**Mahlet Gebremedhin**

**Question 1**

using System;

class TriangleAreaCalculator

{

static void Main()

{

Console.WriteLine("Welcome to P-Thag's Triangle Area Calculator!");

// Prompt user for base input

Console.Write("Enter the base of the triangle: ");

double baseSize = Convert.ToDouble(Console.ReadLine());

// Prompt user for height input

Console.Write("Enter the height of the triangle: ");

double height = Convert.ToDouble(Console.ReadLine());

// Compute the area using the formula: (b \* h) / 2

double area = (baseSize \* height) / 2;

// Display the result

Console.WriteLine($"\nThe area of the triangle is: {area} square units.");

Console.WriteLine("\nThank you for helping P-Thag!");

}

}

THE OUTPUT

Welcome to P-Thag's Triangle Area Calculator!

Enter the base of the triangle: 8

Enter the height of the triangle: 6

The area of the triangle is: 24 square units.

Thank you for helping P-Thag!

**Question 2**

using System;

class ChocolateEggsDivider

{

static void Main()

{

Console.WriteLine("Welcome to the Chocolate Egg Divider!");

// Asking for the total number of chocolate eggs collected

Console.Write("Enter the number of chocolate eggs collected today: ");

int totalEggs = Convert.ToInt32(Console.ReadLine());

// There are 4 sisters

int numberOfSisters = 4;

// Calculating eggs per sister

int eggsPerSister = totalEggs / numberOfSisters;

// Calculating leftover eggs for the duckbear

int duckbearEggs = totalEggs % numberOfSisters;

// Displaying results in a very beginner-friendly way

Console.WriteLine("\nCalculating the distribution...");

Console.WriteLine("Each sister gets: " + eggsPerSister + " eggs.");

Console.WriteLine("The duckbear gets the leftover: " + duckbearEggs + " eggs.");

Console.WriteLine("\nThank you for using the Chocolate Egg Divider!");

// Finding three total egg counts where the duckbear gets more than a sister

Console.WriteLine("\nFinding three cases where the duckbear gets more eggs than each sister...");

for (int testEggs = 1; testEggs <= 100; testEggs++) // Brute force checking up to 100 eggs

{

int testSisterEggs = testEggs / numberOfSisters;

int testDuckbearEggs = testEggs % numberOfSisters;

if (testDuckbearEggs > testSisterEggs)

{

Console.WriteLine($"Case Found! Total Eggs: {testEggs} | Sisters get: {testSisterEggs} | Duckbear gets: {testDuckbearEggs}");

}

}

}

}

**Question 3**

using System;

class KingdomPointsCalculator

{

static void Main()

{

Console.WriteLine("Welcome, Kings! Let's determine who has the greatest kingdom.");

// Asking for province, duchy, and estate counts

Console.Write("Enter the number of provinces you have: ");

int provinces = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the number of duchies you have: ");

int duchies = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the number of estates you have: ");

int estates = Convert.ToInt32(Console.ReadLine());

// Calculating the total score

int totalPoints = (provinces \* 6) + (duchies \* 3) + (estates \* 1);

// Displaying the result

Console.WriteLine($"\nYour total kingdom score is: {totalPoints} points!");

Console.WriteLine("\nMay the greatest king reign supreme!");

}

}

**Question 4**

using System;

class DefenseOfConsolas

{

static void Main()

{

// Change the console title

Console.Title = "Defense of Consolas";

Console.WriteLine("The City of Consolas is under attack!");

Console.WriteLine("We need to deploy defenses around the target block.\n");

// Ask for the target row and column

Console.Write("Target Row? ");

int targetRow = Convert.ToInt32(Console.ReadLine());

Console.Write("Target Column? ");

int targetColumn = Convert.ToInt32(Console.ReadLine());

// Compute deployment positions

int leftRow = targetRow;

int leftCol = targetColumn - 1;

int rightRow = targetRow;

int rightCol = targetColumn + 1;

int topRow = targetRow - 1;

int topCol = targetColumn;

int bottomRow = targetRow + 1;

int bottomCol = targetColumn;

// Change text color for deployment instructions

Console.ForegroundColor = ConsoleColor.Green;

// Display the deployment locations

Console.WriteLine("\nDeploy to:");

Console.WriteLine($"({leftRow}, {leftCol})");

Console.WriteLine($"({topRow}, {topCol})");

Console.WriteLine($"({rightRow}, {rightCol})");

Console.WriteLine($"({bottomRow}, {bottomCol})");

// Reset text color to default

Console.ResetColor();

// Play a beep sound to indicate completion

Console.Beep();

Console.WriteLine("\nThe city is defended! Stand by for further orders.");

}

}

**Question 5**

using System;

class ClocktowerRepair

{

static void Main()

{

Console.WriteLine("Repairing the Clocktower of Consolas!");

// Ask the user for a number

Console.Write("Enter a number to keep the clock running: ");

int number = Convert.ToInt32(Console.ReadLine());

// Determine if the number is even or odd using the modulus operator (%)

if (number % 2 == 0)

{

Console.WriteLine("Tick"); // Even number → Tick

}

else

{

Console.WriteLine("Tock"); // Odd number → Tock

}

Console.WriteLine("\nThe Clocktower is back in motion!");

}

}