



Tecnicatura Superior en Programación

Apellido y Nombre:

Fecha:

NOTA: Para aprobar el examen deberá cumplir 2 condiciones necesarias, tener al menos un mínimo de 20 puntos en cada una de las partes A y B que componen el parcial y una nota total igual o mayor a 60%.

PARTE A

INTRODUCCIÓN A C# (10%)

Crear un método que permita el ingreso de una cadena alfanumerica, al finalizar el ingreso recorra cada uno de los caracteres de la cadena si es un numero indicar que numero se encontró, en qué posición y si es par o impar.

Cadena: sdf4ajhay9gygejhkekj56ikjkjokjkjukjkja

Resultado:

Se encontró el número 4 en la posición 4, el número es par.

Se encontró el número 9 en la posición 10, el número es impar.

Se encontró el número 5 en la posición 20, el número es impar.

Se encontró el número 6 en la posición 21, el número es par.

METODOS (10%)

Codifique un método **recursivo** denominado **operacionRecursiva (int nro)** que mediante recursión permita sumar los dígitos de un número.

Ej.: Entrada = 123

Resultado = 6 => 1+2+3

ARRAY (15%)

Cree un programa que permita ingresar por teclado 1 valor entero mayor a cero. Este primer valor será el tamaño de una matriz de nxn, posteriormente solicite x valores distintos (letras o numeros) igual al tamaño de n, es decir si n es igual a 5 se deben solicitar 5 letras o números. Almacene los valores en alguna variable que lo permita.

Valide que el número sea mayor a cero y que los valores ingresados sean distintos. Mediante el valor ingresado genere una matriz de orden n x n asignando a cada celda uno de los valores ingresados, buscando como resultado la creación de un cuadro latino.

Un **cuadrado latino** es una matriz de $n \times n$ elementos en la que cada casilla está ocupada por uno de los n símbolos de tal modo que cada uno de ellos aparece exactamente una vez en cada columna y en cada fila.

Ejemplo:

Valore Entero

Valor 5 -> la matriz será de 5 x 5

Valores ingresado

Y, 1, Z, 3, X

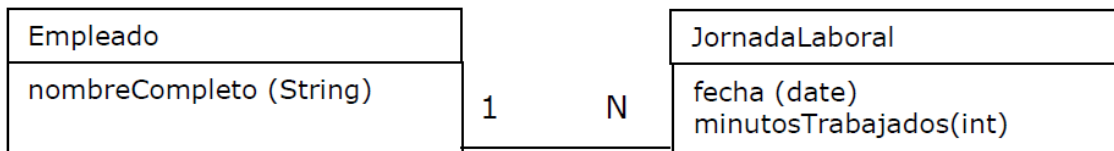


Resultado:

Y	1	Z	3	X
X	Y	1	Z	3
3	X	Y	1	Z
Z	3	X	Y	1
1	Z	3	X	Y

OBJETOS (15%)

Codifique el siguiente modelo de Clases.



Codifique en la clase Empleado del modelo anterior un método llamado **totalTrabajadoMes**(int mes, int año) que retorne el total de minutos trabajados para un mes y año específico. Muestre el resultante en minutos y en formato horas:minutos (HH:MM)

PARTE B

ACCESO A DATOS

LINQ

Cree la clase Cliente

```
class Cliente
{
    private long id;
    private string nombre;
    private string apellido;
    private double saldo;
    private string domicilio;
    private long telefono;
    private long cuit;

    public long Id { get => id; set => id = value; }
    public string Nombre { get => nombre; set => nombre = value; }
    public string Apellido { get => apellido; set => apellido = value; }
    public double Saldo { get => saldo; set => saldo = value; }
    public string Domicilio { get => domicilio; set => domicilio = value; }
    public long Telefono { get => telefono; set => telefono = value; }
    public long Cuit { get => cuit; set => cuit = value; }
}
```



