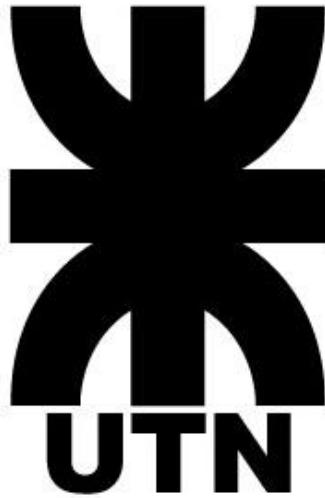

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL RESISTENCIA



INGENIERIA EN SISTEMAS DE INFORMACION

Algoritmos y Estructuras de Datos

Ciclo 2025

Profesora Titular

Ing. en Sist. de Inf. Noelia PINTO

Profesores Adjuntos

Ing. en Sist. de Inf. Diego BOLATTI

Ing. en Sist. de Inf. Nicolás TORTOSA

Jefes de Trabajos Prácticos

Ing. en Sist. de Inf. Ilse HODAPP

Ing. en Sist. de Inf. Axel ROBLES

Ing. en Sist. de Inf. Fernando FIGUEREDO

Algoritmos y
Estructuras de
Datos

Práctico 1

Tratamiento de datos elementales
Uso de Estructuras Secuenciales
Uso de Estructuras Selectivas y Repetitivas
Ejercicios Complementarios

2025

TRABAJO PRACTICO Nro. 1

1. Suponga que los estados iniciales de K, Q y DOS son 4, 7 y 10. Especifique cuáles son los estados iniciales y finales de cada una de las variables después de las siguientes asignaciones.

- 1.01. $TOT := K * (DOS - Q)$
- 1.02. $D := DOS / K + Q * 2$
- 1.03. $E := TRUNC(Q * 0,8 - DOS / 3)$
- 1.04. $R := REDOND(Q / (3 / K))$
- 1.05. $K := ABSO(K - Q) + 2 * DOS$
- 1.06. $RESULTADO := REDOND(ABSO((K - DOS) / Q) + TRUNC(1,5))$

2. Indique los estados iniciales, intermedios y finales para cada variable incluida en los siguientes enunciados:

- 2.01. $A := 1,3; X := 0; Y := 8;$
 $Z := A + Y * 2 - X / 3;$
 $X := X + 2;$
 $C := (Z - X) * 1,2;$
- 2.02. $C := 1;$
 $C := 0;$
 $CA := REDOND(C * 2,4 + 1);$
 $X := CA + 2;$
- 2.03. $BETA := 3/2;$
 $ALFA := BETA * 2;$
 $AUXI := ALFA;$
 $ALFA := BETA;$
 $BETA := AUXI;$
- 2.04. $C := TRUNC(ABSO(4/-2) + 3);$
 $B := REDOND(-4,3);$
 $B := B + C;$
- 2.05. $ALFA := 2,5; JOTA := 2;$
 $BETA := ALFA ** JOTA;$
 $I := 2 * JOTA;$
 $L := JOTA * I;$
 $ALFA := ALFA + BETA;$
- 2.06. $NUM := ABSO(-10); A := 3;$
 $NUM2 := TRUNC(NUM / A) ** 2 / A;$
 $NUM := NUM2 - NUM * 5;$
 $B := REDOND(NUM / NUM2);$
 $A := (A + B) - 9 * 2 - 1;$

3. Encuentre el valor de la variable RESULT después de la ejecución de las siguientes secuencias de operaciones. Realizar prueba de escritorio.

- 3.01.** `RESULT := 5 + 2**3*2;`
- 3.02.** `X := 2; Y := 3;`
`RESULT := X**Y - X;`
- 3.03.** `UNO := 10; DOS := 200;`
`RESULT := ABSO (UNO - DOS);`
`UNO := TRUNC (RESULT / 3) - UNO ** 2;`
`RESULT := (DOS - UNO) * 2;`
- 3.04.** `VBLE1 := 10; VBLE2 := 3;`
`RESULT := TRUNC (VBLE1 / VBLE2);`
`VBLE1 := VBLE1 + VBLE2;`
`RESULT := ABSO (VBLE2 - VBLE1) ** 2;`
`RESULT := RESULT * 0,1;`
- 3.05.** `X := 10; RESULT := 3;`
`TRES := ABSO (RESULT - X) + 2 * RESULT;`
`X := REDOND (TRES / 2);`
`Z := X MOD RESULT;`
`RESULT := (2*(-5) + X)**2;`
- 3.06.** `UNO := 1; DOS := 2; TRES := 4;`
`DOS := ABSO (UNO - TRES);`
`RESULT := DOS * TRES;`

4. Indique en cuáles de los siguientes pares es importante el orden de los enunciados. Es decir, que si se modifica el orden de ellos cambian los resultados (supóngase $X \leftrightarrow Y$, $Y \leftrightarrow Z$, $Z \leftrightarrow X$). Justifique su respuesta.

