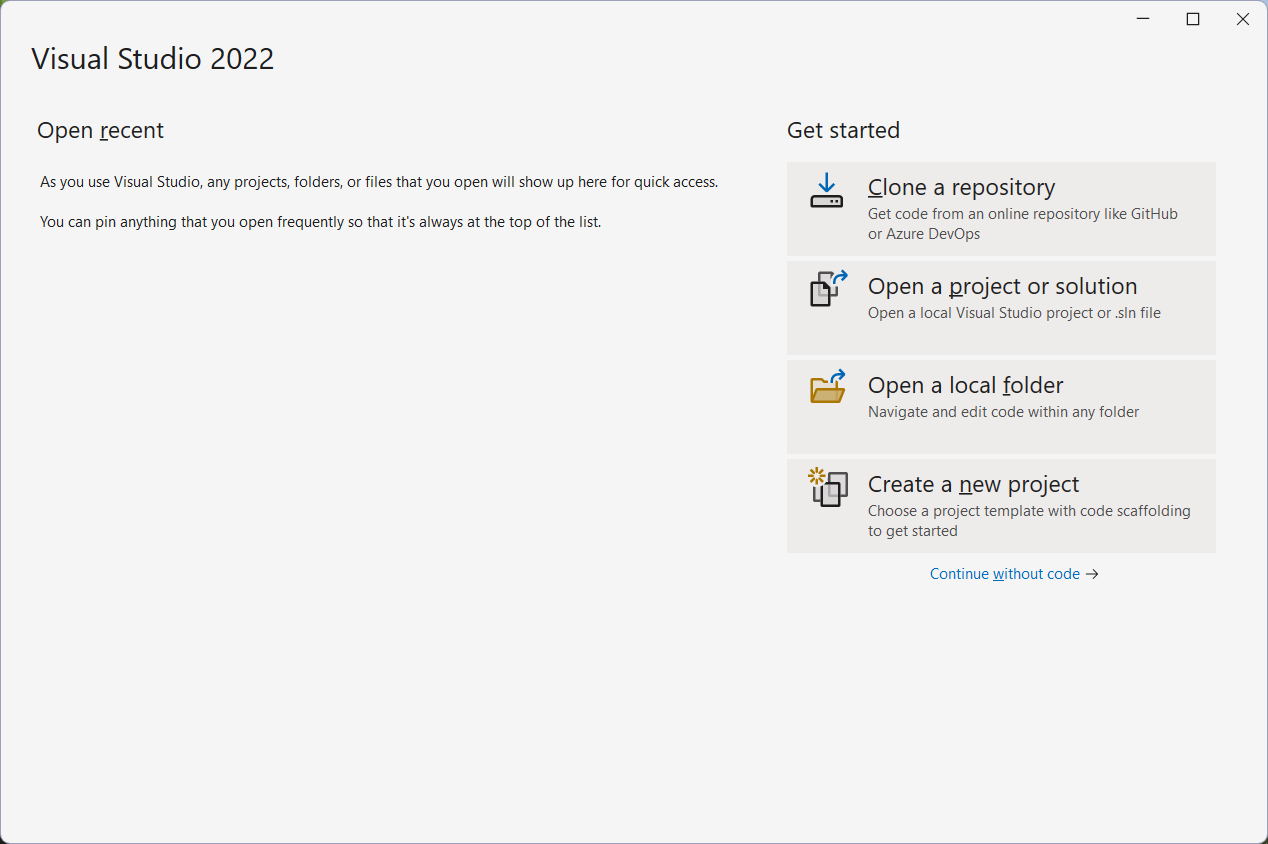
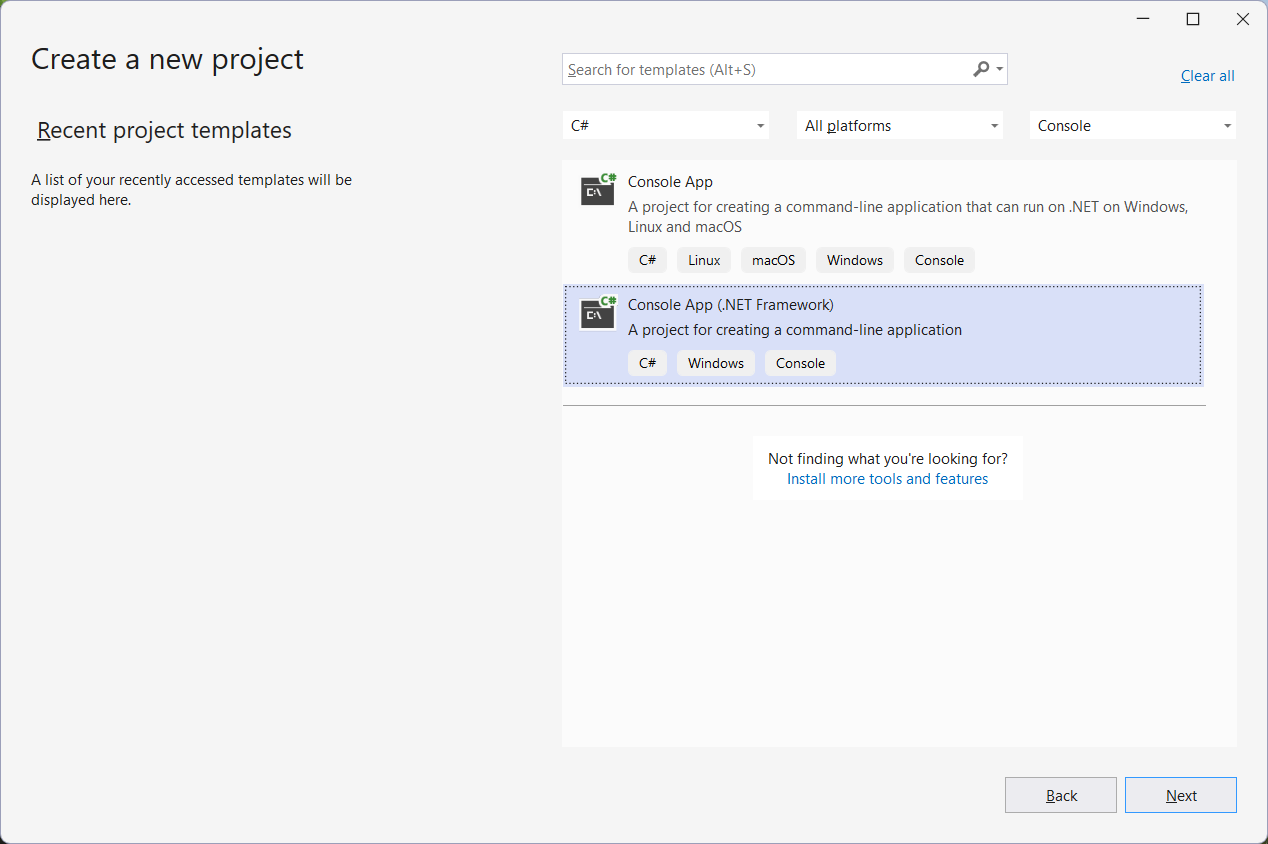
Họ tên: Hoàng Trọng Dũng

Lớp: 11\_ĐH\_CNPM2

MSSV: 1150080129





A screenshot of a computer

AI-generated content may be incorrect.

Thực hành 1: Sử dụng ngôn ngữ lập trình C# để tính và hiển thị chu vi, diện tích

của hình chữ nhật có chiều dài a và chiều rộng b được nhập vào từ bàn phím.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class Program

{

static void Main(string[] args)

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch

{

}

Console.Write("Nhập chiều dài: ");

string input = Console.ReadLine();

if (!TryReadDouble(input, out double length))

{

Console.WriteLine("Giá trị không hợp lệ cho chiều dài.");

Pause();

return;

}

Console.Write("Nhập chiều rộng: ");

input = Console.ReadLine();

if (!TryReadDouble(input, out double width))

{

Console.WriteLine("Giá trị không hợp lệ cho chiều rộng.");

Pause();

return;

}

double perimeter = 2 \* (length + width);

double area = length \* width;

Console.WriteLine($"Chu vi: {perimeter}");

Console.WriteLine($"Diện tích: {area}");

Pause();

}

static bool TryReadDouble(string s, out double value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

if (double.TryParse(s, out value)) return true;

s = s.Replace(',', '.');

return double.TryParse(s, System.Globalization.NumberStyles.Any, System.Globalization.CultureInfo.InvariantCulture, out value);

}

static void Pause()