EXAMEN FINAL
LAB 3
TSP-UTN-FRM

Cree una clase Programa con el método main y agregue el siguiente código dentro del main

```
ICollection<Cliente> clientes = new List<Cliente>() {  
    new Cliente() { Id = 1, Nombre = "Juan", Apellido = "Perez", Domicilio =  
        "Salta 314", Telefono = 154356989, Saldo = 20000, Cuit = 20291126587 } ,  
    new Cliente() { Id = 2, Nombre = "Pedro", Apellido = "Hernandez", Domicilio =  
        "Chaco 456", Telefono = 151454789, Saldo = 36000, Cuit = 20244126587 } ,  
    new Cliente() { Id = 3, Nombre = "Jose", Apellido = "Chatruc", Domicilio =  
        "Lavalle 789", Telefono = 154352589, Saldo = 28000, Cuit = 20291336587 } ,  
    new Cliente() { Id = 4, Nombre = "Carlos", Apellido = "Alonso", Domicilio =  
        "Rioja 14", Telefono = 154354789, Saldo = 45000, Cuit = 20297726587 } ,  
    new Cliente() { Id = 5, Nombre = "Claudio", Apellido = "Ahumada", Domicilio =  
        "Lima 125", Telefono = 154964789, Saldo = 28000, Cuit = 20291125587 } ,  
    new Cliente() { Id = 5, Nombre = "Sebastian", Apellido = "Tobar", Domicilio =  
        "Rawson 123", Telefono = 154357489, Saldo = 22000, Cuit = 20291158447 } ,  
    new Cliente() { Id = 5, Nombre = "Javier", Apellido = "Puebla", Domicilio =  
        "Italia 987", Telefono = 154358589, Saldo = 33000, Cuit = 20301126587 } ,  
    new Cliente() { Id = 5, Nombre = "Fabian", Apellido = "Gilar", Domicilio =  
        "Las Viñas 987", Telefono = 154554789, Saldo = 38000, Cuit = 20311126587 } ,  
    new Cliente() { Id = 5, Nombre = "Victor", Apellido = "Pereira", Domicilio =  
        "25 de Mayo 654", Telefono = 154356589, Saldo = 37000, Cuit = 20321126587 } ,  
    new Cliente() { Id = 5, Nombre = "Nelson", Apellido = "Piquet", Domicilio =  
        "Peru 987", Telefono = 154454789, Saldo = 26000, Cuit = 20331126587 }  
};
```

A continuación del código anterior codifique Mediante LINQ y muestre por pantallas:

- Seleccione todos los Clientes cuyo saldo sea mayor a 30000 ordenados por saldo ascendente. **(5%)**
- Seleccione todos los Clientes cuyo apellido inicien con p ordenados por Nombre. **(5%)**
- Seleccione el campo Saldo y realice las siguientes tareas: **(10%)**
Cuenta la cantidad de saldos. (Mediante clausula Count).
Indique el cliente con el máximo saldo. (Mediante clausula Max).
Indique el promedio de los saldos. (Mediante clausula Average).



ASP.NET

Codifique el siguiente Formulario. Tipografía Verdana, Tamaño tipografía 12px. Use las etiquetas de ASP.NET que correspondan. **(5%)**

Apellido	<input type="text"/>
Nombre	<input type="text"/>
Sexo	<input type="radio"/> Masculino <input type="radio"/> Femenino
Nacionalidad	Argentina ▼
Bloqueado	<input type="checkbox"/>
Edad	<input type="text"/>
Antecedentes	<input type="text"/>

Mediante el uso de C# al hacer click sobre un botón, capture la información del formulario de la tabla superior y muestre los datos cargados en una segunda tabla inferior. Respete el diseño básico de la pantalla. **(10%)**

XML

Cree una página EscribirXML.aspx y Realice la escritura del siguiente XML **(15%)**

```
<result>
  <permisos>
    <permiso tipo="Comision Diaria">
      <id>229</id>
      <sector>Forestales</sector>
      <consejo>NO POSEE</consejo>
    </permiso>
    <permiso tipo="Razones Particulares">
      <id>381</id>
      <sector>Alumbrado Público - Mantenimiento</sector>
      <consejo>NO POSEE</consejo>
    </permiso>
  </permisos>
</result>
```