- | | |
|---|---|
| 4.01. <code>X:=Y</code>
<code>Y:=Z</code> | 4.02. <code>X:=Y</code>
<code>Z:=X</code> |
| 4.03. <code>X:=Z</code>
<code>X:=Y</code> | 4.04. <code>Z:=Y</code>
<code>X:=Y</code> |

5. Diseñe un algoritmo que permita resolver cada uno de los ejercicios enunciados a continuación:

5.01. Leer los datos de un automóvil en la forma "Marca" seguida de "Modelo", y escribir en la forma "Modelo" seguido por "Marca".

5.02. Convertir una suma en dólares a PESOS. Se debe prever que el valor de conversión puede cambiar.

EJERCICIOS CON ESTRUCTURAS SECUENCIALES

6. Diseñe un algoritmo que permita resolver cada uno de los ejercicios enunciados a continuación:

6.01. Escribir un algoritmo que permita calcular el precio de un artículo para un año dado, considerando que la inflación es del 4 por 100 anual.

La fórmula del precio es: $P = C * (1+R)^{(N - A)}$

C - Precio actual.

R - Tasa de Inflación.

N - Año futuro.

A - Año actual.

6.02. Dado un número entero de tres cifras ingresado por el usuario, escriba un algoritmo que determine el valor de la centena, la decena y la unidad. Finalmente, muestre por pantalla los tres componentes por separado y el resultado de la suma de sus dígitos

6.03. Se desea comprar una PC y una impresora. Calcular el precio total: el cual está dado por la suma de los precios de costos, los porcentajes de ganancia del vendedor y un 21% de IVA. Supóngase una ganancia del vendedor del 12% por la PC y 7% por la impresora. Se leen los costos y se imprimen el precio total de ventas.

6.04. Se desea calcular la superficie de un trapecio, para la cual se ingresa la longitud de ambas bases y la altura. En base a la fórmula:

$$S = \frac{(B_{may} + B_{men}) * h}{2}$$

Finalizando el proceso, emitir dicha superficie y los valores ingresados.

6.05. Se desea calcular el valor de la sección (S) de un conductor, la cual se determina en función de la corriente eléctrica I y de la conductividad C del material ($C=I/S$). Además, a la sección así obtenida se le incrementa un 25% por razones de seguridad.

6.06. En una concesionaria de autos oficial de una marca venden 3 modelos diferentes, el gerente desea saber qué porcentaje del total disponible de cada modelo se vendió este mes. Escriba un algoritmo que ingrese los datos necesarios y muestre por pantalla el porcentaje vendido para cada modelo.

6.07 Debido a los frecuentes aumentos de combustible, el propietario de una estación de servicio le pide a usted que escriba un algoritmo para una “calculadora de precios”. El mismo debe recibir el precio actual, el porcentaje de aumento, calcular el nuevo precio y al final mostrar todo.

6.08. Escriba un algoritmo que halle la media geométrica de tres valores X, Y, Z. Este debe emitir los tres valores por separado y luego el valor medio. La media geométrica es igual a $(X*Y*Z) / 3$.

EJERCICIOS CON ESTRUCTURAS CONDICIONALES

7. Escriba un algoritmo que permita ingresar 3 valores numéricos y determine cuál es el mayor, el medio y el menor.

8. Escriba un algoritmo que acepte dos números, calcule la suma e imprima el mensaje de acuerdo al resultado obtenido.

```
suma <= 50
suma > 50 y <= 100
suma > 100 y <= 200
suma > 200
```

9. Escriba un algoritmo que permita conocer la edad de una persona, con solo ingresar la fecha de nacimiento y la fecha actual, ambas en el formato: DIA, MES, AÑO.

10. Una persona decide realizar un viaje a Europa, para lo cual necesita una determinada cantidad de euros. La persona tiene ahorrada una cierta suma en dólares y desea saber si es suficiente y, en caso de haber diferencia (de más o de menos) a cuántos pesos equivale. Realice un algoritmo que solucione el problema, para lo cual deberá prever que se ingresen las equivalencias de monedas que considere necesarias (por ejemplo la cotización en pesos de dólar y/o del euro, o a cuántos euros equivale un dólar).