{

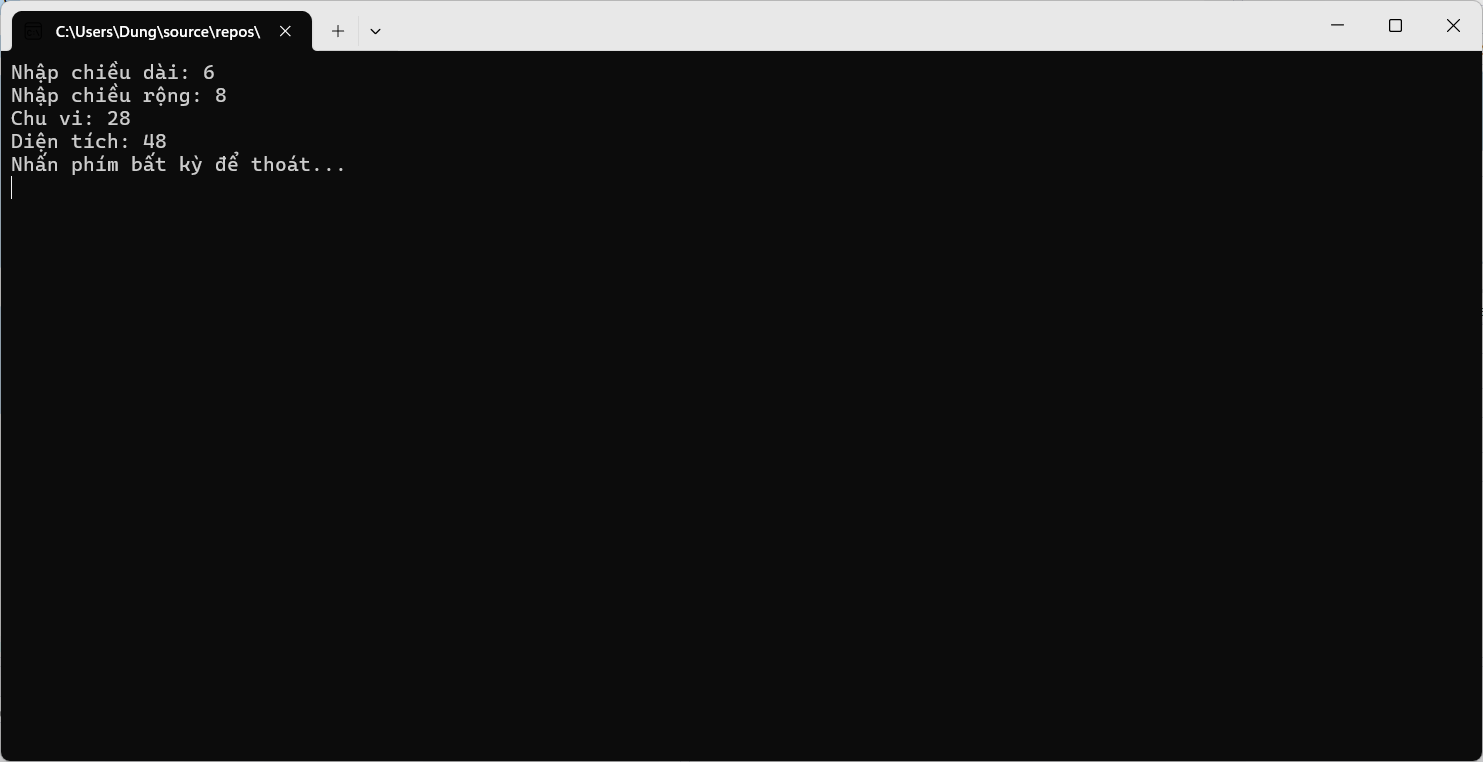
Console.WriteLine("Nhấn phím bất kỳ để thoát...");

Console.ReadKey();

}

}

}



Thực hành 2: Tìm số lớn hơn trong 2 số nguyên a và b được nhập vào từ bàn

phím.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai2

{

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.Write("Nhập số nguyên a: ");

string s = Console.ReadLine();

if (!TryReadInt(s, out int a))

{

Console.WriteLine("Giá trị không hợp lệ cho a.");

Pause();

return;

}

Console.Write("Nhập số nguyên b: ");

s = Console.ReadLine();

if (!TryReadInt(s, out int b))

{

Console.WriteLine("Giá trị không hợp lệ cho b.");

Pause();

return;

}

if (a > b)

{

Console.WriteLine($"Số lớn hơn là a = {a}");

}

else if (b > a)

{

Console.WriteLine($"Số lớn hơn là b = {b}");

}

else

{

Console.WriteLine("Hai số bằng nhau.");

}

Pause();

}

static bool TryReadInt(string s, out int value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return int.TryParse(s, out value);

}

static void Pause()

{

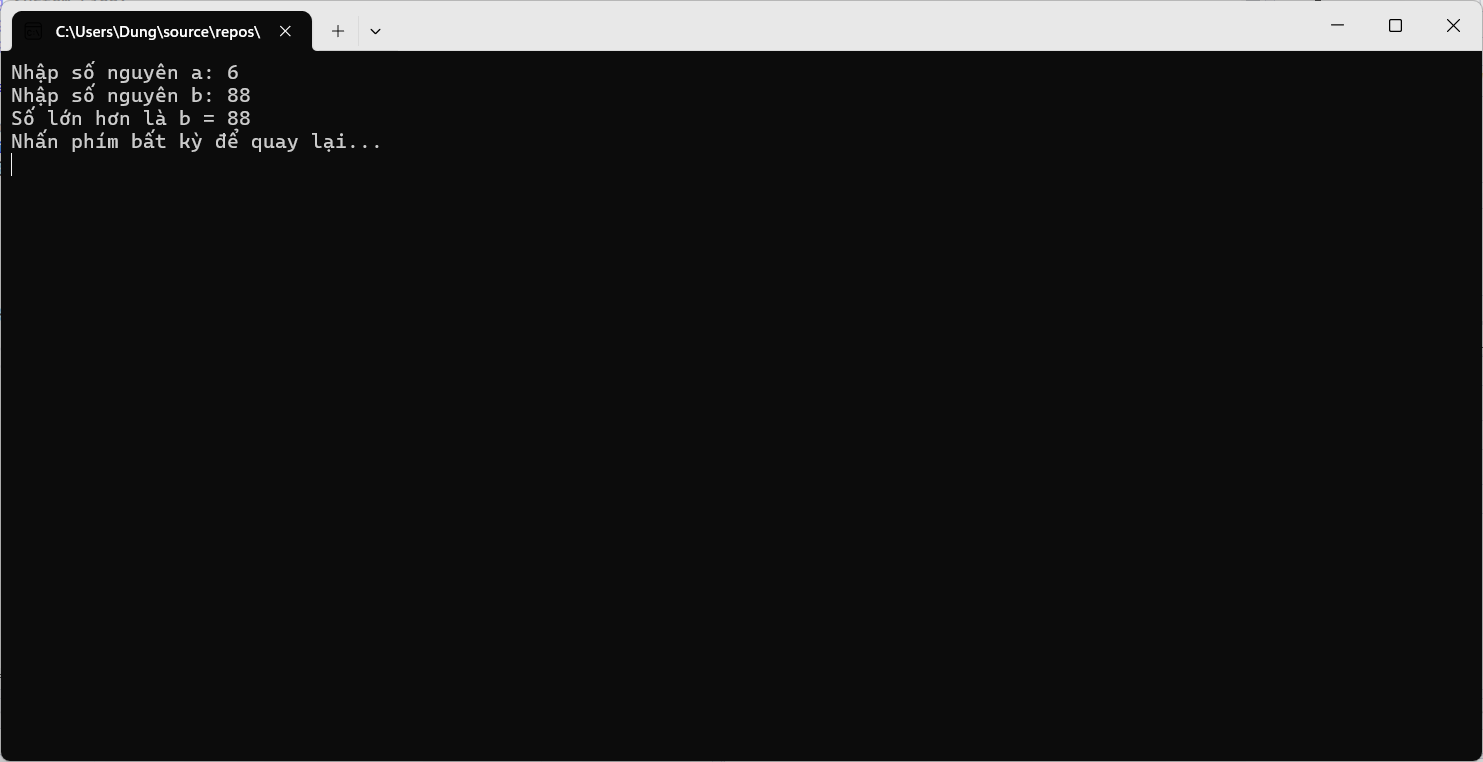
Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}



Thực hành 3: Tìm số có giá trị lớn nhất trong 3 số nguyên a, b, c được nhập vào

từ bàn phím.

using System;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai3

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(false);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.Write("Nhập số nguyên a: ");

string s = Console.ReadLine();

if (!TryReadInt(s, out int a))

{

Console.WriteLine("Giá trị không hợp lệ cho a.");

Pause();

return;

