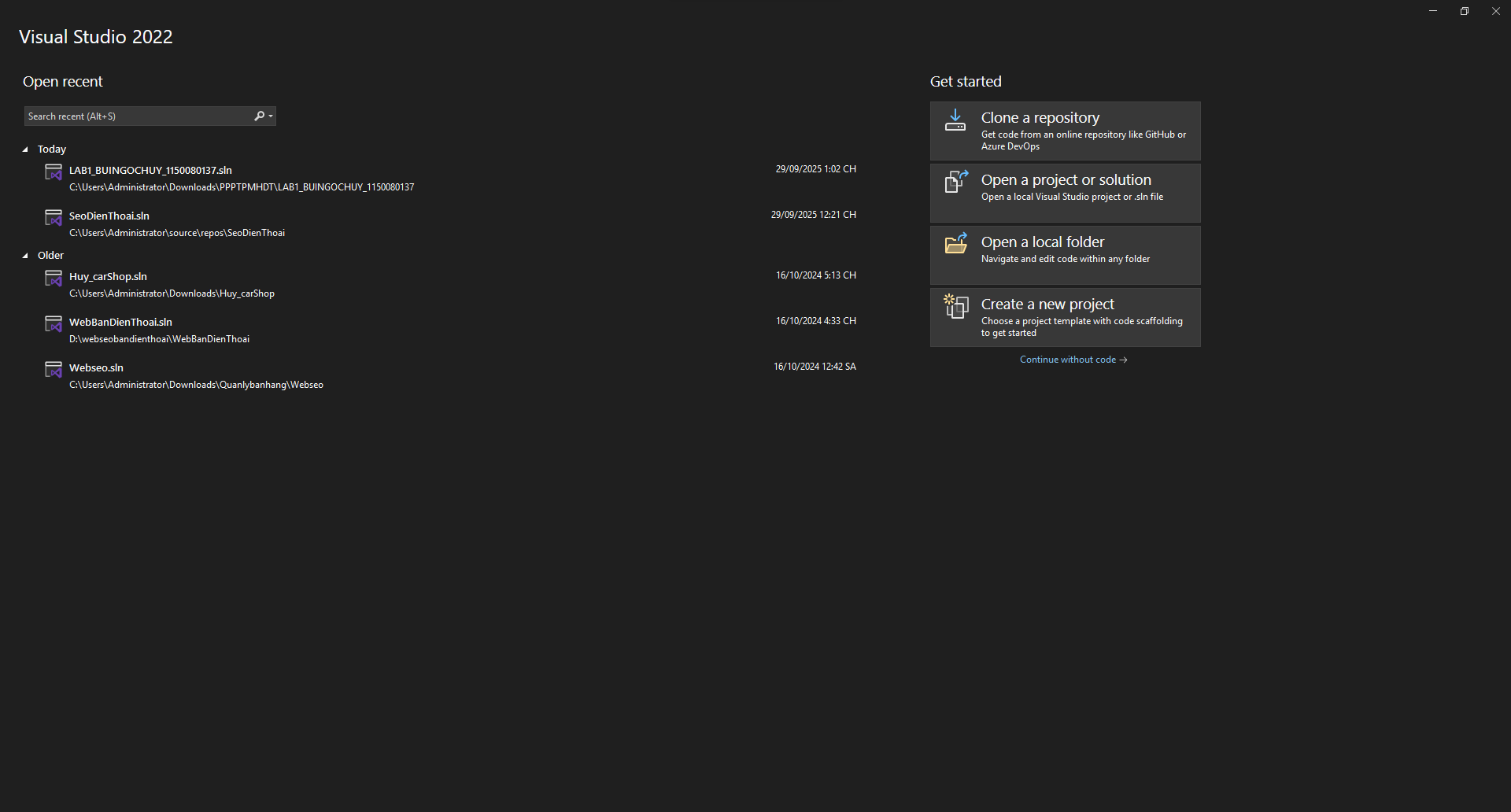
− Làm quen khai báo và xử lý các sự kiện với các control khác nhau.

− Làm quen với cách đặt tên biến lập trình của các control.