11. Dados dos números enteros A y B generar un algoritmo que permita determinar si A es divisor de B o B es divisor de A.

12. Escriba un algoritmo que acepte un número entero mayor a 100 y menor a 1000, y muestre cómo está compuesto (unidades, decenas y centenas) y si es múltiplo de 3.

13. Escriba un algoritmo que acepte un número entero mayor a 100 y menor a 1000 que representa una suma de dinero e indique cuántos billetes de cada denominación necesita, suponiendo que solo existen billetes de 100, 10 y 1 peso.

14. Escriba un algoritmo que lea el monto total de una compra en un supermercado y determine el total a pagar según los siguientes criterios:

Si el monto de la compra supera los \$ 200.000, se aplica un 15% de descuento.

Luego, se consulta el tipo de entrega (el usuario ingresa el número de opción):

- Opción 1 → retiro en sucursal, no se suma ningún cargo adicional.
- Opción 2 → envío a domicilio dentro de Resistencia, se suma un 5% sobre el monto final.
- Opción 3 → envío a domicilio fuera de Resistencia, se suma un 10% sobre el monto final.

EJERCICIOS CON ESTRUCTURAS CONDICIONALES Y REPETITIVAS

15. a) Hacer un algoritmo que calcule la altura aproximada de un edificio en pies, ingresando como dato la cantidad de pisos del mismo y la altura promedio de cada piso, en metros. (1 m = 3.28 pies)

b) Modifique el algoritmo del punto a) para que permita calcular la altura de 50 edificios.

c) Modifique el algoritmo del punto a) para que permita calcular la altura de una cantidad indeterminada de edificios. Prevea una condición de fin.

16. a) Diseñe un algoritmo que obtenga el porcentaje de alumnos de ISI, IQ e IEM, sobre el total de egresados de la UTN-FRRe de un año.

b) Modifique el algoritmo del punto a) para que permita obtener e informar los mismos porcentajes, pero para varias Facultades y al final emitir el total de alumnos por carrera y total general.

17. Elabore un algoritmo que calcule el producto de dos enteros A y B empleando sólo la operación suma.

18. Elabore un algoritmo que calcule el cociente de dos enteros F y G y el resto de la operación, empleando sólo las operaciones suma y diferencia.

19. Escriba un algoritmo que determine si un número es primo.

20. Escriba un algoritmo para imprimir los números primos menores a un valor dado n.

21. Escriba un algoritmo para calcular cada renglón de una factura (valor unitario * cantidad vendida) y el total de la misma, suponiendo que el número de renglones es variable. Emitir un mensaje de error si el valor unitario es ≤ 0 . Realizar la prueba de escritorio con los siguientes valores:

Valor Unitario	Cantidad vendida
2	10
1	25
3	15
2	8,5

22. Escribir un algoritmo que, dado un importe dinero, permita calcular e informar cuánto corresponde pagar por un impuesto, en cuántas cuotas y el valor de las mismas. Tener en cuenta los siguientes datos:

- IMPUESTO = 10% del importe dado.
- Los importes mayores que \$ 10.000 y menor o igual que \$ 50.000 se pagan en dos cuotas.
- Los mayores de \$ 50.000 en tres cuotas.

El algoritmo debe permitir tratar varios importes, finalizando cuando se ingresa 9999 como importe.

23. Elabore un algoritmo para calcular los primeros 50 números de FIBONACCI, sabiendo que dichos números cumplen con lo siguiente:

$$A_0=0, A_1=1, A_2=A_0+A_1, \dots, A_n=A_{n-1}+A_{n-2}.$$

24. Construya un algoritmo capaz de encontrar todas las cifras de tres dígitos que cumplan con la condición de que la suma de los cubos de sus dígitos sea igual al número que la cifra representa.

25. Escriba un algoritmo para resolver el siguiente problema: Una empresa de transportes desea conocer el sueldo de sus 100 choferes. Estos se calculan teniendo en cuenta la categoría (1 o 2) y la asistencia (perfecta: sí o no).

El sueldo se obtiene sumando el sueldo básico, más el 2% de antigüedad por cada año trabajado y \$200.000 de premio por asistencia.

El sueldo básico es de \$700.000 para choferes de categoría 1 y de \$600.000 para los de categoría 2.

26. Una fábrica textil produce telas de dos calidades distintas (primera y segunda) y de dos materiales distintos (seda y algodón). Generar un algoritmo que calcule el peso de varias piezas de tela, el cual está dado por la suma del peso neto más un porcentaje por el apresto, más el peso del núcleo de cartón.

