**Test 1**

**Problem 1: Output** **[10 min, 15 pts]**

What is the output of this program?

static void Main(string[] args)

{

 int x = 0, y = 0, z;

 for (z = 3; z > 0; z = z - 2)

 {

 y = z \* 2;

 Console.WriteLine("y = {0}", y);

 x++;

 }

 Console.WriteLine("y = {0}, z = {1}", y, z);

 x = x - 2;

 while (x != 0)

 {

 x = x + y;

 y = y + x;

 z++;

 Console.WriteLine("x = {0}, y = {1}, z = {2}", x,y,z);

 }

 Console.WriteLine("\nAfter the loop:");

 Console.WriteLine("x = {0}, y = {1}, z = {2}", x, y, z);

 x++;

 while (x != 0)

 {

 x--;

 y = y + x;

 z++;

 Console.WriteLine("x = {0}, y = {1}, z = {2}", x,y,z);

 }

 Console.WriteLine("\nAfter the 2nd loop:");

 Console.WriteLine("x = {0}, y = {1}, z = {2}", x, y, z);

 Console.ReadKey();

}

**Problem 2: Classwork [10 min, 20 pts]**

Write a program that reads from you a number in grams (g) and then converts it to kilograms (kg), displaying messages whether the entered number is: very heavy, heavy, medium, light, very light. (It is up to you to specify the range of heavy and light weights).

**Problem 3: Pattern [25 min, 15 pts]**

Use the following loop to produce the output below. You are **not allowed** to change the bounds of the loop or add new variables; just use the loop as is without any change. You can use constants in your program.

The loop is:

for(int i=20; i>10; i--)

The output is:

40 , -9 , 42 , -8 , 44 , -7 , 46 , -6 , 48 , -5 ,

**Problem 4: Drawing Stars [25 min, 25 pts]**

In this program, you are going to draw stars. Every Console.Write() statement can be one of the following: Console.Write("\*"), Console.Write(" "), or Console.Write("\n"). Your program should be able to produce drawings for any entered number and not only for the numbers shown in the examples below.

**Part (A):** Draw the following for **odd** numbers only:

|  |  |  |
| --- | --- | --- |
| n = 5\*\*\*\*\*\*\*\*\* | n = 7\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | n = 9\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

**Part B:** Draw the following for **odd** numbers only:

|  |  |  |  |
| --- | --- | --- | --- |
| n = 5\*\*\*\*\*\* \* \*\*\*\*\*\*\* \* \*\*\*\*\*\* | n = 7\*\*\*\*\*\*\*\* \* \*\* \* \*\*\*\*\*\*\*\*\* \* \*\* \* \*\*\*\*\*\*\*\* | n = 9\*\*\*\*\*\*\*\*\*\* \* \*\* \* \*\* \* \*\*\*\*\*\*\*\*\*\*\* \* \*\* \* \*\* \* \*\*\*\*\*\*\*\*\*\* | n = 11\*\*\*\*\*\*\*\*\*\*\*\* \* \*\* \* \*\* \* \*\* \* \*\*\*\*\*\*\*\*\*\*\*\*\* \* \*\* \* \*\* \* \*\* \* \*\*\*\*\*\*\*\*\*\*\*\* |

**Problem 5: Prime Numbers [30 min, 25 pts]**

A *Perfect-Prime* number is a **prime numbe**r whose **sum of its digits is even**. Write a program that reads a number from the user and displays the first 3 *Perfect-Prime* numbers after it. An example of the output is shown below:

Please enter a number: 4

The first 3 Perfect-Prime numbers are:

11 , 13 , 17 ,