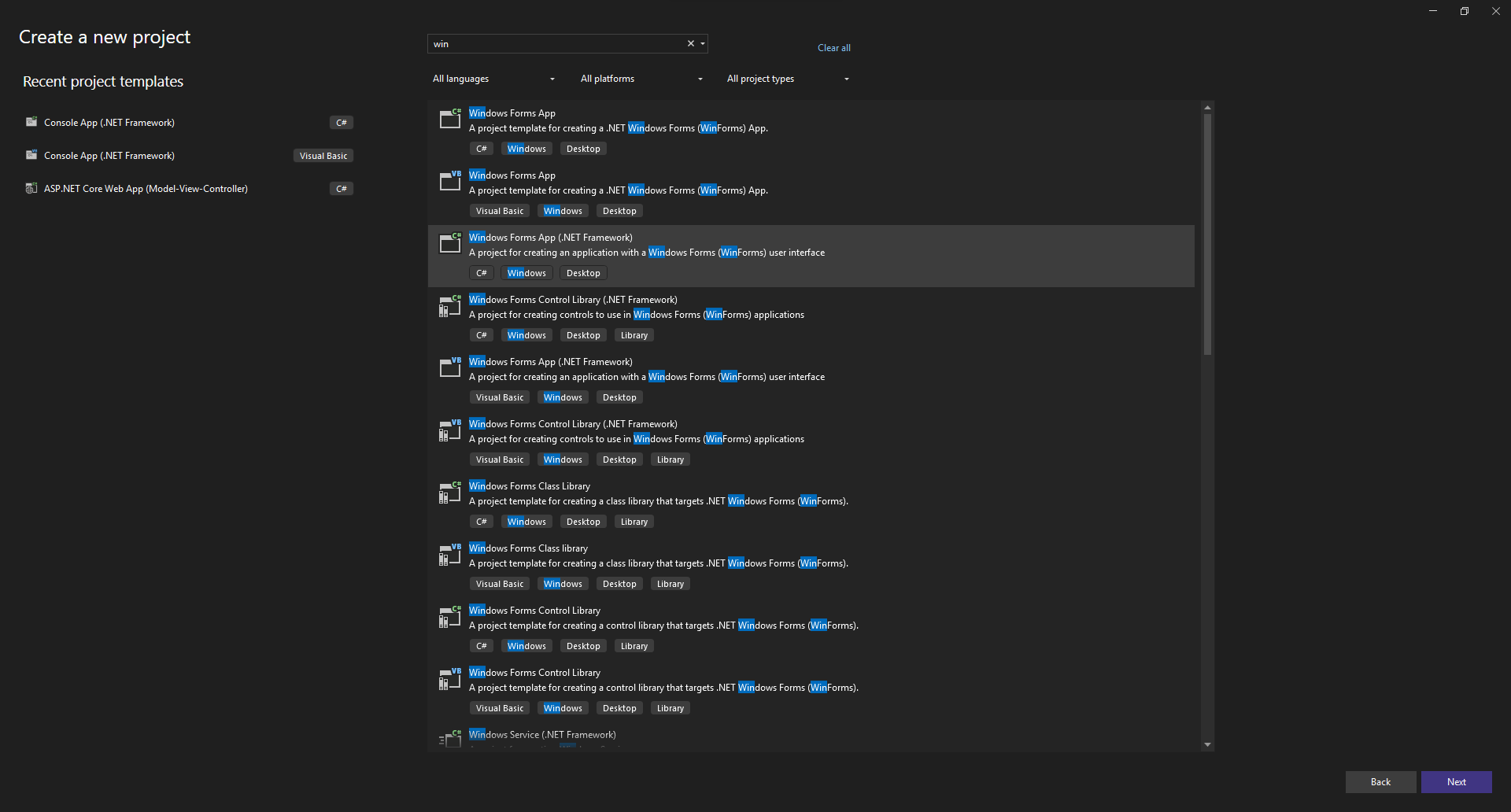
B. Nội dung

1. Tạo Project Windows Forms Application trên Microsoft visualstudio 2022

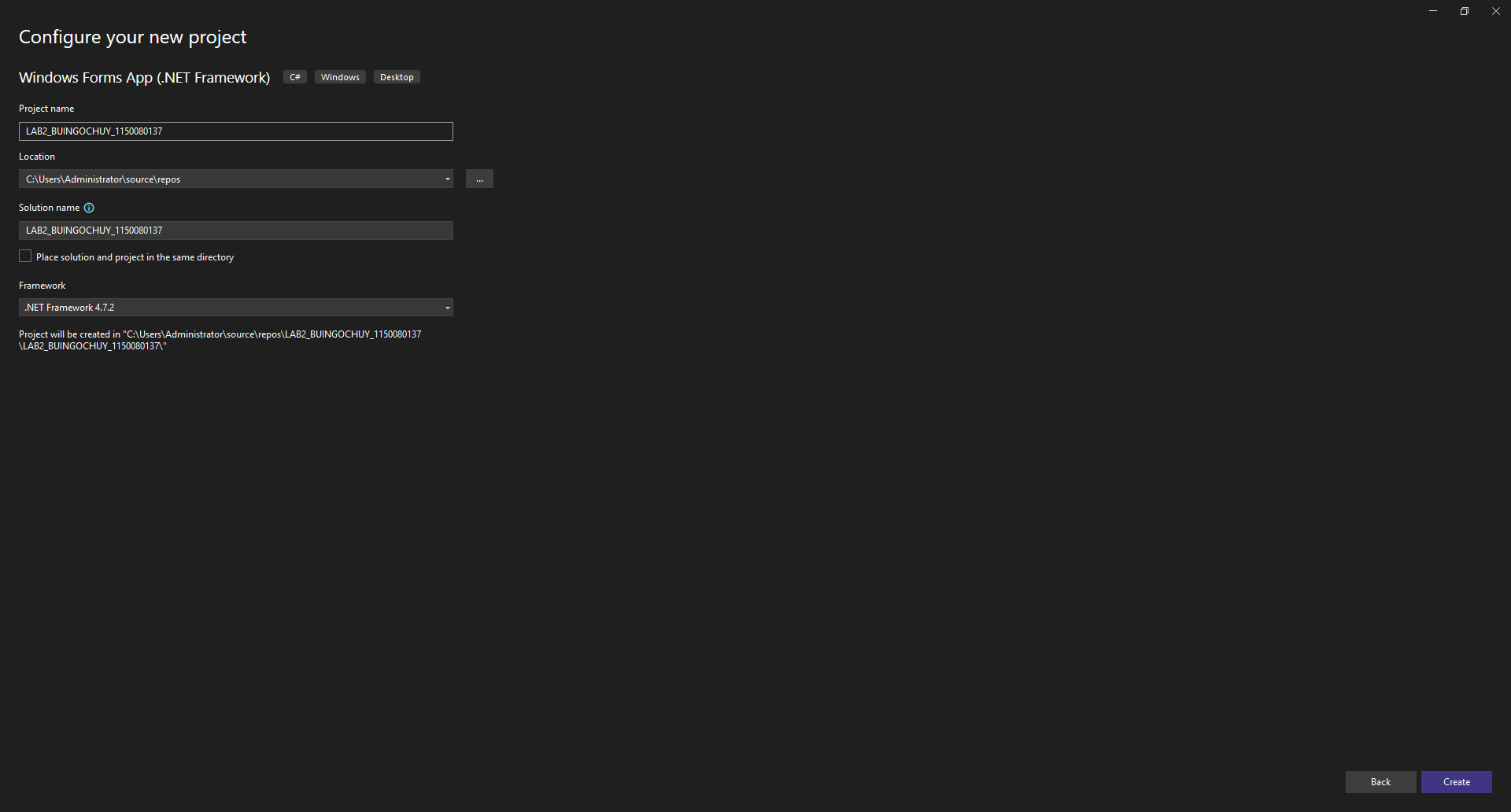
Bước 1: Khởi động chương trình visual studio 2022. Hình ảnh dưới đây là màn hình khởi động của ứng dụng:



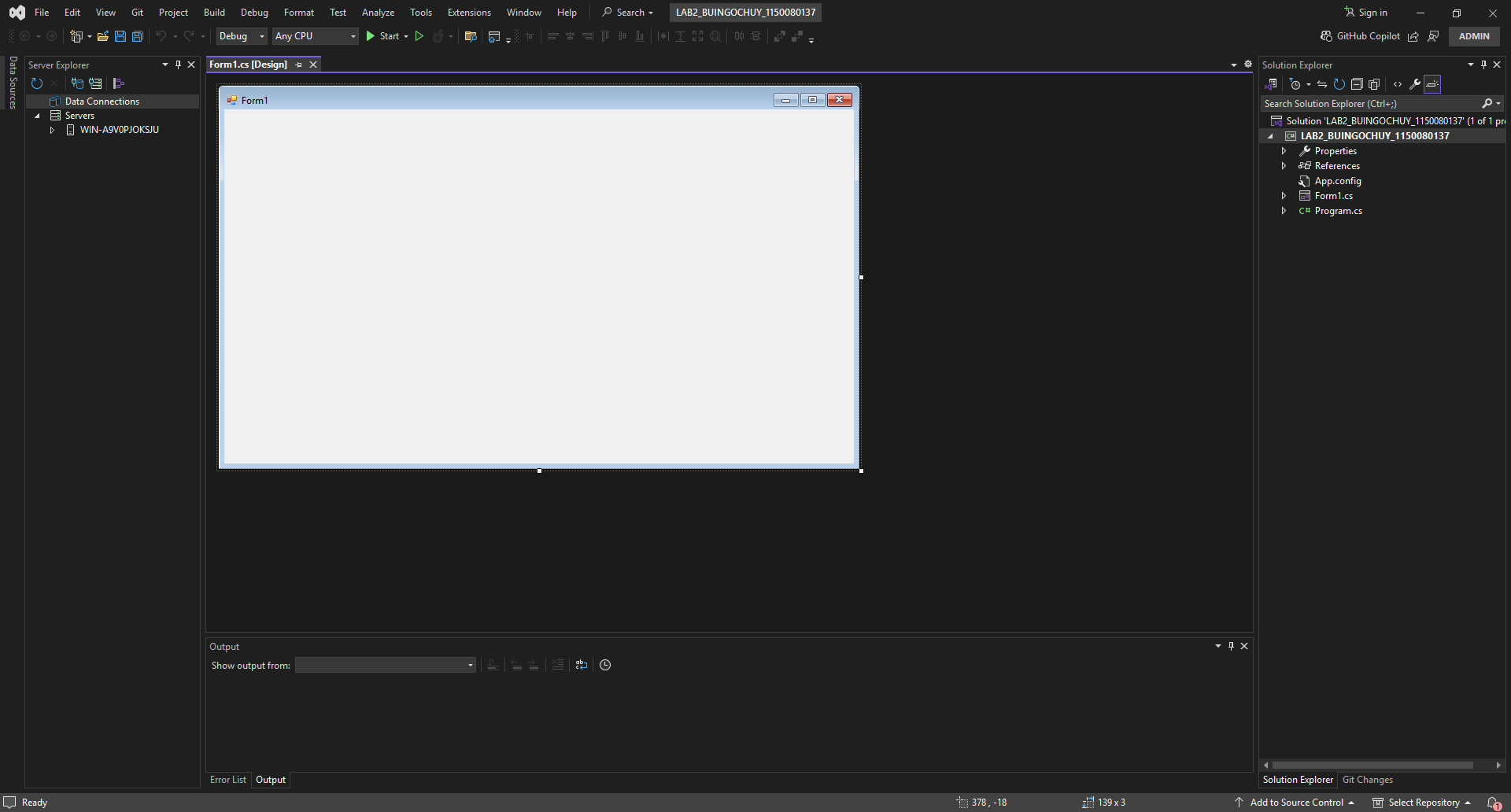
Bước 2: Chọn Create a new project trên màn hình khởi động. Trong màn hình tiếp theo, ở mục All languages → chọn C# trong danh sách ngôn ngữ lập trình. Mục All platforms → chọn Windows. Ở mục All project types → chọn Desktop.