Para realizar el cálculo, tener en cuenta la siguiente información, para cada pieza:

- El peso del m² y la longitud de cada pieza.
- Al peso neto de la tela hay que sumarle un porcentaje por cada pieza, debido al apresto, el cual es del 2% para las telas de seda y del 7% para las de algodón.
- También se debe considerar el núcleo de cartón, que es de 400 gr. para los rollos de telas de primera y de 300 gr. en los de segunda.

Finalizar cuando la variable FIN sea igual a 'SI'.

27. La fecha del domingo de Pascua corresponde al primer domingo después de la primera luna llena que sigue al equinoccio de primavera. La secuencia de cálculos que permite conocer esta fecha es:

$$A = \text{año} \bmod 19$$

$$B = \text{año} \bmod 4$$

$$C = \text{año} \bmod 7$$

$$D = (19 \cdot A + 24) \bmod 30$$

$$E = (2 \cdot B + 4 \cdot C + 6 \cdot D + 5) \bmod 7$$

$$N = (22 + D + E)$$

Donde N indica el número del día del mes de marzo (o abril si N es superior a 31) correspondiente al domingo de Pascua. Realizar un algoritmo que determine esta fecha para los años comprendidos entre 1990 y 2010.

28. Escribir un algoritmo que permita imprimir la siguiente sucesión. Considere que N es un número par, que se ingresa.

```

2  4  6  8  ... N
4  6  8  10 ... N
6  8  10 12 ... N
....
....
N

```

29. Escribir un algoritmo que permita imprimir la siguiente sucesión. Considere que N es un número par, que se ingresa.

```

2 4 6 ..... N
2 4 6 ..... N-2
2 4 6 ... N-4
.....|
2 4 6
2 4
2

```

30. Escriba un algoritmo que acepte un número entero que representa una suma de dinero e indique cuántos billetes de cada denominación necesita, suponiendo que solo existen billetes de 500, 100, 50, 20, 10, 5 y 1 peso.

31. Todo número cuya suma de sus dígitos sea múltiplo de 3 lo es también.

Ej: $117 \rightarrow 1+1+7 = 9$ que es múltiplo de 3, entonces 117 es múltiplo de 3

Realizar un algoritmo que determine si un número es múltiplo de 3 en función de la afirmación antes realizada.

32. Funciones y Procedimientos

NOTA: Para cada función y/o procedimiento solicitado, escriba en pseudocódigo el algoritmo que lo invoque.

32.01. Realice una función que dado un número devuelva su cuadrado.

32.02. Diseñe una función que devuelva la suma de los dígitos del número natural suministrado como parámetro.

32.03. Diseñe una función que permita ingresar 3 números y devuelva el mínimo valor. El programa principal debe permitir que este proceso se repita la cantidad de veces que el usuario desee.

32.04. Elaborar una función que reciba un número entero y retorne -1 si el número es negativo.

Si el número es positivo debe devolver una clave calculada de la siguiente manera: Se suma cada dígito que compone el número y a esa suma se le calcula el módulo 7.

Por ejemplo: para la cifra 513, la clave será $5+1+3=9$; $9 \bmod 7=2$.

Utilice la función para diseñar un algoritmo que permita leer varios valores y determine, para cada uno, si el número leído fue negativo o, si fue positivo, que clave le corresponde.

32.05. Escribir una función que verifique si un caracter introducido es un dígito o no.

32.06. Escribir una función que lea desde el teclado las unidades y el precio de un producto que se quiere comprar, y en función de las unidades introducidas calcule un descuento o no, según corresponda: cuando las unidades excedan media docena se aplicará 4% y el 10% cuando excedan la docena. La función debe devolver como resultado el valor del descuento o cero, en caso de que no corresponda.

32.07. Repita el ejercicio del punto 5 pero utilice procedimiento en lugar de función.

32.08. Crear un procedimiento que intercambie los valores de dos variables numéricas.

32.09. Desarrollar los procedimientos mostrar_0, mostrar_1,..., mostrar_9 que visualicen por pantalla, respectivamente, los números 0,1,...,9 en tamaño grande, tal y como se muestra a continuación:

```

****      *      ****      ****      *      *      ****      ****      ****      ****      ****
*      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *
****      *      ****      ****      *      *      ****      ****      ****      ****      ****

```


32.10. Escribir un único procedimiento "mostrar_nro" que reciba como parámetro un dígito y lo muestre por pantalla de la misma forma que se indica en el ejercicio 9.