}

Console.Write("Nhập số nguyên b: ");

s = Console.ReadLine();

if (!TryReadInt(s, out int b))

{

Console.WriteLine("Giá trị không hợp lệ cho b.");

Pause();

return;

}

Console.Write("Nhập số nguyên c: ");

s = Console.ReadLine();

if (!TryReadInt(s, out int c))

{

Console.WriteLine("Giá trị không hợp lệ cho c.");

Pause();

return;

}

int max = a;

if (b > max) max = b;

if (c > max) max = c;

Console.WriteLine($"Số lớn nhất là: {max}");

Pause();

}

static bool TryReadInt(string s, out int value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return int.TryParse(s, out value);

}

static void Pause()

{

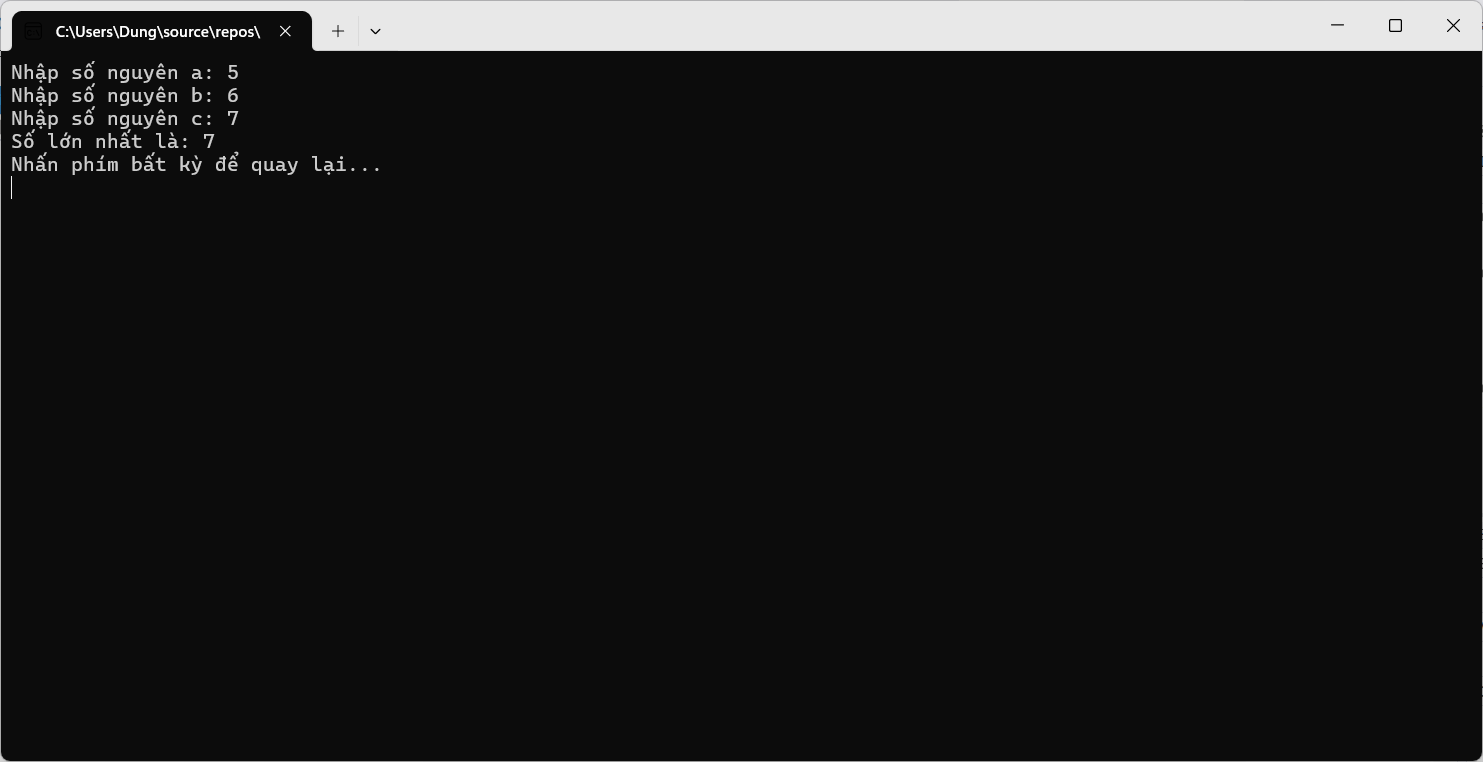
Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}



Thực hành 4: Cho biết một tháng có bao nhiêu ngày. Biết tháng, năm được nhập

vào từ bàn phím.

using System;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai4

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(true);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.Write("Nhập tháng (1-12): ");

string s = Console.ReadLine();

if (!TryReadInt(s, out int month) || month < 1 || month > 12)

{

Console.WriteLine("Giá trị tháng không hợp lệ.");

Pause();

return;

}

Console.Write("Nhập năm: ");

s = Console.ReadLine();

if (!TryReadInt(s, out int year) || year <= 0)

{

Console.WriteLine("Giá trị năm không hợp lệ.");

Pause();

return;

}

int days;

switch (month)

{

case 1: // Jan

case 3: // Mar

case 5: // May

case 7: // Jul

case 8: // Aug

case 10: // Oct

case 12: // Dec

days = 31;

break;

case 4: // Apr

case 6: // Jun

case 9: // Sep

case 11: // Nov

days = 30;

break;

case 2: // Feb

days = IsLeapYear(year) ? 29 : 28;

break;

default:

days = 0;

break;

}

Console.WriteLine($"Tháng {month} năm {year} có {days} ngày.");

Pause();

}

static bool TryReadInt(string s, out int value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return int.TryParse(s, out value);

}

static bool IsLeapYear(int year)

{

// Năm nhuận: chia hết cho 400 hoặc (chia hết cho 4 nhưng không chia hết cho 100)

return (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0);

}

static void Pause()

{

Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

Thực hành 5: Viết chương trình nhập vào một số nguyên n. Cho biết:

a) n là số chẵn hay số lẻ?

b) n là số âm hay số không âm?

using System;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai5

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(true);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.Write("Nhập một số nguyên n: ");

string s = Console.ReadLine();

if (!TryReadInt(s, out int n))

{

Console.WriteLine("Giá trị không hợp lệ.");

Pause();

return;

}

// a) chẵn hay lẻ

if (n % 2 == 0)

Console.WriteLine($"a) {n} là số chẵn.");

else

Console.WriteLine($"a) {n} là số lẻ.");

// b) âm hay không âm

if (n < 0)

Console.WriteLine($"b) {n} là số âm.");

else

Console.WriteLine($"b) {n} là số không âm.");

Pause();

}

static bool TryReadInt(string s, out int value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return int.TryParse(s, out value);

}

static void Pause()

{

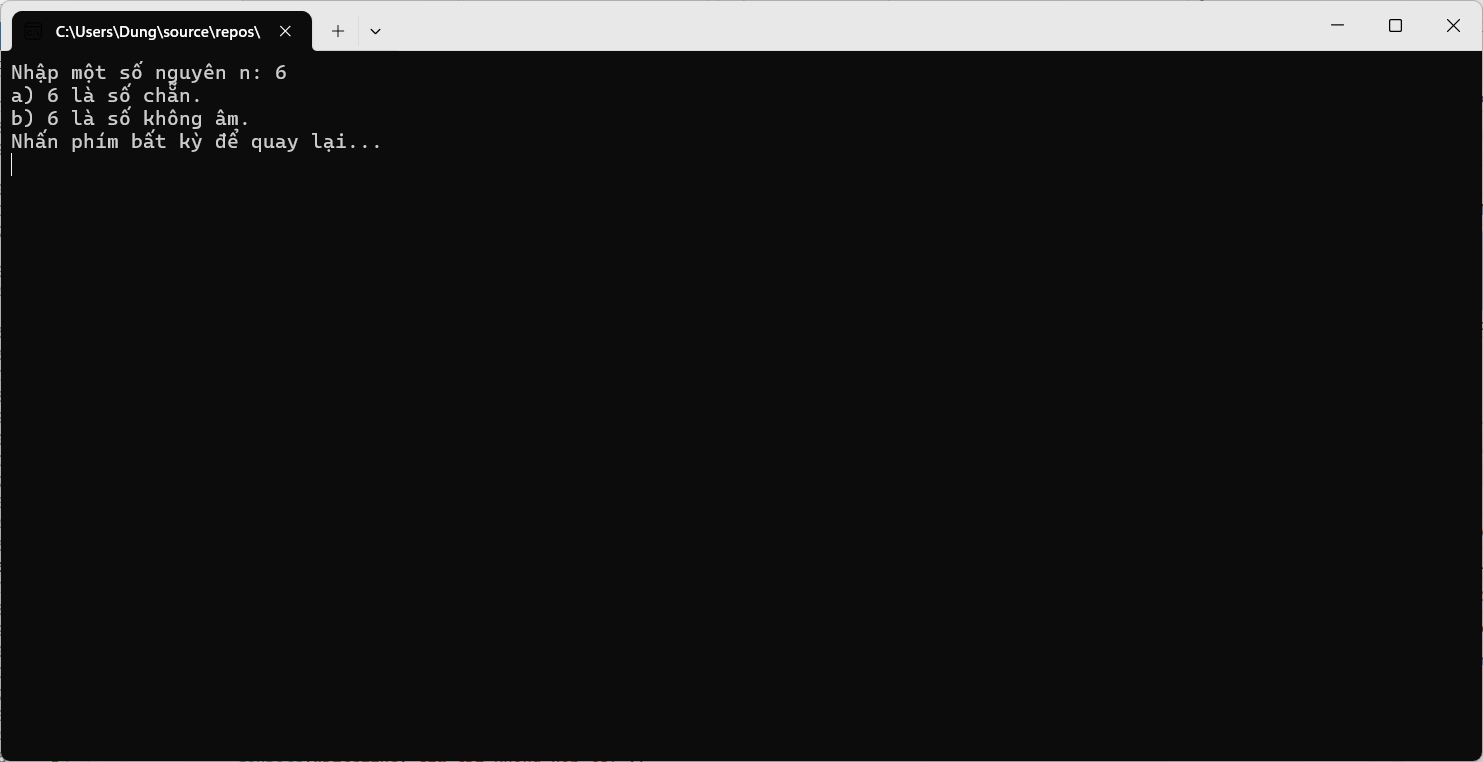
Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}



