EXAMEN PROGRAMACION 1RA EV

1. (0,5) Crea una funcion que reciba como parametro 2 enteros y devuelva un real que sea el primer parametro mas 1, todo ello entre el segundo.

```
public static double Funcion1(int a, int b)
{
   return (a + 1) / (double)b;
}
```

2. (0,5) Crea una funcion que se le pasen 5 reales (a, b, c, d y e) mas otro real que se llama x. La funcion devuelve un real representado por el siguiente polinomio:

$a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x^1 + e$

```
public static double Funcion2(double a, double b, double c, double d, double e, double x)
{
   return a * Math.Pow(x, 4) + b * Math.Pow(x, 3) + c * Math.Pow(x, 2) + d * x + e;
}
```

3. (1) Crea una funcion que se le pasen 10 enteros y devuelva el menor de ellos.

```
public static int Funcion3(int a, int b, int c, int d, int e, int f, int g, int h, int i, int j)
{
   int min = a;
   if (b < min)
      min = b;
   if (c < min)
      min = c;
   if (d < min)
      min = d;
   if (e < min)
      min = e;
   if (f < min)
      min = f;
   if (g < min)
      min = g;
   if (h < min)
      min = h;
   if (i < min)
      min = i;
   if (j < min)
      min = j;

   return min;
}
```

```
public static int EncuentraMinimo(int[] numeros)
{
    int minimo = numeros[0];
    for (int i = 1; i < numeros.Length; i++)
    {
        if (numeros[i] < minimo)
        {
            minimo = numeros[i];
        }
    }
    return minimo;
}
```

4. (1) Crea una funcion que se le pasen 3 enteros (no ordenados), y devuelva la distancia maxima entre el numero que tiene el valor central, con respecto al minimo y al maximo de los otros dos. Por ejemplo, si se le pasa (3, 1, 7) devolvera un 4, ya que la distancia de 3 y 1 es 2 y la distancia de 3 y 7 es 4.

5. (0,5) Crea una funcion que devuelva el numero de digitos de un numero entero. Para hacer esta funcion no se pueden usar strings.
Por ejemplo, el numero de digitos del 154 es 3, el de 75366 es 5. Para ello, puedes ir dividiendo el numero por 10 hasta que…

```
public static int Funcion5(int num)
{
    int digitos = 1;

    while (num / 10 > 0)
    {
        num /= 10;
        digitos++;
    }

    return digitos;
}
```

6. (1) Realiza una funcion recursiva que, dado un numero n, devuelva:
$1^2 + 2^2 + 3^2 + 4^2 + 5^2 \ldots n^2$

Consejo, recuerda que cuando hicimos el sumatorio o el productorio, la funcion recursiva devolvia 5 + 4 + 3 + 2 + 1 en vez de 1+2+3+4+5

```
public static int CalculaSumaDeCuadrados(int n)
{
    if (n == 1)
    {
        return 1;
    }
    else
    {
```

```
        return n * n + CalculaSumaDeCuadrados(n - 1);
    }
}
```

7. (1) Crea una funcion que reciba como parametro un texto, y devuelva la posicion de la primera vocal que encuentre (sea mayuscula o minuscula) y de la ultima vocal que encuentre.

```
public static (int, int) EncuentraPosicionesDeVocales(string texto)
{
    int primeraVocal = -1;
    int ultimaVocal = -1;
    string vocales = "aeiouAEIOU";
    for (int i = 0; i < texto.Length; i++)
    {
        if (vocales.Contains(texto[i]))
        {
            if (primeraVocal == -1)
            {
                primeraVocal = i;
            }
            ultimaVocal = i;
        }
    }
    return (primeraVocal, ultimaVocal);
}
```

8. (1) Crea una funcion que reciba como parametro un numero y devuelva la posicion de ese numero dentro de los numeros primos. Si el numero no es primo, devolvera un -1.
Por ejemplo, sabemos que los numeros primos son: 2, 3, 5, 7, 11, 13, … Si a esta funcion se le pasa un 7 devolvera un 4, ya que 7 es el cuarto numero primo. Si a esta funcion se le pasa un 6, devolvera un -1, ya que este no es primo.

```
public static int EncuentraPosicionDePrimo(int numero)
{
    if (numero < 2)
    {
        return -1;
    }
    int posicion = 1;
    int contador = 2;
    while (contador <= numero)
    {
        bool esPrimo = true;
        for (int i = 2; i <= Math.Sqrt(contador); i++)
        {
            if (contador % i == 0)
            {
                esPrimo = false;
                break;
            }
```

9. (1) Haz una funcion que se le pasen como parametros un enum y un booleano. El enum es el estado de una maquina (PREPARADA, PROCESANDO, EJECUTANDO, TERMINANDO). Si el booleano que se le pasa es false, devolvera justamente el enum que se le pasa como parametro, si es true, devolvera el siguiente estado de la maquina.
Por ejemplo, si se le pasa PROCESANDO y un true, devolvera EJECUTANDO. Si se le pasa TERMINANDO y un false, devolvera TERMINANDO.

```
public enum MaquinaEstado
{
   PREPARADA,
   PROCESANDO,
   EJECUTANDO,
   TERMINANDO
}

public static MaquinaEstado EstadoSiguiente(MaquinaEstado estadoActual, bool siguiente)
{
   int valorActual = (int)estadoActual;

   if (siguiente)
   {
      valorActual++;
      if (valorActual > 3)
         valorActual = 0;
   }

   return (MaquinaEstado)valorActual;
}

MaquinaEstado estadoActual = MaquinaEstado.PROCESANDO;
bool siguiente = true;

MaquinaEstado estadoSiguiente = EstadoSiguiente(estadoActual, siguiente);

Console.WriteLine(estadoSiguiente); // Salida: EJECUTANDO
```

10. (1) Crea una funcion que se le pase un numero y devuelva la suma de todos sus divisores desde 1 hasta ese numero (sin contar ese numero)

```
public static int SumaDivisores(int numero)
{
   int suma = 0;

   for (int i = 1; i < numero; i++)
   {
      if (numero % i == 0)
         suma += i;
   }

   return suma;
}
```

```
int numero = 12;

int suma = SumaDivisores(numero);

Console.WriteLine(suma); // Salida: 16 (ya que los divisores de 12 son 1, 2, 3, 4 y 6, y su suma es 16)
```

11. (1.5) Crea una funcion que reciba como parametros los siguientes datos:
a. Los coeficientes a, b, c, d, y e de un polinomio del tipo
$a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x^1 + e$
b. Un intervalo de muestreo xmin, xmax
c. Un periodo de muestreo de Lambda x = 0,1
d. La funcion debe devolver el valor maximo que toma ese polinomio en el intervalo dado y dado el tamaño del sampler.

```
public static double ValorMaximoPolinomio(double a, double b, double c, double d, double e, double xmin, double xmax, double lambda)
{
    double valorMaximo = double.MinValue;

    for (double x = xmin; x <= xmax; x += lambda)
    {
        double y = a * Math.Pow(x, 4) + b * Math.Pow(x, 3) + c * Math.Pow(x, 2) + d * x + e;

        if (y > valorMaximo)
            valorMaximo = y;
    }

    return valorMaximo;
}

double a = 1;
double b = -2;
double c = 3;
double d = -4;
double e = 5;
double xmin = -1;
double xmax = 1;
double lambda = 0.1;

double valorMaximo = ValorMaximoPolinomio(a, b, c, d, e, xmin, xmax, lambda);

Console.WriteLine(valorMaximo); // Salida: 5 (ya que el valor máximo del polinomio en el intervalo [-1, 1] es 5, que se alcanza en x = 0)
```