32.11. Diseñe un algoritmo que muestre un menú con las opciones sumar, restar, multiplicar y dividir, el algoritmo solicitará una opción y realizará la tarea elegida, se debe usar un procedimiento para mostrar el menú, pedir los datos en el algoritmo principal y después usar funciones para realizar los cálculos.

32.12. Escribir una función que lea desde el teclado las unidades y el precio que quiere comprar, y en función de las unidades introducidas le haga un descuento o no (cuando las unidades excedan media docena se aplicará 4% y el 10% cuando excedan la docena).

Ejercicios Complementarios TP N° 1

1. La Mutual del personal del Banco, a pedido de un socio, canjea un plazo fijo, pagándole el capital inicial. Luego, al cobrarse el plazo fijo, le paga los intereses, descontando los días correspondientes y \$5 por gastos. Se desea liquidar el pago de la diferencia entre el capital inicial y capital con intereses, sabiendo que el interés es el 5% a 30 días y teniendo como datos el capital inicial, capital con intereses y los días de anticipo.
2. Se desea ingresar dos números X, Y. Hallar la media aritmética y la geométrica de ambos valores. Diseñe un algoritmo que sea capaz de hallar el máximo entre los dos valores.
3. Escriba un algoritmo que determine si un valor entero es par o impar.
4. Genere un algoritmo que determine el factorial de un número dado N.
5. Se desea determinar si un estudiante aprobó una materia. Para ello se le toman 5 exámenes (E1, E2, E3, E4, E5). Se considera aprobada la materia a aquellos alumnos que aprobaron los 5 exámenes: deberán rendir un examen de complemento aquellos que aprobaron 3 exámenes, siempre que uno de estos tres sea el último. Tendrán la posibilidad de volver a rendir el último examen aquellos alumnos que no lo aprobaron, pero aprobaron por lo menos el 3ro. y el 4to.
6. Las medidas de las tablas de surf, como invento norteamericano que son, se expresan en el sistema métrico anglosajón, es decir, en pies y en pulgadas. De esta manera si decimos que nuestra tabla es una 6'2" estaremos diciendo que su longitud es de 1,88 m. Escriba un algoritmo que calcule la longitud de una tabla en metros, conociendo la misma en formato anglosajón. Las equivalencias con nuestro sistema métrico serían las siguientes: 1 pie = 12 plg, 1 plg = 2,54 cm, 1 m = 100cm.
7. Se desea obtener una tabla con 5 valores de e^x , usando para ello la siguiente fórmula:

$$e^x = 1 - x/1! + x^2/2! - x^3/3! + \dots + x^{20}/20!$$
8. Se desea realizar el cálculo de una serie e^x que se detiene cuando un término es menor o igual que 0,000001.

Algoritmos y
Estructuras de
Datos

Práctico 2

Tratamiento de Secuencias de Datos Elementales
Tratamiento de Secuencias de Registros
 Corte de Control
 Mezcla
 Actualización
Tratamiento de Archivos Indexados
Complemento Teórico
Ejercicios Complementarios

2025

TRABAJO PRACTICO Nro. 2**1. Tratamiento de Secuencias de Datos Elementales.**

1.01. Dada una secuencia de letras del alfabeto que finaliza con una marca "*", contar cuantas letras "A" hay en la secuencia.

1.02. Dada una secuencia de letras del alfabeto que finaliza con la letra "Z", contar cuantas consonantes hay en la secuencia.

1.03. Se dispone de una secuencia de caracteres y se desea obtener una secuencia de salida que resulte de copiar la secuencia de entrada, descartando el caracter "\$".

1.04. Se dispone de una secuencia de números de socios, y se desea saber la cantidad de socios que están registrados.

1.05. La secuencia de socios del problema anterior tiene el inconveniente de que los números están ordenados pero no son correlativos. Se desea generar una secuencia que contenga los números de socios que no figuran en la secuencia de socios.

1.06. Dada una secuencia de enteros que almacena la cantidad de habitantes de las ciudades capitales de las 23 provincias de la República Argentina, discriminados 4 categorías: menores de 18 años (varones y mujeres) y mayores de 18 años (varones y mujeres). Se pide determinar la población total y los siguientes porcentajes: masculinos, femeninos, mayores de 18 y menores de 18.

1.07. Se tiene una secuencia de enteros que contiene todos los CUIT de los empleados de una empresa, la misma termina con el CUIT 0. Para evitar largas esperas en los días de pago, la empresa necesita organizarlos de acuerdo al último dígito de su documento, generar una secuencia con los CUIT de las personas que su número de documento termine con 0, 1, 2 o 3.