Thực hành 6: Viết chương trình nhập vào 2 số thực dương chỉ chiều dài và chiều

rộng của hình chữ nhật. In ra màn hình chu vi và diện tích của hình chữ nhật đó.

using System;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai6

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(true);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.Write("Nhập chiều dài (>0): ");

string s = Console.ReadLine();

if (!TryReadDouble(s, out double length) || length <= 0)

{

Console.WriteLine("Giá trị chiều dài không hợp lệ.");

Pause();

return;

}

Console.Write("Nhập chiều rộng (>0): ");

s = Console.ReadLine();

if (!TryReadDouble(s, out double width) || width <= 0)

{

Console.WriteLine("Giá trị chiều rộng không hợp lệ.");

Pause();

return;

}

double perimeter = 2 \* (length + width);

double area = length \* width;

Console.WriteLine($"Chu vi hình chữ nhật: {perimeter}");

Console.WriteLine($"Diện tích hình chữ nhật: {area}");

Pause();

}

static bool TryReadDouble(string s, out double value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return double.TryParse(s, out value);

}

static void Pause()

{

Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Thực hành 7: Viết chương trình nhập vào ba số thực chỉ độ dài của ba đoạn

thẳng. Kiểm tra nếu ba đoạn thẳng này lập thành được một tam giá thì hiển thị

chu vi và diện tích của tam giác đó.

using System;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai7

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(true);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.Write("Nhập độ dài đoạn thẳng thứ nhất (>0): ");

string s = Console.ReadLine();

if (!TryReadDouble(s, out double a) || a <= 0)

{

Console.WriteLine("Giá trị không hợp lệ cho đoạn thẳng thứ nhất.");

Pause();

return;

}

Console.Write("Nhập độ dài đoạn thẳng thứ hai (>0): ");

s = Console.ReadLine();

if (!TryReadDouble(s, out double b) || b <= 0)

{

Console.WriteLine("Giá trị không hợp lệ cho đoạn thẳng thứ hai.");

Pause();

return;

}

Console.Write("Nhập độ dài đoạn thẳng thứ ba (>0): ");

s = Console.ReadLine();

if (!TryReadDouble(s, out double c) || c <= 0)

{

Console.WriteLine("Giá trị không hợp lệ cho đoạn thẳng thứ ba.");

Pause();

return;

}

// Kiểm tra điều kiện tam giác: tổng hai cạnh phải lớn hơn cạnh còn lại

if (a + b > c && a + c > b && b + c > a)

{

double perimeter = a + b + c;

double sHalf = perimeter / 2.0;

double area = Math.Sqrt(Math.Max(0.0, sHalf \* (sHalf - a) \* (sHalf - b) \* (sHalf - c)));

Console.WriteLine($"Tam giác hợp lệ.");

Console.WriteLine($"Chu vi: {perimeter}");

Console.WriteLine($"Diện tích: {area}");

}

else

{

Console.WriteLine("Ba đoạn thẳng không lập thành tam giác.");

}

Pause();

}

static bool TryReadDouble(string s, out double value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return double.TryParse(s, out value);

}

static void Pause()

{

Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}

A computer screen with white text

AI-generated content may be incorrect.

Thực hành 8: Viết chương trình giải phương trình bậc 2: ax2 + bx + c = 0.

using System;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai8

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(true);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.WriteLine("Giải phương trình bậc 2: ax^2 + bx + c = 0");

Console.Write("Nhập a: ");

string s = Console.ReadLine();

if (!TryReadDouble(s, out double a))

{

Console.WriteLine("Giá trị không hợp lệ cho a.");

Pause();

return;

}

Console.Write("Nhập b: ");

s = Console.ReadLine();

if (!TryReadDouble(s, out double b))

{

Console.WriteLine("Giá trị không hợp lệ cho b.");

Pause();

return;

}

Console.Write("Nhập c: ");

s = Console.ReadLine();

if (!TryReadDouble(s, out double c))

{

Console.WriteLine("Giá trị không hợp lệ cho c.");

Pause();

return;

}

const double EPS = 1e-12;

if (Math.Abs(a) < EPS)

{

// Linear or degenerate

if (Math.Abs(b) < EPS)

{

if (Math.Abs(c) < EPS)

Console.WriteLine("Phương trình có vô số nghiệm.");

else

Console.WriteLine("Phương trình vô nghiệm.");

}

else

{

double x = -c / b;

Console.WriteLine($"Phương trình bậc nhất có nghiệm: x = {x}");

}

Pause();

return;

}

double delta = b \* b - 4 \* a \* c;

if (delta > EPS)

{

double sqrtD = Math.Sqrt(delta);

double x1 = (-b + sqrtD) / (2 \* a);

double x2 = (-b - sqrtD) / (2 \* a);

Console.WriteLine($"Phương trình có 2 nghiệm thực phân biệt:");

Console.WriteLine($"x1 = {x1}");

Console.WriteLine($"x2 = {x2}");

}

else if (Math.Abs(delta) <= EPS)

{

double x = -b / (2 \* a);

Console.WriteLine($"Phương trình có nghiệm kép: x = {x}");

}

else

{

// complex roots

double real = -b / (2 \* a);

double imag = Math.Sqrt(-delta) / (2 \* a);

Console.WriteLine($"Phương trình có 2 nghiệm phức:");

Console.WriteLine($"x1 = {real} + {imag}i");

Console.WriteLine($"x2 = {real} - {imag}i");

}

Pause();

}

static bool TryReadDouble(string s, out double value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return double.TryParse(s, out value);

}

static void Pause()

{

Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}

A computer screen with a black background

AI-generated content may be incorrect.

Thực hành 9: Tính tổng các phần tử trong mảng.

using System;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai9

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(true);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

Console.Write("Nhập số lượng phần tử n (>=1): ");

string s = Console.ReadLine();

if (!TryReadInt(s, out int n) || n <= 0)

{

Console.WriteLine("Giá trị n không hợp lệ.");

Pause();

return;

}

double sum = 0.0;

for (int i = 0; i < n; i++)

{

double val;

while (true)

{

Console.Write($"Nhập phần tử thứ {i + 1}: ");

s = Console.ReadLine();

if (TryReadDouble(s, out val)) break;

Console.WriteLine("Giá trị không hợp lệ, vui lòng nhập lại.");

}

sum += val;

}

Console.WriteLine($"Tổng các phần tử trong mảng là: {sum}");

Pause();

}

static bool TryReadInt(string s, out int value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return int.TryParse(s, out value);

}

static bool TryReadDouble(string s, out double value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return double.TryParse(s, out value);

}

static void Pause()