Bước 3: Trong cửa sổ tiếp theo, bạn điền đầy đủ các thông tin về Project.



Đây là kết quả!



Màn hình làm việc cho ứng dụng Winform bao gồm các phần cơ bản sau:

Số 1: Toolbox → chứa các control cho phép kéo thả vào form.

Số 2: màn hình thiết kế → nơi thiết kế trực quan, có thể chuyển đổi qua lại giữa

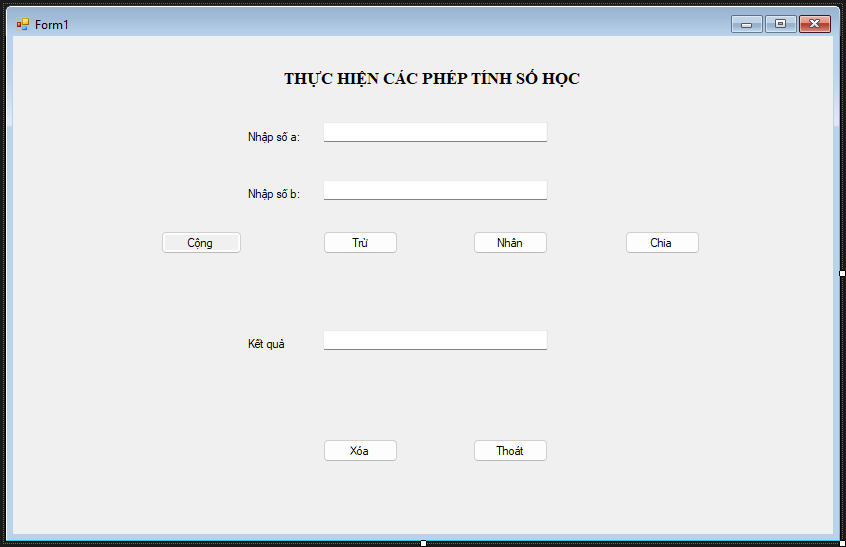
code editor và giao diện.

Số 3: Solution Exlorer → lập trình viên quản lý các thành phần trong dự án.

Số 4: Property → cho phép tùy chỉnh lại các thành phần của control và form.

2. Thực hành

**Thực hành 1:** Thiết kế form như sau:



Code

using System;

using System.Windows.Forms;

namespace LAB2\_BUINGOCHUY\_1150080137

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

// Hàm dùng chung để lấy số an toàn

private bool TryGetNumbers(out double a, out double b)

{

a = 0; b = 0;

if (!double.TryParse(txtA.Text.Trim(), out a))

{

MessageBox.Show("Giá trị nhập vào ô a không hợp lệ!", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtA.Focus();

return false;

}

if (!double.TryParse(txtB.Text.Trim(), out b))

{

MessageBox.Show("Giá trị nhập vào ô b không hợp lệ!", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtB.Focus();

return false;

}

return true;

}

// Nút Cộng

private void btnCong\_Click(object sender, EventArgs e)

{

double a, b;

if (TryGetNumbers(out a, out b))

{

txtKetQua.Text = (a + b).ToString();

}

}

// Nút Trừ

private void btnTru\_Click(object sender, EventArgs e)

{

double a, b;

if (TryGetNumbers(out a, out b))

{

txtKetQua.Text = (a - b).ToString();

}

}

// Nút Nhân

private void btnNhan\_Click(object sender, EventArgs e)

{

double a, b;

if (TryGetNumbers(out a, out b))

{

txtKetQua.Text = (a \* b).ToString();

}

}

// Nút Chia

private void btnChia\_Click(object sender, EventArgs e)

{

double a, b;

if (TryGetNumbers(out a, out b))

{

if (b == 0)

{

MessageBox.Show("Mẫu số không được phép bằng 0. Vui lòng nhập lại!", "Lỗi chia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtB.Clear();

txtB.Focus();

return; }

txtKetQua.Text = (a / b).ToString();

}

}

// Nút Xóa

private void btnXoa\_Click(object sender, EventArgs e)

{

txtA.Clear();

txtB.Clear();

txtKetQua.Clear();

txtA.Focus();

}

// Nút Thoát

private void btnThoat\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("Bạn có chắc muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

this.Close();

}

}

}

}

Kết quả



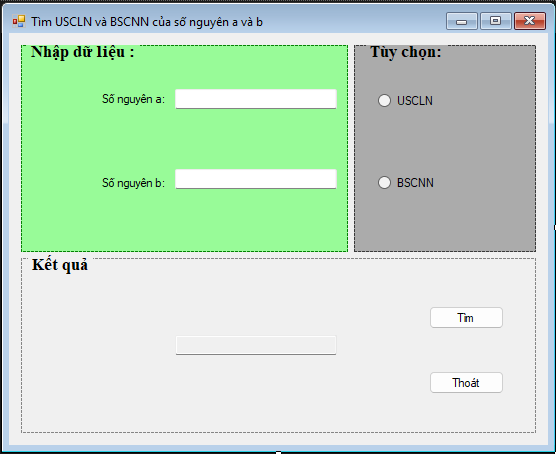
**Áp dụng 1:** Thêm vào một project có tên là “ApDung1”, thực hiện tìm ước số

chung lớn nhất và bội số chung nhỏ nhất của 2 số nguyên a và b:

Yêu cầu:

− Viết 2 hàm để tính ước số chung lớn nhất và bội số chung nhỏ nhất của 2 số a và b. Khi nhấn vào nút Tìm, nếu người dùng chọn tùy chọn tìm USCLN thì hiển thị ước số chung lớn nhất, ngược lại hiển thị bội số chung nhỏ nhất.

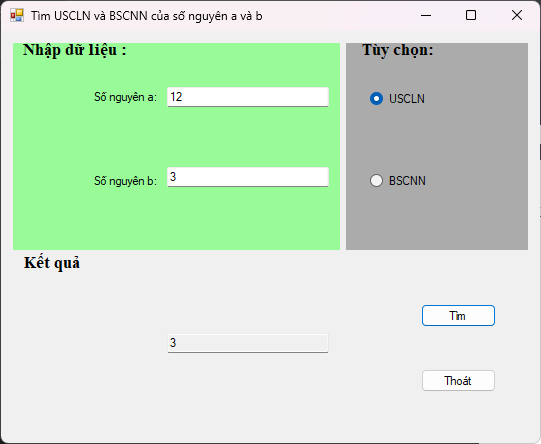
− Nút thoát để thoát khỏi chương trình.