1.08. Teniendo en cuenta el ejercicio anterior, se solicita que la secuencia de salida sea una secuencia de caracteres y los CUIT se separen unos de otros con el caracter "-".

1.09. Se dispone de una secuencia de caracteres. Se desea determinar la cantidad de *palabras* que comienzan con la letra "I".

1.10. Se dispone de una secuencia de caracteres. Se desea permita contar la cantidad de *palabras* que comienzan con una letra dada.

1.11. Dada una secuencia de caracteres, determinar la cantidad de *palabras* de 4 caracteres (letras)

1.12. Se dispone de una secuencia de caracteres. Se desea listar las *palabras* que comiencen con "ALG".

1.13. A partir del ejercicio anterior, determinar el porcentaje que representan las *palabras* que comienzan con "ALG" sobre todas las palabras de la secuencia

1.14. Se dispone de una secuencia de caracteres y se desea saber la cantidad de caracteres (incluidos los espacios) que existen entre una coma y la siguiente. Se debe considerar que puede haber más de un par de comas, y que las subsecuencias inicial y final deben descartarse por no cumplir la condición enunciada. Supóngase que las comas son siempre contiguas al último carácter de la palabra.

1.15. Se desea saber la cantidad promedio de palabras que contienen las oraciones de una secuencia de caracteres. Supóngase que los puntos son siempre contiguos al último carácter de la palabra. La secuencia finaliza con una marca.

1.16 Se dispone de una secuencia de números de DNI asignados a un circuito electoral, generar otra secuencia de números que contenga los DNI múltiplos de 3.

1.17. Se desea calcular el costo de un telegrama, que se determina en función del número de palabras (que vale V1 cada una), salvo que el promedio de letras por palabra supere las cinco letras, caso en que cada palabra vale V2.

1.18. Escribir un algoritmo que produzca una secuencia de salida que contenga una oración formada por las palabras en posición par de cada oración de una secuencia texto de entrada, que además comienzan con la letra "M".

1.19. Dada una secuencia de caracteres, generar otra secuencia con todas las palabras de tres caracteres. Informar el porcentaje de las mismas, sobre el total de palabras de la secuencia de entrada. Considerar que puede haber más de un blanco entre palabras.

1.20. Se dispone de dos secuencias de texto formadas por oraciones bimembres (sujeto y predicado, separados por comas ','). Se desea una secuencia texto de salida formada por oraciones bimembres, de la siguiente forma:

- El sujeto, de la secuencia 1, y el predicado, de la secuencia 2.

Al finalizar informar cuantas oraciones tiene cada secuencia.

1.21. Se dispone de dos secuencias de texto formadas por oraciones bimembres (sujeto y predicado, separados por comas ','). Se desea una secuencia texto de salida formada por oraciones bimembres, de la siguiente forma:

- El sujeto, de la secuencia 2, y el predicado, de la secuencia 1.

Al finalizar informar cuantas oraciones tiene cada secuencia.

1.22. La empresa Ideas Argentinas S.A. posee datos de sus empleados en dos secuencias de caracteres; la primera secuencia (Sec1) formada por los nombres (uno por persona) de los empleados separados por blancos y la segunda secuencia (Sec2) posee los números de documento de cada uno de los empleados (palabras de 8 dígitos numéricos). Ambas secuencias poseen la misma cantidad de datos, es decir al primer nombre de la primera secuencia le corresponde el primer número de documento de la segunda secuencia y así sucesivamente. La secuencia de números de documentos no posee espacios en blanco ni ningún otro tipo de caracter que separe un número de documento de otro.

A la empresa le interesa tener en una nueva secuencia (Sec3) los datos de todas aquellas personas que cumplan la condición de que el nombre se encuentre en una posición impar.

La nueva secuencia debe generarse de la siguiente manera: el número de documento seguido (sin espacios) por una coma y luego (sin espacios) por el nombre y seguido al nombre un #.

1.23. Realice un algoritmo para el enunciado del ejercicio 1.22, pero considerando que los datos que se deben copiar en la Sec3 son los de aquellas personas que cumplan la condición de que: el número de documento comienza con un número impar.

1.24. Realice un algoritmo para el enunciado del ejercicio 1.22, pero considerando que los datos que se deben copiar en la Sec3 son los de aquellas personas que cumplan la condición de que: el nombre no comienza con una vocal.