{

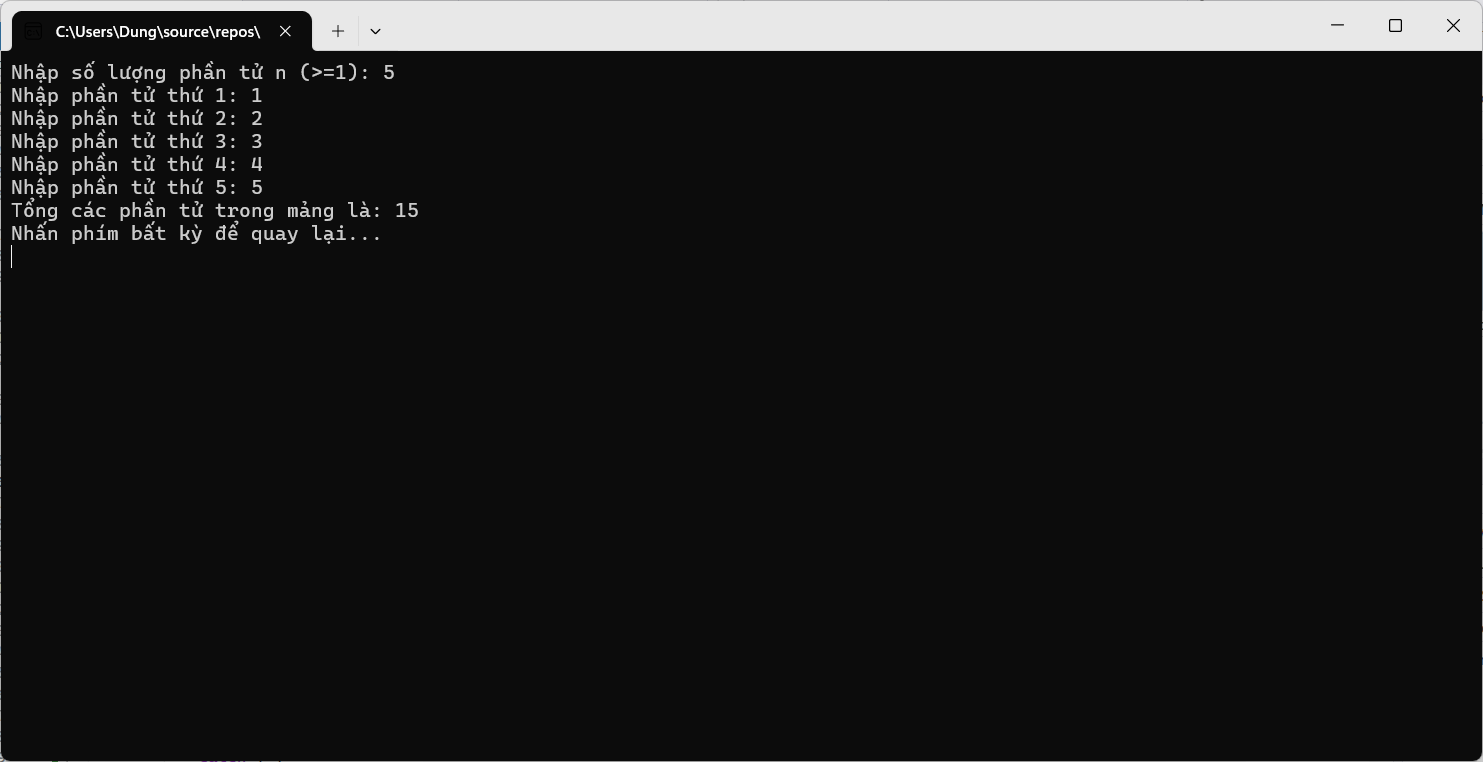
Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}



Thực hành 10: Dùng giải thuật sắp xếp chọn (Selection Sort) để sắp xếp tăng

dần mảng các số nguyên. Mảng các số nguyên được lưu trong tệp văn bản có tên

là “input\_array.txt”.

using System;

using System.IO;

using System.Linq;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai10

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(false);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

string[] tryPaths = new[]

{

"input\_array.txt",

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "input\_array.txt"),

Path.Combine(Directory.GetCurrentDirectory(), "input\_array.txt")

};

string path = tryPaths.FirstOrDefault(File.Exists);

if (path == null)

{

Console.WriteLine("Không tìm thấy file 'input\_array.txt' trong thư mục hiện tại.");

Console.WriteLine("Hãy chắc chắn file tồn tại và được đặt trong thư mục chạy chương trình.");

Pause();

return;

}

string text;

try

{

text = File.ReadAllText(path);

}

catch (Exception ex)

{

Console.WriteLine($"Không thể đọc file: {ex.Message}");

Pause();

return;

}

var tokens = text.Split((char[])null, StringSplitOptions.RemoveEmptyEntries);

int[] arr;

try

{

arr = tokens.Select(t => int.Parse(t.Trim())).ToArray();

}

catch (Exception)

{

Console.WriteLine("File chứa dữ liệu không hợp lệ. Hãy cung cấp các số nguyên cách nhau bằng khoảng trắng hoặc xuống dòng.");

Pause();

return;

}

Console.WriteLine("Mảng ban đầu: " + string.Join(" ", arr));

// Selection sort ascending

SelectionSort(arr);

Console.WriteLine("Mảng sau khi sắp xếp tăng dần: " + string.Join(" ", arr));

Pause();

}

static void SelectionSort(int[] a)

{

int n = a.Length;

for (int i = 0; i < n - 1; i++)

{

int minIdx = i;

for (int j = i + 1; j < n; j++)

{

if (a[j] < a[minIdx]) minIdx = j;

}

if (minIdx != i)

{

int tmp = a[i];

a[i] = a[minIdx];

a[minIdx] = tmp;

}

}

}

static void Pause()

{

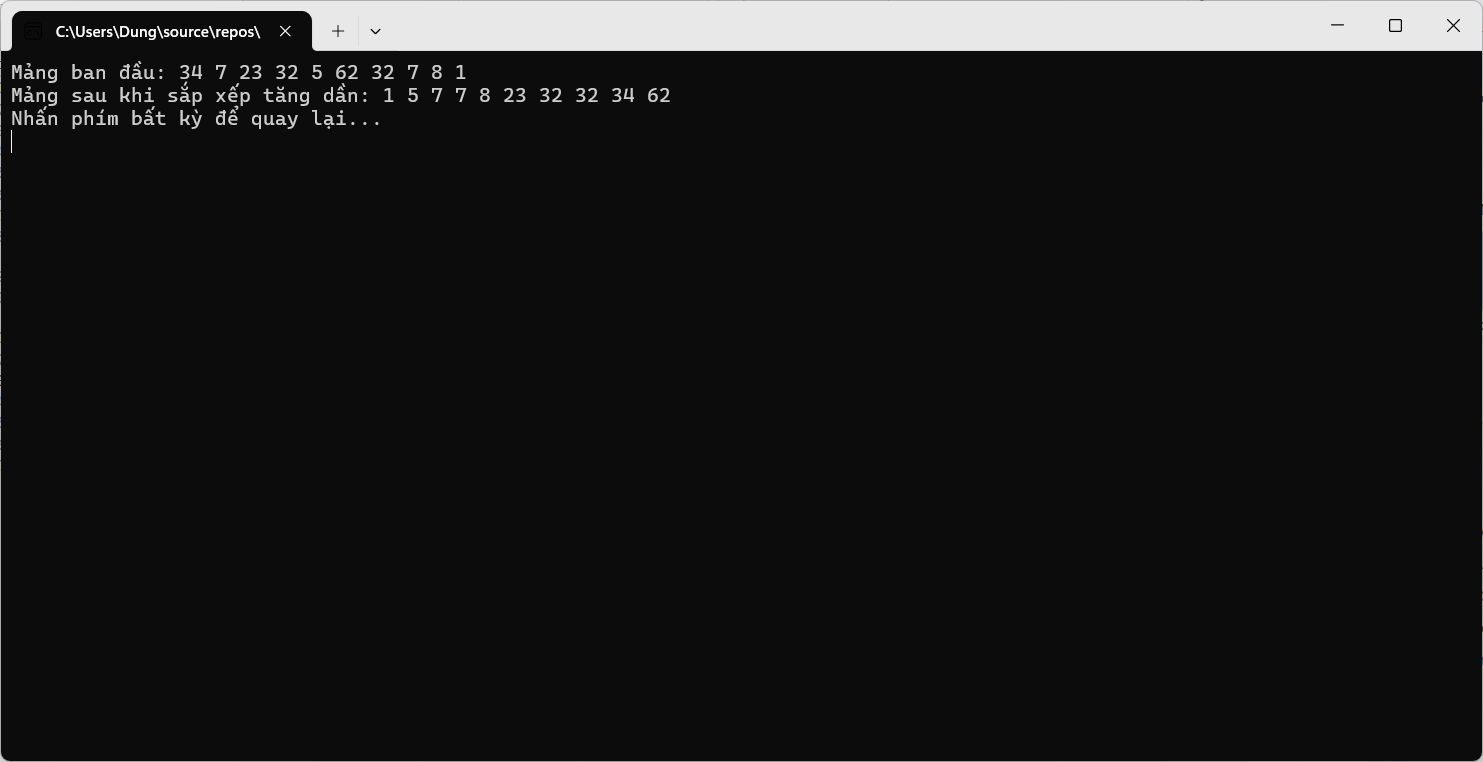
Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}



