



UNCUYO  
UNIVERSIDAD  
NACIONAL DE CUYO



FACULTAD  
DE INGENIERÍA

UNIVERSIDAD NACIONAL DE CUYO

FACULTAD DE INGENIERÍA

ALGORITMOS Y ESTRUCTURA DE DATOS I

Primer Año

Licenciatura en Ciencias de la Computación



## Trabajo Práctico Nº 1

### **Parte I: Arreglos**

#### **Ejercicios obligatorios**

1. Elaborar un algoritmo que lea un vector, busque el mayor elemento en valor absoluto y muestre el resultado.
2. Elaborar un algoritmo que lea dos vectores, verifique si tienen la misma dimensión y los sume en un nuevo vector. Calcule la norma cuadrática de este último vector. Muestre el vector resultado y su norma cuadrática.
3. Elaborar un algoritmo que lea una matriz y un vector, y que verifique si es posible la multiplicación. En el caso de ser posible realice la operación correspondiente, caso contrario, que muestre el mensaje “dimensiones incorrectas”.
4. Elaborar un algoritmo que lea dos matrices, calcule la diferencia de las mismas y almacene el resultado en una tercera matriz.
5. Elaborar un algoritmo que lea una matriz y determine si es triangular superior. En caso afirmativo el algoritmo debe calcular el determinante de dicha matriz.

#### **Ejercicios propuestos**

6. Elaborar un algoritmo que lea dos vectores, verifique si tienen la misma dimensión y obtenga el producto escalar de los mismos. Muestre el resultado.
7. Elaborar un algoritmo que lea dos matrices, calcule su producto y almacene el resultado en una tercera matriz. Verifique si esta última matriz es estrictamente diagonal dominante por filas.
8. Elaborar un algoritmo que lea una matriz y determine si es triangular inferior. En caso afirmativo el algoritmo debe calcular la matriz transpuesta de la misma.



## **Parte II: TAD Conjuntos**

**Todas las implementaciones son obligatorias**

1. A partir del TAD **Set (Conjunto)** implementar las siguientes operaciones utilizando la estructura **Array**:

### **Create\_Set(Array):**

**Descripción:** Crea un TAD de tipo **Set** a partir de un arreglo recibido como parámetro.

**Entrada:** el Arreglo sobre el cual se quiere construir el TAD **Set**

**Salida:** Un **Array** que representa el **TAD Set**

### **Union(Array S, Array T):**

**Descripción:** Aplica la operación **UNIÓN** sobre los conjuntos (Sets) **S** y **T**.

**Precondición:** La operación debe garantizar que no hay elementos duplicados en los arreglos. (Ver Nota más abajo)

**Entrada:** Dos arreglos que representan los **Sets S** y **T**

**Salida:** Un **Array** que representa un nuevo **TAD Set**

### **Intersection(Array S, Array T):**

**Descripción:** Aplica la operación **INTERSECCIÓN** sobre los conjuntos **S** y **T**.

**Precondición:** La operación debe garantizar que no hay elementos duplicados en los arreglos. (Ver Nota más abajo)

**Entrada:** Dos arreglos que representan los **Sets S** y **T**

**Salida:** Un **Array** que representa un nuevo **TAD Set**

### **Difference(Array S, Array T):**

**Descripción:** Aplica la operación **DIFERENCIA** sobre los conjuntos **S** y **T**.

**Precondición:** La operación debe garantizar que no hay elementos duplicados en los arreglos. (Ver Nota más abajo)

**Entrada:** Dos arreglos que representan los **Sets S** y **T**

**Salida:** Un **Array** que representa un nuevo **TAD Set**

2. Todas las operaciones del TAD conjunto deberán incluirse en un archivo (modulo) **set.py**

**Nota:** Las operaciones sobre **Set**, asumen que los Arrays **T** y **S** no contienen elementos duplicados. Pero puede ocurrir que por error se pase un array con elementos duplicados. Entonces, deberían implementar una función **check\_duplicates(Array)** la cual verifique si hay o no duplicados en los Array pasados como parámetros. En caso de que haya duplicados, debería salir con algún error.