

---

# Algoritmos y Estructura de Datos 2019 Curso de Verano

---

Director de cátedra: Dr.  
Oscar Bruno

---

Ing.Roxana L.Leituz

---

- 80) Dada una pila y un valor **I**, desarrollar un procedimiento que elimine los 2 primeros nodos de la pila y deje el valor **I** como primero. (Definir parametros y codificar).
- 81) Dada una pila y un valor **I**, desarrollar un procedimiento que inserte **I** como tercer valor de la pila. (Definir parametros y codificar).
- 82) Idem ejercicio 60 pero retornando un parametro con valor '**S**' o '**N**' segun haya sido exitoso o no el requerimiento. (Definir parametros y codificar).
- 83) Dada una pila y dos valores **X** e **I**, desarrollar un procedimiento que inserte el valor **X** en la posicion **I** de la pila si es posible. (Definir parametros y codificar).
- 84) Dada una pila y un valor **X**, desarrollar un procedimiento que inserte el valor **X** en la ultima posicion de la pila y la retorne. (Definir parametros y codificar).
- 85) Dada una pila y dos valores **X** e **Y**, desarrollar un procedimiento que reemplace cada valor igual a **X** que se encuentre en la pila por el valor **Y** retornando la pila modificada. En caso de no haber ningun valor igual a **X** retornar la pila sin cambio. (Definir parametros y codificar).
- 86) Definir una funcion **INVERSA** que evalúe dos conjuntos de caracteres separados por un punto y retorne True si los conjuntos son inversos (ej: ABcDe.eDcBA) o False si no lo son. Los conjuntos deben ingresarse por teclado. (Definir parametros y codificar).
- 87) Desarrollar un procedimiento que ingrese por teclado un conjunto de Apellidos y Nombre de alumnos y los imprima en orden inverso al de ingreso. (Definir parametros y codificar).
- 88) Dada una pila desarrollar un procedimiento que ordene la misma de acuerdo al valor de sus nodos y la retorne. Solo se deben usar pilas. (Definir parametros y codificar).
- 89) Dada una cola (nodo = registro + puntero), desarrollar y codificar un procedimiento que elimine 2 nodos de la misma (indicar con un parametro '**S**'/'**N**' si ello fue, o no posible)
- 90) Dada una cola (nodo = registro + puntero), desarrollar y codificar una funcion que devuelva la cantidad de nodos que tiene.
- 91) Dadas dos colas **COLA** y **COLB** (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere una unica cola **COLAB** a partir de ellas. (Primero los nodos de **COLA** y luego los de **COLB**).
- 92) Dada una cola (nodo = registro + puntero), imprimirla en orden natural si tiene mas de 100 nodos, caso contrario imprimirla en orden inverso.
- 93) Dadas dos colas **COLA** y **COLB** (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere otra cola **COLAB** por apareo del campo **ARRIBO** del registro (define orden creciente en ambas).

**Nota:** **COLA** y **COLB** dejan de ser utiles despues del apareo.

94) Dado un archivo de registros de alumnos, donde cada registro contiene:

- |   |                                 |
|---|---------------------------------|
| a) Apellido y Nombre del alumno (35 caracteres) | b) Numero de legajo (7 digitos) |
| a) Division asignada (1 a 100)                  |                                 |

ordenado por numero de legajo, desarrollar el algoritmo y codificacion del programa que imprima el listado de alumnos por division, ordenado por division y numero de legajo crecientes, a razon de 55 alumnos por hoja.

95) Idem Ejercicio Nro. 93, pero el listado de alumnos por division debe realizarse ordenado creciente por division y decreciente por numero de legajo.

96) Idem Ejercicio Nro. 93 pero considerando que las divisiones asignadas son 100 y se identifican con un codigo de 4 caracteres.

97) Dado un arreglo de **N** (< 30) colas (nodo = registro + puntero), desarrollar y codificar un procedimiento que aparee las colas del arreglo en las mismas condiciones que las definidas en el Ejercicio Nro. 72.

**Nota:** Retornar la cola resultante y no mantener las anteriores.

98) Dada una lista (nodo = registro + puntero), desarrollar y codificar una funcion que devuelva la cantidad de nodos que tiene.

99) Dadas dos listas **LISTA** y **LISTB** (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere una unica lista **LISTC** a partir de ellas. (Primero los nodos de **LISTA** y luego los de **LISTB**).

100) Dada una **LISTA** (nodo = registro + puntero), imprimirla en orden natural si tiene mas de 100 nodos, caso contrario imprimirla en orden inverso.

101) Dadas dos listas **LISTA** y **LISTB** (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere otra lista **LISTC** por apareo del campo **LEGAJO** del registro (define orden creciente en ambas). **Nota:** **LISTA** y **LISTB** dejan de ser utiles despues del apareo).

- 102) Dado un archivo de registros de alumnos **ARCHA** sin ningun orden, donde cada registro contiene: a) Apellido y Nombre del alumno (34 caracteres) b) Numero de legajo (6 dlgitos)  
a) Division asignada (1 a 100)  
Se debe desarrollar el algoritmo y codificacion del programa que genere un archivo **ARCHL** igual al anterior pero ordenado por numero de legajo.  
**Nota:** Memoria estatica 64 Kb; dinamica suficiente si ningun nodo ocupa mas de 12 bytes.
- 103) Dado un archivo de registros de alumnos **ARCHA** sin ningun orden donde cada registro contiene: a) Apellido y Nombre del alumno (34 caracteres) b) Numero de legajo (6 dlgitos) c) Division asignada (3 dlgitos)  
Se debe desarrollar el algoritmo y codificacion del programa que imprima el listado de alumnos por division ordenado por division y numero de legajo crecientes, a razon de 55 alumnos por hoja.