Thực hành 11: Chèn thêm một số nguyên được nhạp vào từ bàn phím vào mảng

đã sắp xếp tăng dần nhưng không làm mất tính tăng dần của mảng.

using System;

using System.IO;

using System.Linq;

using System.Text;

using System.Runtime.InteropServices;

namespace HoangTrongDung\_1150080129\_BTtuan5

{

internal class bai11

{

[DllImport("kernel32.dll")]

private static extern bool SetConsoleOutputCP(uint wCodePageID);

[DllImport("kernel32.dll")]

private static extern bool SetConsoleCP(uint wCodePageID);

public static void Run()

{

try

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

}

catch { }

try

{

Console.OutputEncoding = new UTF8Encoding(true);

Console.InputEncoding = Encoding.UTF8;

}

catch { }

string[] tryPaths = new[]

{

"input\_array.txt",

Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "input\_array.txt"),

Path.Combine(Directory.GetCurrentDirectory(), "input\_array.txt")

};

string path = tryPaths.FirstOrDefault(File.Exists);

if (path == null)

{

Console.WriteLine("Không tìm thấy file 'input\_array.txt' trong thư mục hiện tại.");

Console.WriteLine("Hãy chắc chắn file tồn tại và được đặt trong thư mục chạy chương trình.");

Pause();

return;

}

string text;

try

{

text = File.ReadAllText(path);

}

catch (Exception ex)

{

Console.WriteLine($"Không thể đọc file: {ex.Message}");

Pause();

return;

}

var tokens = text.Split((char[])null, StringSplitOptions.RemoveEmptyEntries);

int[] arr;

try

{

arr = tokens.Select(t => int.Parse(t.Trim())).ToArray();

}

catch (Exception)

{

Console.WriteLine("File chứa dữ liệu không hợp lệ. Hãy cung cấp các số nguyên cách nhau bằng khoảng trắng hoặc xuống dòng.");

Pause();

return;

}

Array.Sort(arr);

Console.WriteLine("Mảng đã sắp xếp tăng dần: " + string.Join(" ", arr));

int newValue;

while (true)

{

Console.Write("Nhập số nguyên cần chèn vào mảng: ");

string input = Console.ReadLine();

if (TryReadInt(input, out newValue))

break;

Console.WriteLine("Giá trị không hợp lệ, vui lòng nhập lại.");

}

int[] newArr = InsertIntoSortedArray(arr, newValue);

Console.WriteLine($"Mảng sau khi chèn số {newValue}: " + string.Join(" ", newArr));

Console.WriteLine($"Kích thước mảng: {arr.Length} → {newArr.Length}");

Pause();

}

static int[] InsertIntoSortedArray(int[] sortedArray, int value)

{

int insertPosition = BinarySearchInsertPosition(sortedArray, value);

int[] newArray = new int[sortedArray.Length + 1];

for (int i = 0; i < insertPosition; i++)

{

newArray[i] = sortedArray[i];

}

newArray[insertPosition] = value;

for (int i = insertPosition; i < sortedArray.Length; i++)

{

newArray[i + 1] = sortedArray[i];

}

return newArray;

}

static int BinarySearchInsertPosition(int[] sortedArray, int value)

{

int left = 0;

int right = sortedArray.Length;

while (left < right)

{

int mid = left + (right - left) / 2;

if (sortedArray[mid] <= value)

{

left = mid + 1;

}

else

{

right = mid;

}

}

return left;

}

static bool TryReadInt(string s, out int value)

{

value = 0;

if (string.IsNullOrWhiteSpace(s)) return false;

s = s.Trim();

return int.TryParse(s, out value);

}

static void Pause()

{

Console.WriteLine("Nhấn phím bất kỳ để quay lại...");

Console.ReadKey();

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

Thực hành 1: Thiết kế Form này sẽ cho phép người dùng nhập các giá trị là số thực vào 2 biến a và b. Tùy vào các chức năng của các phép toán (cộng, trừ, nhân, chia) mà hiển thị ở phần kết quả.

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 WindowsForms\_1150080129\_HoangTrongDung\_BTbuoi5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

btnCong.Click += BtnCong\_Click;

btnTru.Click += BtnTru\_Click;

btnNhan.Click += BtnNhan\_Click;

btnChia.Click += BtnChia\_Click;

btnXoa.Click += BtnXoa\_Click;

btnThoat.Click += BtnThoat\_Click;

textBox3.ReadOnly = true;

this.StartPosition = FormStartPosition.CenterScreen;

}

private void Form1\_Load(object sender, EventArgs e)

{

}

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

{

if (!double.TryParse(textBox1.Text.Trim(), out a))

{

MessageBox.Show("Vui lòng nhập số hợp lệ cho số a!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

textBox1.Focus();

b = 0;

return false;

}

if (!double.TryParse(textBox2.Text.Trim(), out b))

{

MessageBox.Show("Vui lòng nhập số hợp lệ cho số b!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

textBox2.Focus();

return false;

}

return true;

}

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

{

if (LayGiaTriNhapVao(out double a, out double b))

{

double ketQua = a + b;

textBox3.Text = ketQua.ToString();

}

}

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

{

if (LayGiaTriNhapVao(out double a, out double b))

{

double ketQua = a - b;

textBox3.Text = ketQua.ToString();

}

}

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

{

if (LayGiaTriNhapVao(out double a, out double b))

{

double ketQua = a \* b;

textBox3.Text = ketQua.ToString();

}

}

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

{

if (LayGiaTriNhapVao(out double a, out double b))

{

if (b == 0)

{

MessageBox.Show("Không thể chia cho 0!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox2.Focus();

}

else

{

double ketQua = a / b;

textBox3.Text = ketQua.ToString();

}

}

}

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

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox1.Focus();

}

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

{

DialogResult result = MessageBox.Show("Bạn có muốn thoát khỏi chương trình?",

"Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

this.Close();

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

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:

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 WindowsForms\_1150080129\_HoangTrongDung\_BTbuoi5

{

public partial class apdung1 : Form

{

public apdung1()

{

InitializeComponent();

}

private void apdung1\_Load(object sender, EventArgs e)

{

rdoGCD.Checked = true;

btnCalculate.Click += btnCalculate\_Click;

btnExit.Click += btnExit\_Click;

txtNumberA.KeyPress += txtNumber\_KeyPress;

txtNumberB.KeyPress += txtNumber\_KeyPress;

}

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

{

try

{

if (string.IsNullOrWhiteSpace(txtNumberA.Text) || string.IsNullOrWhiteSpace(txtNumberB.Text))

{

MessageBox.Show("Vui lòng nhập đầy đủ hai số!", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (!int.TryParse(txtNumberA.Text.Trim(), out int numberA) || !int.TryParse(txtNumberB.Text.Trim(), out int numberB))

{

MessageBox.Show("Vui lòng nhập số nguyên hợp lệ!", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (numberA <= 0 || numberB <= 0)

{

MessageBox.Show("Vui lòng nhập số nguyên dương!", "Lỗi nhập liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (rdoGCD.Checked)

{

int gcd = CalculateGCD(numberA, numberB);

txtResult.Text = $"ƯSCLN({numberA}, {numberB}) = {gcd}";

}

else if (rdoLCM.Checked)

{

int lcm = CalculateLCM(numberA, numberB);

txtResult.Text = $"BSCNN({numberA}, {numberB}) = {lcm}";

}

else

{

MessageBox.Show("Vui lòng chọn tùy chọn tính toán!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show($"Có lỗi xảy ra: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

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

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

this.Close();

}

}

private void txtNumber\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar))

{

e.Handled = true;

}

}

private int CalculateGCD(int a, int b)

{

while (b != 0)

{

int temp = b;

b = a % b;

a = temp;

}

return a;

}

private int CalculateLCM(int a, int b)

{

return Math.Abs(a \* b) / CalculateGCD(a, b);

}

private void rdoLCM\_CheckedChanged(object sender, EventArgs e)

{

if (rdoLCM.Checked || rdoGCD.Checked)

{

txtResult.Clear();

}

}

private void txtResult\_TextChanged(object sender, EventArgs e)

{

}

private void grpOptions\_Enter(object sender, EventArgs e)

{

}

private void grpResult\_Enter(object sender, EventArgs e)

{

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void groupBox3\_Enter(object sender, EventArgs e)

{

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

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:

Nhóm thành viên Password

Phát triển công nghệ 1496 hoặc 2673

Nghiên cứu viên 7462

Thiết kế mô hình 8884 hoặc 3842 hoặc 3383

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.