1.25. Dada una secuencia texto de entrada que contiene palabras alfanuméricas, escribir un algoritmo que genere dos secuencias de salida. Una de ellas contendrá solo las palabras de la secuencia original que comienzan con vocal, en las cuales se reemplazarán todas las vocales por '#', por ejemplo: entrada "avión1", salida "#v###n1" y la otra será una secuencia numérica en la que se almacenarán las cantidades de vocales que se encontraron en cada una de las palabras que cumplieron la condición. Por final de proceso se deberá informar el promedio de palabras por oración.

1.26. Se posee 2 secuencias (S1 y S2) con las cuales se desea generar una nueva secuencia (SAL) donde se intercalen las palabras de las secuencias de entrada, de la siguiente manera: copiar de S1 aquellas palabras que empiezan con una letra que ingresa el usuario y de S2 aquellas palabras que estén en posición par.

1.27. Elaborar un procedimiento que dada una secuencia de caracteres como entrada genere otra del mismo tipo como salida.

La secuencia de caracteres que se recibe incluye números de tarjetas de crédito, donde cada número tiene 16 dígitos. Se desea obtener como resultado una nueva secuencia de salida con los números de tarjetas válidos.

El algoritmo para la validación de números de tarjetas de crédito es el siguiente:

Para entender mejor el método usaremos un número correcto: 4073-7001-0977-4812, al que nos referiremos en el texto. Nos centramos en los caracteres que ocupan las posiciones 1 y 3

4 0 7 3 7 0 0 1 0 9 7 7 4 8 1 2

Por cada uno de ellos obtenemos el doble del valor que representan. Si el número resultante es menor que 9, se deja tal cual, en caso contrario, se le resta 9. En nuestro ejemplo: $4 * 2 = 8$; $7 * 2 = 14$, mayor que nueve, se le resta 9. $14 - 9 = 5$. Si el número resultante de la suma de las multiplicaciones y de los dígitos en las posiciones 1 y 3, es múltiplo de 2, y a la vez menor o igual a 15, es un número de tarjeta válido. Para nuestro ejemplo: $8 + 5 = 13$; por lo tanto el número es válido.

1.28. La galería de pintura y arte nacional, **PINTA DE ARGENTINA**, almacena información sobre los artistas y sus obras de arte en secuencias de caracteres. Durante todo el año, las obras de arte son expuestas en eventos de subasta y exposición, en los cuales se comercializan al público en general. A fin de año la Comisión Directiva de la Galería solicita que, a partir de toda esa información, se generen algunos informes.

Se debe tener en cuenta lo siguiente:

- En la secuencia **ARTISTAS**, se almacena estilo de arte ("R" – Renacentista, "M" – Arte Moderno, "B" – Barroco, "S" – Surrealismo), el nombre de cada artista, lugar de nacimiento, edad, y cantidad de obras por artista. Los datos de cada artista están separados entre sí por el símbolo "+" y finalizan con el símbolo "?".
- En la secuencia **OBRAS**, se almacena el nombre de la obra, el año en que fue hecha, su precio, precedido siempre del signo "\$" (solo 3 dígitos) y su estado ("V" – Vendido, "R" – Reservado, "U" – Obra Única).

Todos los datos de las obras están separados por el símbolo "," y finalizan con el símbolo "/".

El creador de cada obra se determina de acuerdo al dato "cantidad de obras" de la secuencia ARTISTAS, por ej.: el autor RENE BARTOL tiene 2 obras, por lo cual las primeras 2 obras de la secuencia OBRAS le pertenecen, las siguientes 6, pertenecen a JUAN B JUSTO, etc.

A continuación, un ejemplo de ambas secuencias:

SECUENCIA ARTISTAS

M+RENE BARTOL+ROSARIO+34+2?R+JUAN B JUSTO+NEUQUEN+61+5?.....

SECUENCIAS OBRAS

SOL Y PARANA,1997,\$913,V/GRITO DE ESPERANZA,2003,\$235,R/PENAS,1997,\$781,V/.....

A partir de lo expuesto anteriormente, se pide:

- a. Generar una secuencia de salida con información de los artistas Renacentistas. La secuencia debe contener el nombre del artista, seguido de sus obras (nombre y año de creación). Los datos correspondientes al mismo artista deben separarse entre sí con el signo "+" y finalizar con el signo "?".
- b. Al final del proceso informar:
 - b.1. la mayor cantidad de obras vendidas por un artista.
 - b.2. el porcentaje de obras de artistas "renacentistas" sobre el total de obras.