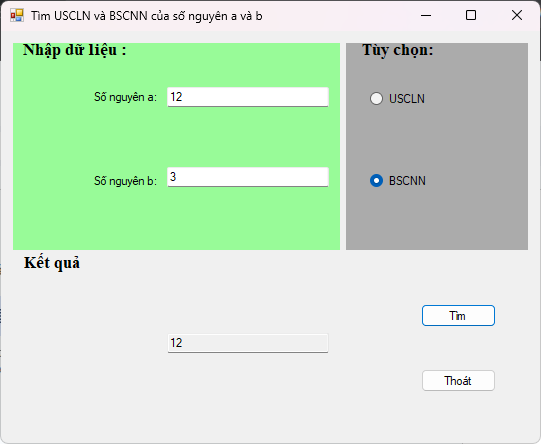


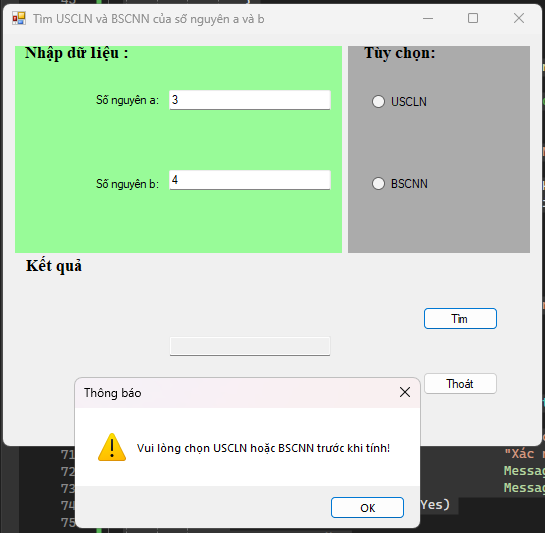
Code:

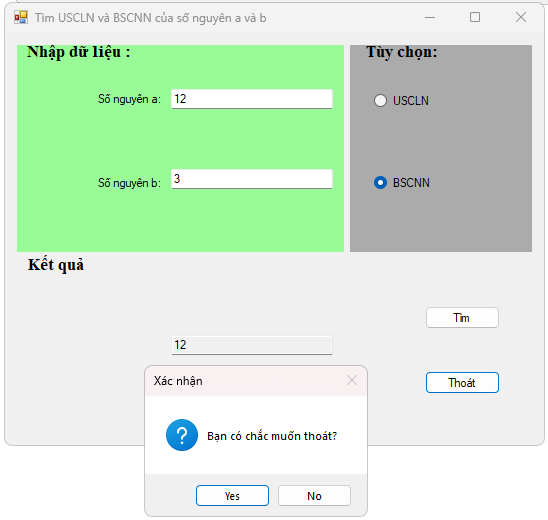
| using System;  using System.Windows.Forms;  namespace ApDung1  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  // Hàm tính Ước số chung lớn nhất (USCLN) bằng thuật toán Euclid  private int USCLN(int a, int b)  {  while (b != 0)  {  int temp = a % b;  a = b;  b = temp;  }  return a;  }  // Hàm tính Bội số chung nhỏ nhất (BSCNN)  private int BSCNN(int a, int b)  {  return (a \* b) / USCLN(a, b);  }  private void btnTim\_Click(object sender, EventArgs e)  {  try  {    int a = int.Parse(txtA.Text.Trim());  int b = int.Parse(txtB.Text.Trim());    if (radUSCLN.Checked)  {  txtKetQua.Text = USCLN(a, b).ToString();  }    else if (radBSCNN.Checked)  {  txtKetQua.Text = BSCNN(a, b).ToString();  }  // Nếu chưa chọn radio nào thì thông báo  else  {  MessageBox.Show("Vui lòng chọn USCLN hoặc BSCNN trước khi tính!",  "Thông báo",  MessageBoxButtons.OK,  MessageBoxIcon.Warning);  }  }  catch  {    MessageBox.Show("Vui lòng nhập số nguyên hợp lệ!");  }  }    private void btnThoat\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Bạn có chắc muốn thoát?",  "Xác nhận",  MessageBoxButtons.YesNo,  MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  this.Close();  }  }  }  } |
| --- |

Kết quả:



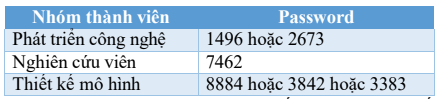






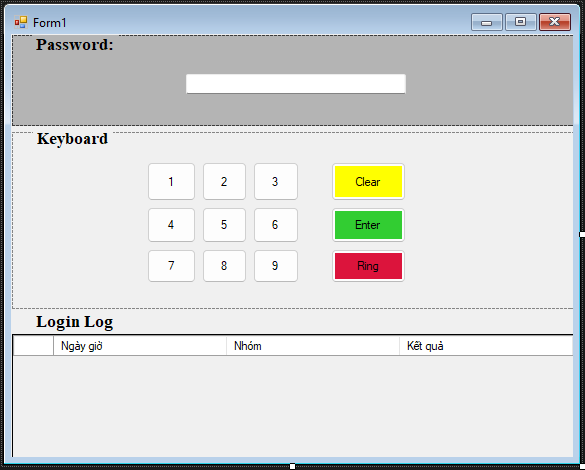
**Áp dụng 2:** Xây dựng ứng dụng mô phỏng thiết bị Security Panel theo yêu cầu:

Một phòng Lab muốn lắp thiết bị bảo vệ đặt được ở bên ngoài cửa ra vào có tên là Security. Chỉ cho phép những cá nhân nào nhập đúng password đã được quy định mới được vào phòng. Bảng dưới là password được cấp cho những nhóm thành viên:



Mỗi một lần nhập password sẽ có 2 trạng thái: chấp nhận hoặc từ chối. Tất cả các thông tin truy cập sẽ được hiển thị trong khung thông tin phía dưới bàn phím số. Nếu là chấp nhận thì ghi lại và hiển thị thông về về ngày, thời gian, tên nhóm và chấp nhận. Nếu là từ chối thì ghi lại và hiển thị thông về về ngày, thời gian, từ chối.

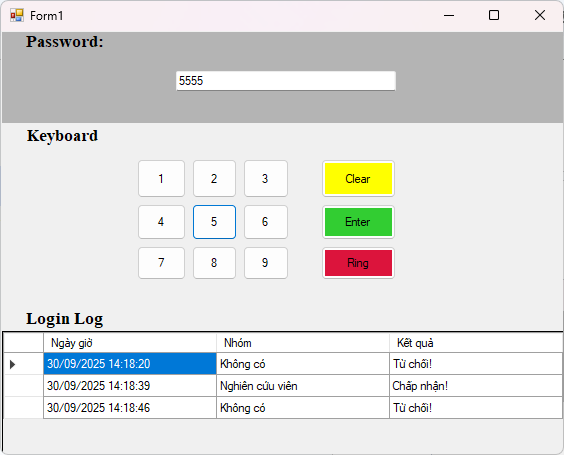
Giao diện ứng dụng:

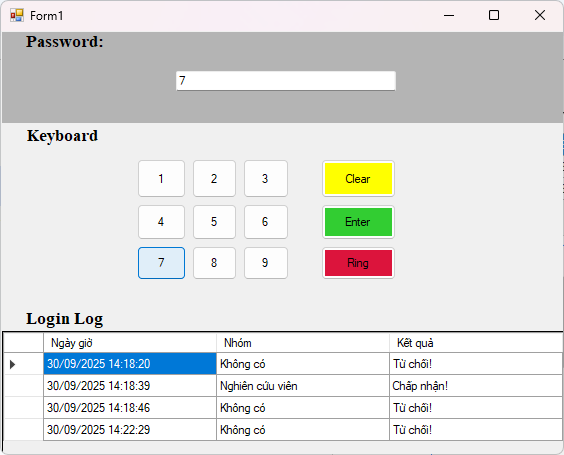


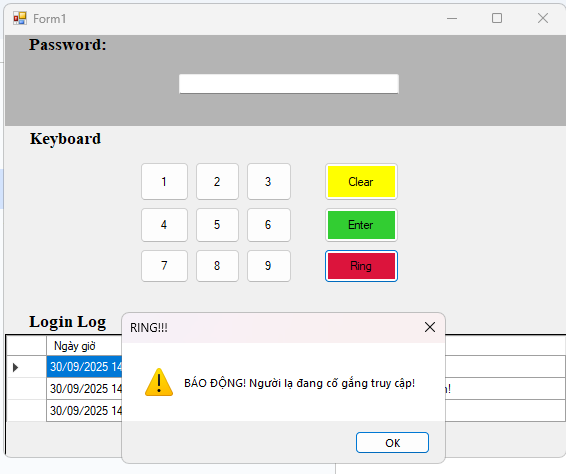
Code

| using System;  using System.Collections.Generic;  using System.Windows.Forms;  using static System.Windows.Forms.VisualStyles.VisualStyleElement.Window;  namespace ApDung2  {  public partial class Form1 : Form  {  // Danh sách nhóm và password hợp lệ  private Dictionary<string, List<string>> passwords;  public Form1()  {  InitializeComponent();  // Khởi tạo danh sách mật khẩu  passwords = new Dictionary<string, List<string>>()  {  { "Phát triển công nghệ", new List<string> { "1496", "2673" } },  { "Nghiên cứu viên", new List<string> { "7462" } },  { "Thiết kế mô hình", new List<string> { "8884", "3842", "3383" } }  };  // Setup DataGridView  dgvLog.ReadOnly = true;  dgvLog.AllowUserToAddRows = false;  dgvLog.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;  }  // Xử lý khi nhấn nút số (gán sự kiện cho btn1..btn9)  private void NumberButton\_Click(object sender, EventArgs e)  {  Button btn = sender as Button;  txtPassword.Text += btn.Text;  }  // Nút Clear  private void btnClear\_Click(object sender, EventArgs e)  {  txtPassword.Clear();  txtPassword.Focus();  }  // Nút Enter  private void btnEnter\_Click(object sender, EventArgs e)  {  string input = txtPassword.Text.Trim();  string groupName = "Không có";  string result = "Từ chối!";  // Kiểm tra mật khẩu  foreach (var pair in passwords)  {  if (pair.Value.Contains(input))  {  groupName = pair.Key;  result = "Chấp nhận!";  break;  }  }  // Thêm vào log (3 cột: Ngày giờ, Nhóm, Kết quả)  dgvLog.Rows.Add(DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss"), groupName, result);  // Reset password box  txtPassword.Clear();  txtPassword.Focus();  }  // Nút Ring  private void btnRing\_Click(object sender, EventArgs e)  {  MessageBox.Show("BÁO ĐỘNG! Người lạ đang cố gắng truy cập!",  "RING!!!",  MessageBoxButtons.OK,  MessageBoxIcon.Warning);  }  }  } |
| --- |

Kết quả:







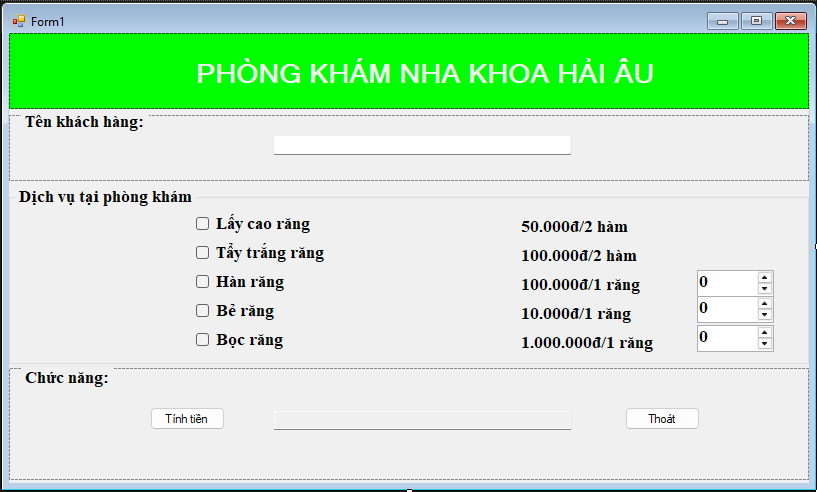
**Thực hành 2:** Tính tiền công dịch vụ tại phòng khám nha khoa.

Giả sử, tại một phòng khám nha khoa cần tính tiền các loại dịch vụ khác nhau.

Một biểu mẫu thanh toán gồm:

* Tên khách hàng.
* Các loại dịch vụ như: lấy cao răng (50.000đ), tẩy trắng răng (100.000đ), hàn răng (100.000đ), bẻ răng (10.000đ), bọc răng (1.000.000đ).
* Hàn răng, bẻ răng và bọc răng được tính theo chiếc răng. Giá tiền trên được tính theo 1 chiếc răng.
* Biểu mẫu tính tiền không được để trống tên khách hàng. Có cảnh báo nếu trống.

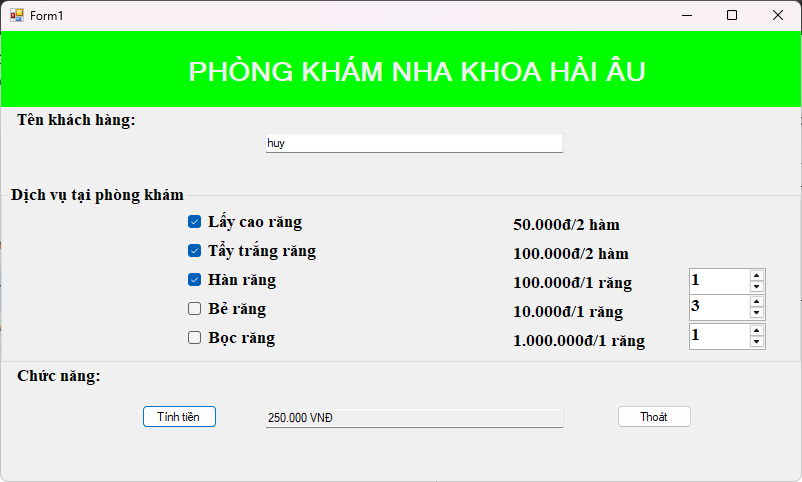
Tham khảo giao diện ứng dụng sau:

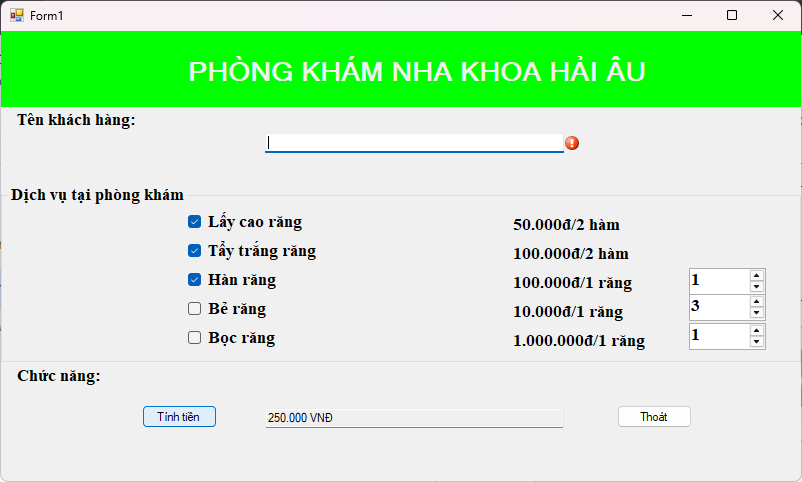


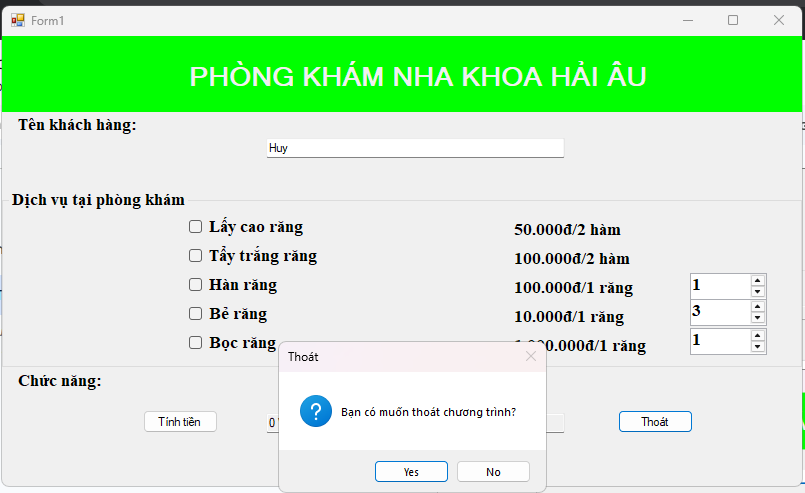
Code

| using System;  using System.ComponentModel;  using System.Windows.Forms;  namespace ThucHanh2  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  // Nút Tính tiền  private void btnTinhTien\_Click(object sender, EventArgs e)  {  double thanhTien = 0;  // Kiểm tra tên khách hàng  if (ValidateChildren(ValidationConstraints.Enabled))  {  if (chkLayCaoRang.Checked)  thanhTien += 50000;  if (chkTayTrangRang.Checked)  thanhTien += 100000;  if (chkHanRang.Checked)  thanhTien += (int)nupHanRang.Value \* 100000;  if (chkBeRang.Checked)  thanhTien += (int)nupBeRang.Value \* 10000;  if (chkBocRang.Checked)  thanhTien += (int)nupBocRang.Value \* 1000000;  txtThanhTien.Text = thanhTien.ToString("N0") + " VNĐ";  }  }  // Kiểm tra rỗng Tên khách hàng  private void txtTenKH\_Validating(object sender, CancelEventArgs e)  {  if (string.IsNullOrEmpty(txtTenKH.Text.Trim()))  {  e.Cancel = true;  txtTenKH.Focus();  err.SetError(txtTenKH, "Tên khách hàng không được để trống!");  }  else  {  e.Cancel = false;  err.SetError(txtTenKH, null);  }  }  // Nút Thoát  private void btnThoat\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Bạn có muốn thoát chương trình?",  "Thoát",  MessageBoxButtons.YesNo,  MessageBoxIcon.Question);  if (result == DialogResult.Yes)  {  this.Close();  }  }  }  } |
| --- |

Kết quả:





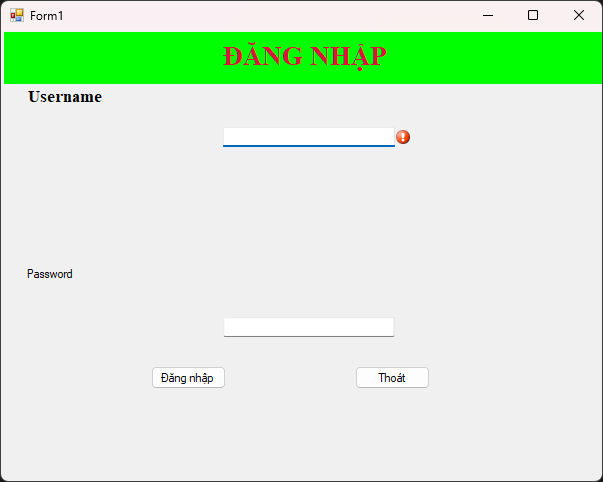


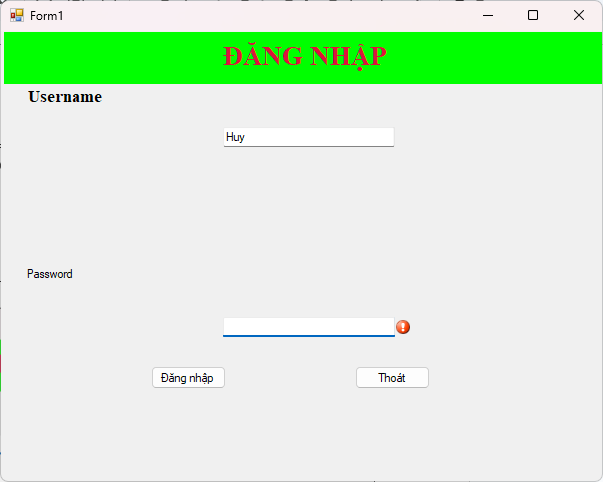
**Áp dụng 3:** Thiết kế và lập trình form Đăng nhập của một ứng dụng. Thông tin cần đăng nhập gồm username và password. Yêu cầu người dùng không được để trống trường thông tin nào. Nếu để trống trường thông tin nào thì cảnh báo người dùng để bổ sung.

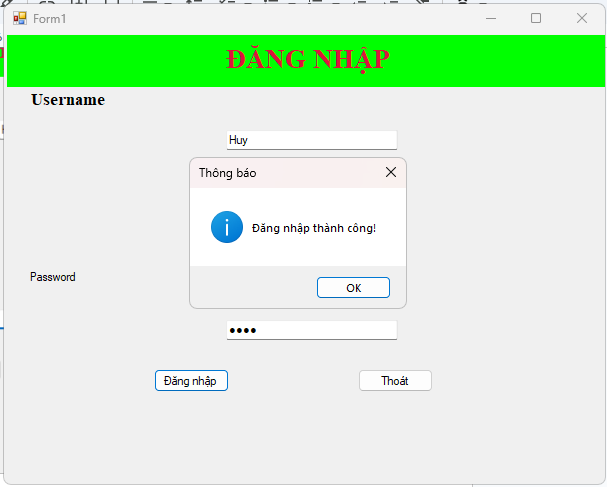
Code:

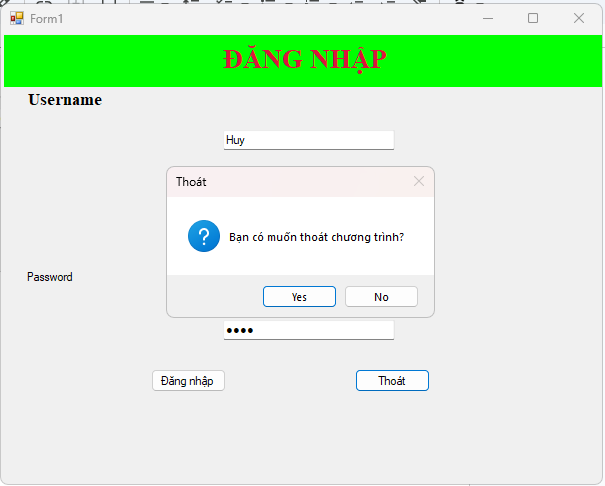
| using System;  using System.ComponentModel;  using System.Windows.Forms;  namespace Apdung3  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  // Gán sự kiện Validating bằng code (nếu chưa gán trong Designer)  this.txtUsername.Validating += new CancelEventHandler(txtUsername\_Validating);  this.txtPassword.Validating += new CancelEventHandler(txtPassword\_Validating);  }  // Nút Đăng nhập  private void btnLogin\_Click(object sender, EventArgs e)  {  // ép gọi sự kiện Validating cho tất cả control  if (ValidateChildren(ValidationConstraints.Enabled))  {  MessageBox.Show("Đăng nhập thành công!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  }  }  // Kiểm tra Username  private void txtUsername\_Validating(object sender, CancelEventArgs e)  {  if (string.IsNullOrWhiteSpace(txtUsername.Text))  {  e.Cancel = true;  err.SetError(txtUsername, "Username không được để trống!");  }  else  {  e.Cancel = false;  err.SetError(txtUsername, null);  }  }  // Kiểm tra Password  private void txtPassword\_Validating(object sender, CancelEventArgs e)  {  if (string.IsNullOrWhiteSpace(txtPassword.Text))  {  e.Cancel = true;  err.SetError(txtPassword, "Password không được để trống!");  }  else  {  e.Cancel = false;  err.SetError(txtPassword, null);  }  }  private void btnExit\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Bạn có muốn thoát chương trình?",  "Thoát",  MessageBoxButtons.YesNo,  MessageBoxIcon.Question);  if (result == DialogResult.Yes)  this.Close();  }  }  } |
| --- |

Kết quả









**Thực hành 3:** Thiết kế form frmXuLySo với control ListBox như sau:

Yêu cầu:

− Chỉ cho nhập số vào textbox Nhập số nguyên

− Khi nhập dữ liệu vào textbox Nhập số nguyên và nhấn Enter (hoặc click

vào nút Nhập số) thì số mới nhập này được thêm vào Listbox đồng thời dữ

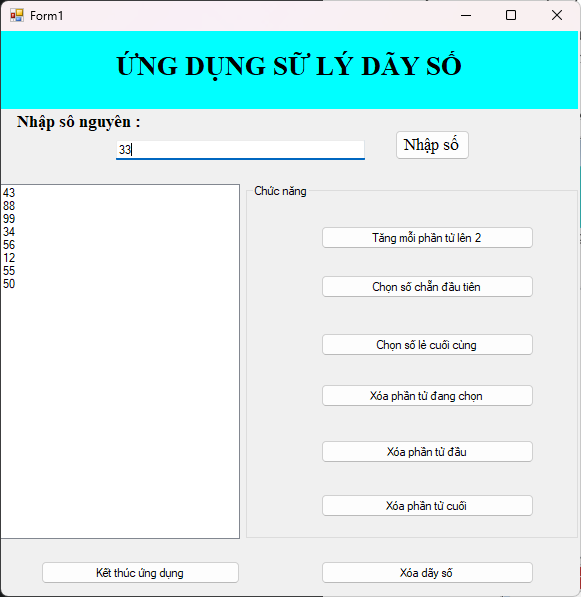
liệu trong textbox bị xóa và focus được chuyển về textbox

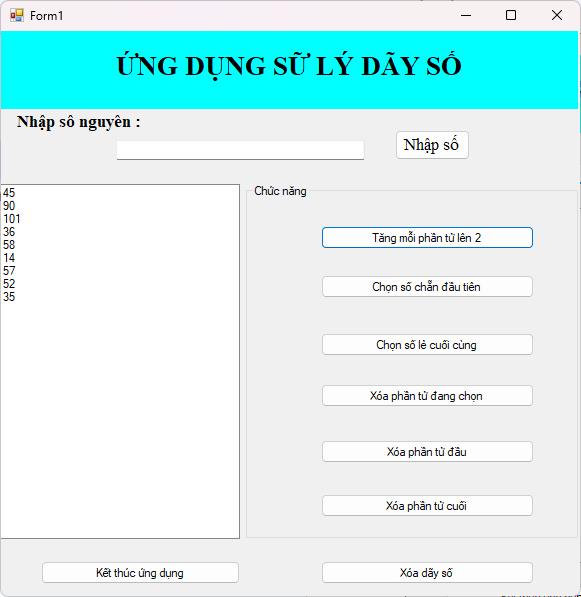
− Khi nhấn vào các phím chức năng, yêu cầu sẽ được thực hiện trên listbox.

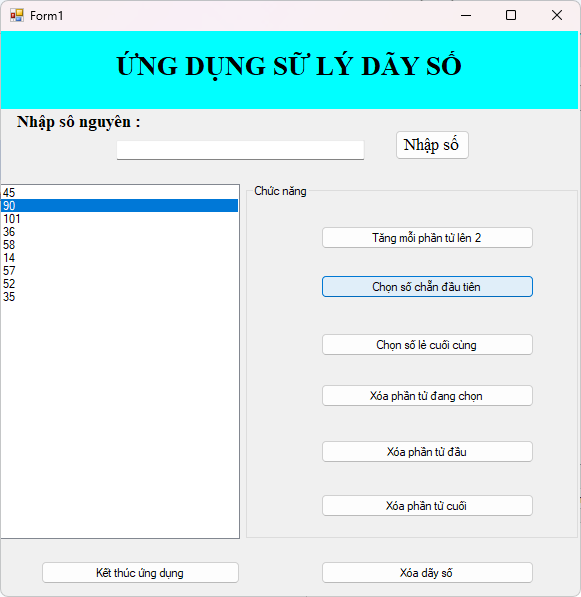
Code:

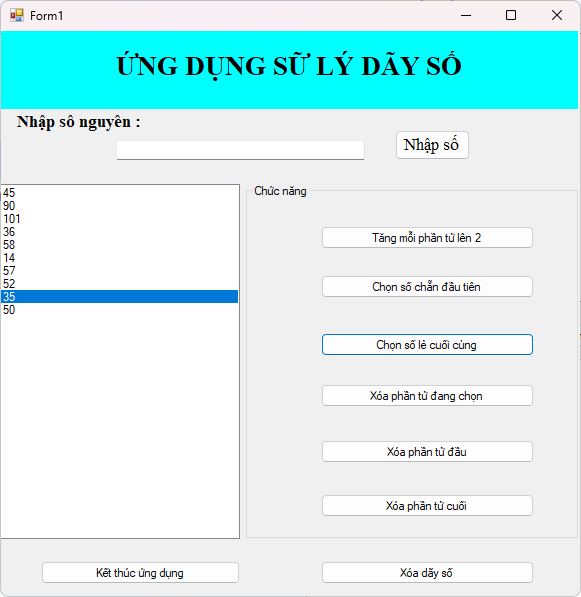
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace ThucHanh3  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  // Khi form load, clear list  private void Form1\_Load(object sender, EventArgs e)  {  lsbDaySo.Items.Clear();  }  // Nút nhập số  private void btnInput\_Click(object sender, EventArgs e)  {  int a;  string num = txtInput.Text.Trim();  if (int.TryParse(num, out a))  {  lsbDaySo.Items.Add(a);  txtInput.Clear();  txtInput.Focus();  }  else  {  MessageBox.Show("Không phải số nguyên! Hãy nhập lại!",  "Thông báo",  MessageBoxButtons.OK,  MessageBoxIcon.Warning);  txtInput.Clear();  txtInput.Focus();  }  }  // Enter trong TextBox = Nhập số  private void txtInput\_KeyDown(object sender, KeyEventArgs e)  {  if (e.KeyCode == Keys.Enter)  {  btnInput.PerformClick();  e.SuppressKeyPress = true; // chặn tiếng beep  }  }  // Nút tăng 2  private void btnTang2\_Click(object sender, EventArgs e)  {  if (lsbDaySo.Items.Count == 0)  {  MessageBox.Show("Dãy số đang trống. Vui lòng nhập dữ liệu!",  "Thông báo",  MessageBoxButtons.OK,  MessageBoxIcon.Error);  return;  }  for (int i = 0; i < lsbDaySo.Items.Count; i++)  {  int val = (int)lsbDaySo.Items[i];  lsbDaySo.Items[i] = val + 2;  }  }  // Chọn số chẵn đầu tiên  private void btnChanDau\_Click(object sender, EventArgs e)  {  for (int i = 0; i < lsbDaySo.Items.Count; i++)  {  if ((int)lsbDaySo.Items[i] % 2 == 0)  {  lsbDaySo.SelectedIndex = i;  return;  }  }  MessageBox.Show("Không tìm thấy số chẵn!", "Thông báo");  }  // Chọn số lẻ cuối cùng  private void btnLeCuoi\_Click(object sender, EventArgs e)  {  for (int i = lsbDaySo.Items.Count - 1; i >= 0; i--)  {  if ((int)lsbDaySo.Items[i] % 2 != 0)  {  lsbDaySo.SelectedIndex = i;  return;  }  }  MessageBox.Show("Không tìm thấy số lẻ!", "Thông báo");  }  // Xóa phần tử đang chọn  private void btnXoaChon\_Click(object sender, EventArgs e)  {  if (lsbDaySo.SelectedIndex == -1)  {  MessageBox.Show("Bạn chưa chọn số cần xóa!", "Thông báo",  MessageBoxButtons.OK,  MessageBoxIcon.Error);  return;  }  while (lsbDaySo.SelectedIndex != -1)  {  lsbDaySo.Items.RemoveAt(lsbDaySo.SelectedIndex);  }  }  // Xóa phần tử đầu  private void btnXoaDau\_Click(object sender, EventArgs e)  {  if (lsbDaySo.Items.Count > 0)  lsbDaySo.Items.RemoveAt(0);  }  // Xóa phần tử cuối  private void btnXoaCuoi\_Click(object sender, EventArgs e)  {  if (lsbDaySo.Items.Count > 0)  lsbDaySo.Items.RemoveAt(lsbDaySo.Items.Count - 1);  }  // Xóa dãy số  private void btnXoaDaySo\_Click(object sender, EventArgs e)  {  lsbDaySo.Items.Clear();  }  // Kết thúc ứng dụng  private void btnClose\_Click(object sender, EventArgs e)  {  DialogResult result = MessageBox.Show("Bạn có muốn thoát?",  "Thoát",  MessageBoxButtons.YesNo,  MessageBoxIcon.Question);  if (result == DialogResult.Yes)  this.Close();  }  private void txtInput\_KeyDown(object sender, CancelEventArgs e)  {  }  }  } |
| --- |