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 WindowsForms\_1150080129\_HoangTrongDung\_BTbuoi5

{

public partial class apdung2 : Form

{

// Password dictionary for each group

private Dictionary<string, List<string>> groupPasswords;

private string currentInput = "";

public apdung2()

{

InitializeComponent();

InitializePasswords();

InitializeInterface();

}

private void InitializePasswords()

{

groupPasswords = 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" } }

};

}

private void InitializeInterface()

{

// Set up the password textbox

txtPassword.Text = "";

txtPassword.Font = new Font("Microsoft Sans Serif", 16F, FontStyle.Bold);

// Set up the access log ListView

lvAccessLog.View = View.Details;

lvAccessLog.FullRowSelect = true;

lvAccessLog.GridLines = true;

// Welcome message

this.Text = "Security Panel - Lab Access Control System";

}

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

{

Button btn = sender as Button;

if (btn != null && currentInput.Length < 10) // Limit input length

{

currentInput += btn.Text;

UpdatePasswordDisplay();

}

}

private void UpdatePasswordDisplay()

{

txtPassword.Text = new string('\*', currentInput.Length);

}

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

{

currentInput = "";

txtPassword.Text = "";

}

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

{

if (string.IsNullOrEmpty(currentInput))

{

MessageBox.Show("Vui lòng nhập mật khẩu!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

string groupName = ValidatePassword(currentInput);

DateTime accessTime = DateTime.Now;

if (!string.IsNullOrEmpty(groupName))

{

// Access granted

LogAccess(accessTime, groupName, "Chấp nhận");

ShowAccessResult(true, groupName);

}

else

{

// Access denied

LogAccess(accessTime, "", "Từ chối");

ShowAccessResult(false, "");

}

// Clear input after processing

currentInput = "";

txtPassword.Text = "";

}

private string ValidatePassword(string password)

{

foreach (var group in groupPasswords)

{

if (group.Value.Contains(password))

{

return group.Key;

}

}

return "";

}

private void LogAccess(DateTime dateTime, string groupName, string result)

{

ListViewItem item = new ListViewItem(dateTime.ToString("dd/MM/yyyy HH:mm:ss"));

item.SubItems.Add(groupName);

item.SubItems.Add(result);

// Color coding for results

if (result == "Chấp nhận")

{

item.BackColor = Color.LightGreen;

}

else

{

item.BackColor = Color.LightCoral;

}

lvAccessLog.Items.Insert(0, item); // Insert at top for newest first

// Keep only last 50 entries to prevent memory issues

while (lvAccessLog.Items.Count > 50)

{

lvAccessLog.Items.RemoveAt(lvAccessLog.Items.Count - 1);

}

}

private void ShowAccessResult(bool granted, string groupName)

{

if (granted)

{

MessageBox.Show($"TRUY CẬP ĐƯỢC CHẤP NHẬN!\n\nNhóm: {groupName}\nThời gian: {DateTime.Now:dd/MM/yyyy HH:mm:ss}\n\nChào mừng bạn đến phòng Lab!",

"Truy cập thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Flash the form green briefly

FlashForm(Color.LightGreen);

}

else

{

MessageBox.Show($"TRUY CẬP BỊ TỪ CHỐI!\n\nMật khẩu không đúng.\nThời gian: {DateTime.Now:dd/MM/yyyy HH:mm:ss}\n\nVui lòng liên hệ quản trị viên nếu cần hỗ trợ.",

"Truy cập thất bại", MessageBoxButtons.OK, MessageBoxIcon.Error);

// Flash the form red briefly

FlashForm(Color.LightCoral);

}

}

private async void FlashForm(Color color)

{

Color originalColor = this.BackColor;

this.BackColor = color;

await Task.Delay(500);

this.BackColor = originalColor;

}

private async void btnRing\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("CHUÔNG KHẨN CẤP!\n\nBạn có muốn gọi bảo vệ không?",

"Báo động", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation);

if (result == DialogResult.Yes)

{

DateTime ringTime = DateTime.Now;

LogAccess(ringTime, "CHUÔNG KHẨN CẤP", "Đã gọi bảo vệ");

MessageBox.Show("Đã thông báo cho bộ phận bảo vệ!\nThời gian: " + ringTime.ToString("dd/MM/yyyy HH:mm:ss"),

"Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// Flash red multiple times for emergency

for (int i = 0; i < 3; i++)

{

FlashForm(Color.Red);

await Task.Delay(300);

}

}

}

private void grpPassword\_Enter(object sender, EventArgs e)

{

// Keep existing functionality if needed

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

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.

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 WindowsForms\_1150080129\_HoangTrongDung\_BTbuoi5

{

public partial class bai2 : Form

{

// Định nghĩa giá dịch vụ

private const decimal GIA\_LAY\_CAO\_RANG = 50000;

private const decimal GIA\_TAY\_TRANG\_RANG = 100000;

private const decimal GIA\_HAN\_RANG = 100000;

private const decimal GIA\_NHO\_RANG = 10000;

private const decimal GIA\_BOC\_RANG = 1000000;

public bai2()

{

InitializeComponent();

InitializeForm();

}

private void InitializeForm()

{

// Khởi tạo các giá trị mặc định

nudHanRang.Minimum = 0;

nudHanRang.Maximum = 32;

nudNhoRang.Minimum = 0;

nudNhoRang.Maximum = 32;

nudBocRang.Minimum = 0;

nudBocRang.Maximum = 32;

// Đặt giá trị mặc định

nudHanRang.Value = 0;

nudNhoRang.Value = 0;

nudBocRang.Value = 0;

}

private void chkHanRang\_CheckedChanged(object sender, EventArgs e)

{

// Kích hoạt/vô hiệu hóa NumericUpDown tương ứng

nudHanRang.Enabled = chkHanRang.Checked;

if (!chkHanRang.Checked)

{

nudHanRang.Value = 0;

}

}

private void chkNhoRang\_CheckedChanged(object sender, EventArgs e)

{

// Kích hoạt/vô hiệu hóa NumericUpDown tương ứng

nudNhoRang.Enabled = chkNhoRang.Checked;

if (!chkNhoRang.Checked)

{

nudNhoRang.Value = 0;

}

}

private void chkBocRang\_CheckedChanged(object sender, EventArgs e)

{

// Kích hoạt/vô hiệu hóa NumericUpDown tương ứng

nudBocRang.Enabled = chkBocRang.Checked;

if (!chkBocRang.Checked)

{

nudBocRang.Value = 0;

}

}

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

{

try

{

// Kiểm tra tên khách hàng không được để trống

if (string.IsNullOrWhiteSpace(txtCustomerName.Text))

{

MessageBox.Show("Vui lòng nhập tên khách hàng!", "Cảnh báo",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtCustomerName.Focus();

return;

}

// Kiểm tra ít nhất một dịch vụ được chọn

if (!chkLayCaoRang.Checked && !chkTayTrangRang.Checked &&

!chkHanRang.Checked && !chkNhoRang.Checked && !chkBocRang.Checked)

{

MessageBox.Show("Vui lòng chọn ít nhất một dịch vụ!", "Cảnh báo",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

decimal tongTien = 0;

// Tính tiền các dịch vụ

if (chkLayCaoRang.Checked)

{

tongTien += GIA\_LAY\_CAO\_RANG;

}

if (chkTayTrangRang.Checked)

{

tongTien += GIA\_TAY\_TRANG\_RANG;

}

if (chkHanRang.Checked)

{

if (nudHanRang.Value == 0)

{

MessageBox.Show("Vui lòng nhập số răng cần hàn!", "Cảnh báo",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

nudHanRang.Focus();

return;

}

tongTien += GIA\_HAN\_RANG \* nudHanRang.Value;

}

if (chkNhoRang.Checked)

{

if (nudNhoRang.Value == 0)

{

MessageBox.Show("Vui lòng nhập số răng cần nhổ!", "Cảnh báo",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

nudNhoRang.Focus();

return;

}

tongTien += GIA\_NHO\_RANG \* nudNhoRang.Value;

}

if (chkBocRang.Checked)

{

if (nudBocRang.Value == 0)

{

MessageBox.Show("Vui lòng nhập số răng cần bọc!", "Cảnh báo",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

nudBocRang.Focus();

return;

}

tongTien += GIA\_BOC\_RANG \* nudBocRang.Value;

}

// Hiển thị kết quả

txtTongTien.Text = $"Tổng tiền: {tongTien:N0} đ";

// Hiển thị thông báo thành công

string thongBao = $"Khách hàng: {txtCustomerName.Text.Trim()}\n";

thongBao += "Dịch vụ đã chọn:\n";

if (chkLayCaoRang.Checked)

thongBao += $"- Lấy cao răng: {GIA\_LAY\_CAO\_RANG:N0} đ\n";

if (chkTayTrangRang.Checked)

thongBao += $"- Tẩy trắng răng: {GIA\_TAY\_TRANG\_RANG:N0} đ\n";

if (chkHanRang.Checked)

thongBao += $"- Hàn răng ({nudHanRang.Value} răng): {GIA\_HAN\_RANG \* nudHanRang.Value:N0} đ\n";

if (chkNhoRang.Checked)

thongBao += $"- Nhổ răng ({nudNhoRang.Value} răng): {GIA\_NHO\_RANG \* nudNhoRang.Value:N0} đ\n";

if (chkBocRang.Checked)

thongBao += $"- Bọc răng ({nudBocRang.Value} răng): {GIA\_BOC\_RANG \* nudBocRang.Value:N0} đ\n";

thongBao += $"\nTổng cần thanh toán: {tongTien:N0} đ";

MessageBox.Show(thongBao, "Thông tin thanh toán",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Đã xảy ra lỗi: {ex.Message}", "Lỗi",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

DialogResult result = MessageBox.Show("Bạn có muốn thoát khỏi chương trình?",

"Xác nhận thoát",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

this.Close();

}

}

}

}

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

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.

using System;

using System.Windows.Forms;

using System.Text;

namespace WindowsForms\_1150080129\_HoangTrongDung\_BTbuoi5

{

public partial class apdung3 : Form

{

public apdung3()

{

InitializeComponent();

this.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

}

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

{

if (string.IsNullOrWhiteSpace(txtUsername.Text))

{

MessageBox.Show("Vui lòng nhập tên đăng nhập!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtUsername.Focus();

return;

}

if (string.IsNullOrWhiteSpace(txtPassword.Text))

{

MessageBox.Show("Vui lòng nhập mật khẩu!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtPassword.Focus();

return;

}

MessageBox.Show($"Đăng nhập thành công!\nTên đăng nhập: {txtUsername.Text}", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

txtUsername.Clear();

txtPassword.Clear();

txtUsername.Focus();

}

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

{

this.Close();

}

private void apdung3\_Load(object sender, EventArgs e)

{

txtUsername.Focus();

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Thực hành 3: Thiết kế form frmXuLySo với control ListBox

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.

A screenshot of a computer

AI-generated content may be incorrect.

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 WindowsForms\_1150080129\_HoangTrongDung\_BTbuoi5

{

public partial class bai3 : Form

{

public bai3()

{

InitializeComponent();

// Set focus to textbox when form loads

this.Load += (s, e) => txtNumberInput.Focus();

}

// Event handler for KeyPress - only allow digits, backspace, and negative sign

private void txtNumberInput\_KeyPress(object sender, KeyPressEventArgs e)

{

// Allow digits, backspace, and negative sign (only at the beginning)

if (!char.IsDigit(e.KeyChar) && e.KeyChar != '\b')

{

// Allow negative sign only at the beginning and if there's no text or cursor is at position 0

if (e.KeyChar == '-')

{

TextBox txt = sender as TextBox;

if (txt.SelectionStart != 0 || txt.Text.Contains("-"))

{

e.Handled = true; // Block the character

}

}

else

{

e.Handled = true; // Block non-numeric characters

}

}

}

// Event handler for KeyDown - handle Enter key

private void txtNumberInput\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

AddNumberToList();

e.Handled = true; // Prevent the beep sound

}

}

// Event handler for Add Number button

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

{

AddNumberToList();

}

// Method to add number to the list

private void AddNumberToList()

{

if (!string.IsNullOrWhiteSpace(txtNumberInput.Text))

{

if (int.TryParse(txtNumberInput.Text, out int number))

{

lstNumbers.Items.Add(number);

txtNumberInput.Clear();

txtNumberInput.Focus();

}

else

{

MessageBox.Show("Vui lòng nhập một số nguyên hợp lệ!", "Lỗi",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtNumberInput.SelectAll();

txtNumberInput.Focus();

}

}

}

// Increase each element by 2

private void btnIncreaseBy2\_Click(object sender, EventArgs e)

{

if (lstNumbers.Items.Count == 0)

{

MessageBox.Show("Danh sách rỗng!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

for (int i = 0; i < lstNumbers.Items.Count; i++)

{

int currentValue = (int)lstNumbers.Items[i];

lstNumbers.Items[i] = currentValue + 2;

}

}

// Select first even number

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

{

for (int i = 0; i < lstNumbers.Items.Count; i++)

{

int number = (int)lstNumbers.Items[i];

if (number % 2 == 0)

{

lstNumbers.SelectedIndex = i;

return;

}

}

MessageBox.Show("Không tìm thấy số chẵn nào trong danh sách!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

// Select last odd number

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

{

for (int i = lstNumbers.Items.Count - 1; i >= 0; i--)

{

int number = (int)lstNumbers.Items[i];

if (number % 2 != 0)

{

lstNumbers.SelectedIndex = i;

return;

}

}

MessageBox.Show("Không tìm thấy số lẻ nào trong danh sách!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

// Delete selected item

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

{

if (lstNumbers.SelectedIndex >= 0)

{

lstNumbers.Items.RemoveAt(lstNumbers.SelectedIndex);

}

else

{

MessageBox.Show("Vui lòng chọn một phần tử để xóa!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

// Delete first item

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

{

if (lstNumbers.Items.Count > 0)

{

lstNumbers.Items.RemoveAt(0);

}

else

{

MessageBox.Show("Danh sách rỗng!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

// Delete last item

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

{

if (lstNumbers.Items.Count > 0)

{

lstNumbers.Items.RemoveAt(lstNumbers.Items.Count - 1);

}

else

{

MessageBox.Show("Danh sách rỗng!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

// Clear all items

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

{

if (lstNumbers.Items.Count > 0)

{

DialogResult result = MessageBox.Show("Bạn có chắc chắn muốn xóa tất cả các số?",

"Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

lstNumbers.Items.Clear();

}

}

else

{

MessageBox.Show("Danh sách đã rỗng!", "Thông báo",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

// Exit application

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

{

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

"Xác nhận thoát", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

Application.Exit();

}

}

}

}

Áp dụng 4: Sử dụng Form, TextBox, ComboBox và Button

Thiết kế Form gồm có 1 TextBox điền Họ tên, 3 ComboBox có các Label đi kèm có nhãn: Ngày, Tháng, Năm

* ComboBox Ngày nạp các giá trị từ 1 đến 31 (không cho nhập thêm dữ liệu)
* ComboBox Tháng nạp các giá trị từ 1 đến 12 (không cho phép nhập thêm dữ liệu)
* ComboBox Năm nạp các giá trị từ 1930 đến 2010 (cho phép nhập thêm dữ liệu)

Khi ấn nút lệnh Kiểm tra, nếu ngày tháng năm sinh là hợp lệ sẽ hiện ra MessageBox thông báo Họ tên và ngày sinh. Nếu dữ liệu ngày tháng không hợp lệ sẽ hiện ra MessageBox thông báo “Ngày sinh không hợp lệ!”

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

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 WindowsForms\_1150080129\_HoangTrongDung\_BTbuoi5

{

public partial class apdung4 : Form

{

public apdung4()

{

InitializeComponent();

LoadComboBoxData();

}

private void LoadComboBoxData()

{

for (int i = 1; i <= 31; i++)

{

cmbNgay.Items.Add(i.ToString());

}

for (int i = 1; i <= 12; i++)

{

cmbThang.Items.Add(i.ToString());

}

for (int i = 1930; i <= 2010; i++)

{

cmbNam.Items.Add(i.ToString());

}

}

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

{

if (string.IsNullOrWhiteSpace(txtHoTen.Text) ||

cmbNgay.SelectedItem == null ||

cmbThang.SelectedItem == null ||

string.IsNullOrWhiteSpace(cmbNam.Text))

{

MessageBox.Show("Vui lòng điền đầy đủ thông tin!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

try

{

int ngay = int.Parse(cmbNgay.SelectedItem.ToString());

int thang = int.Parse(cmbThang.SelectedItem.ToString());

int nam = int.Parse(cmbNam.Text);

DateTime ngaySinh = new DateTime(nam, thang, ngay);

string message = $"Họ tên: {txtHoTen.Text}\nNgày sinh: {ngay:D2}/{thang:D2}/{nam}";

MessageBox.Show(message, "Thông tin", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception)

{

MessageBox.Show("Ngày sinh không hợp lệ!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}