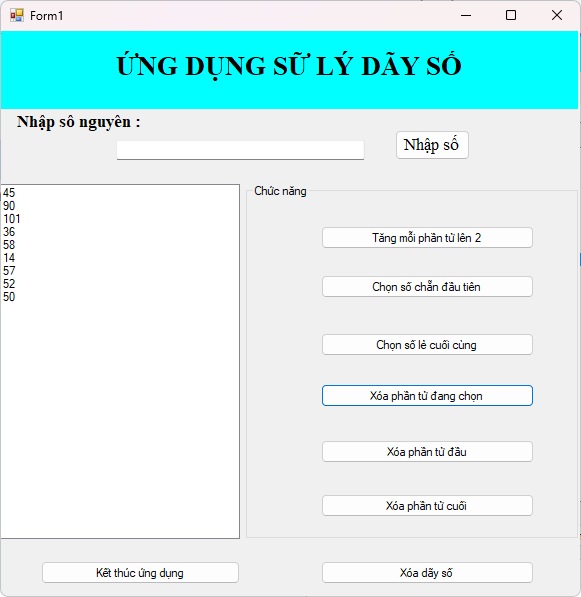
Kết quả

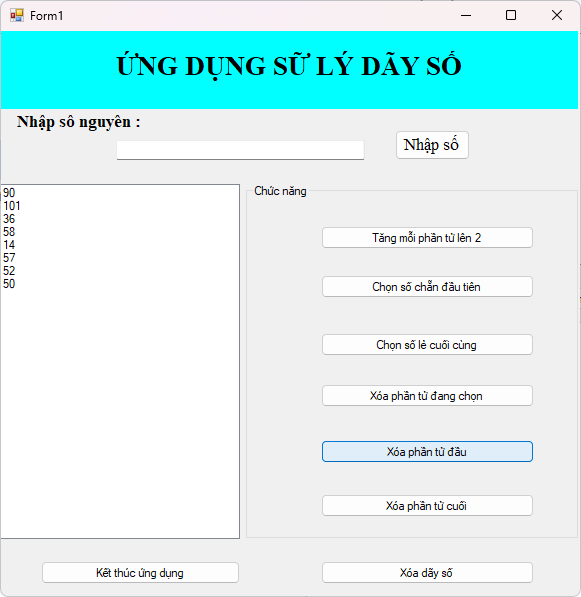


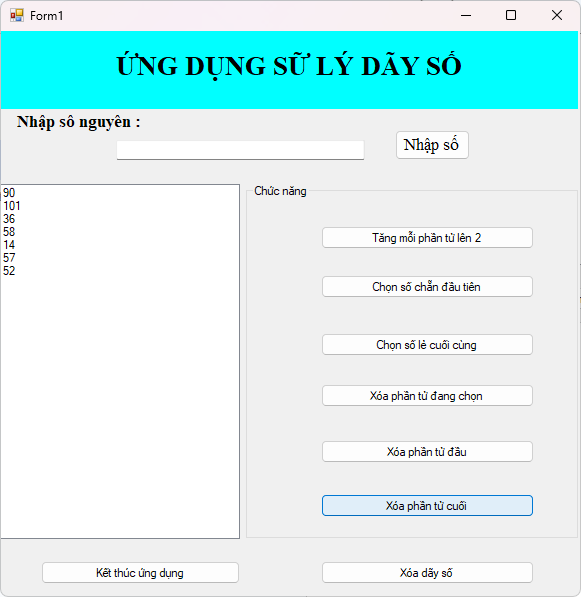


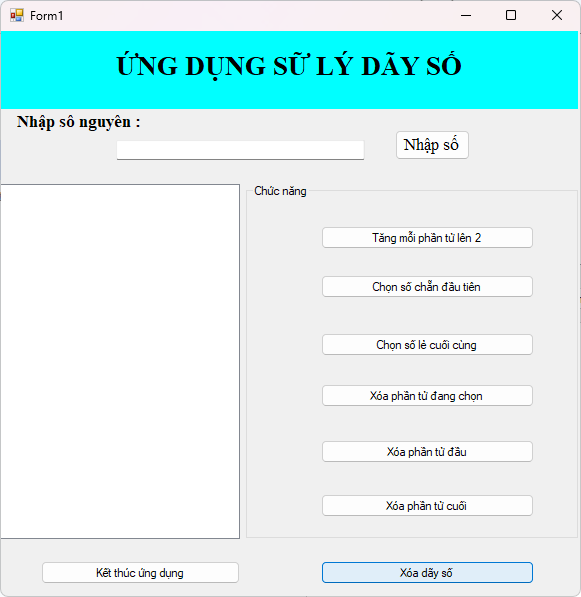


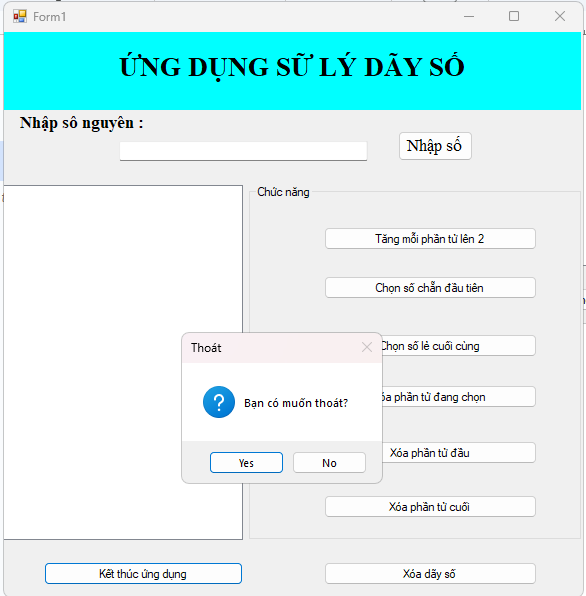












**Áp dụng 4:**Sử dụng Form, ListBox và Button

Thiết kế Form gồm có 2 ListBox, 2 Label và 4 nút lệnh có giao diện như sau:

Chọn các mặt hàng bên “Danh sách các mặt hàng”, khi Click vào nút >, sẽ chuyển các phần tử được lựa chọn sang bên “Các mặt hàng lựa chọn”, (đồng thời xoá đi các phần tử được lựa chọn bên “Danh sách các mặt hàng”)

Khi Click nút >> sẽ chuyển toàn bộ các phần tử bên “Danh sách các mặt hàng” sang bên “Các mặt hàng lựa chọn”, (đồng thời xoá trắng bên “Danh sách các mặt hàng”)

Chọn các mặt hàng bên “Các mặt hàng lựa chọn”, khi Click vào nút <, sẽ chuyển các phần tử được lựa chọn sang bên “Danh sách các mặt hàng”, (đồng thời xoá đi các phần tử được lựa chọn đó bên “Các mặt hàng lựa chọn”.

Khi Click nút << sẽ chuyển toàn bộ các phần tử bên “Các mặt hàng lựa chọn” sang bên “Danh sách các mặt hàng”, (đồng thời xoá trắng bên “Các mặt hàng lựa chọn”)

Code:

|  |
| --- |

Kết quả