2. Tratamiento de Secuencias de Registros.

2.01. Se dispone de una secuencia con los datos de los alumnos de la facultad:

- | | |
|---------------------|-----------------------------------|
| - Apellido y Nombre | - Carrera (ISI - IEM - IQ) |
| - N° de Legajo | - Fecha de Nacimiento |
| - Fecha de Ingreso | - D.N.I. |
| - Sexo (M o F) | - Fecha de último examen aprobado |
| | - Nota |

Se desea un listado de los mismos, con el siguiente formato:

Nro. Legajo	Apellido y Nombre	Documento	Carrera
-------------	-------------------	-----------	---------

2.02. Se dispone de una secuencia de facturas con los siguientes datos:

- N° de Factura
- Fecha
- Total

Se desea un listado de las facturas cuyas fechas sean posteriores a una fecha dada, y cuyos importes totales no alcancen los \$1.000.-, con el siguiente formato:

Nro. de factura	Fecha	Importe total
-----------------	-------	---------------

2.03. Dada la siguiente secuencia de datos con el siguiente formato:

- | | |
|---------------------|----------------------------------|
| - N° de socio | - N° de teléfono |
| - Apellido y Nombre | - Carrera (ISI - IEM - IQ - LAR) |
| - Domicilio | - Cantidad de unidades prestadas |

Correspondiente a los alumnos socios de la Biblioteca, generar la secuencia de los alumnos de "ISI" que tengan prestadas más de 4 unidades bibliográficas.

2.04. A partir de la secuencia del ejercicio 2.01 se desea generar otra secuencia con los alumnos de una carrera dada que aprobaron alguna materia este año, con nota mayor a 7 (siete).

2.05. A partir de la secuencia del ejercicio 2.01 se desea generar un listado con los alumnos que aprobaron la última materia rendida con nota superior a 7, e ingresaron el año pasado, con el siguiente formato:

Nro. Legajo	Apellido y Nombre	Carrera
-------------	-------------------	---------

2.06. Se cuenta con una secuencia que contiene el Apellido y Nombre y Promedio General de los Graduados de una Facultad, y se solicita generar un listado con diversas recomendaciones para cubrir vacantes en una importante Empresa. Las mencionadas recomendaciones serán establecidas del siguiente modo: si el promedio es menor que 7, la recomendación será negativa ("NO"); por el contrario, si el promedio es menor que 8, la recomendación será positiva ("SI"); de lo contrario, si el promedio es menor que 9, la recomendación será favorable ("F"); por último, si el promedio supera los 9 puntos, la recomendación será muy favorable("MF").

El listado deber respetar el siguiente formato:

Apellido y nombre	Promedio	Recomendación
-------------------	----------	---------------

Además, se solicita la impresión del total de graduados que recibieron cada una de las recomendaciones, y el promedio global de los mismos.

2.07. Se dispone de un archivo con los datos de un padrón electoral con la siguiente información:

- | | |
|---------------------|---|
| - Nombre y apellido | - Clase |
| - DNI | - Dirección |
| - N° de Mesa | - Observaciones |
| - N° de Circuito | - Partido (0= independiente, 1='A', 2='B', 3='C') |

Generar:

- a) Una secuencia de salida con nombre y apellido, DNI, y dirección de todas las personas afiliadas al partido "C".
- b) Una secuencia de salida con nombre y apellido, DNI, y dirección de todas las personas no afiliadas a ningún partido y de clase posterior a 1940.

2.08. Se dispone de los siguientes archivos:

Archivo COMPRAS

Nro. de Cliente	Fecha última compra	Monto
-----------------	---------------------	-------

Archivo CLIENTES

Nro. de Cliente	Nombre y Apellido	Domicilio	Teléfono	DNI
-----------------	-------------------	-----------	----------	-----

Ambos archivos están ordenados por Nro. de Cliente en forma ascendente.

Se pide: Listar los nombres, fecha y monto de última compra, de todos aquellos clientes cuya última compra sea posterior a una fecha dada y el monto supere un monto dado.

2.09. Una casa deportiva dispone de un archivo de productos, ordenado por código de producto, que contiene los siguientes datos:

Cod_prod	Tipo (1,2,3)	Marca	Modelo	Descripción	Cant_existente	Precio_Unitario
----------	--------------	-------	--------	-------------	----------------	-----------------

Se ha registrado en el mercado un aumento de precios, el cual no es uniforme para todos los artículos, sino que varía según el tipo de los mismos de la siguiente manera: tipo 1- Calzados: 10%, tipo 2 – Indumentaria: 25%, tipo 3 - Accesorios (pelotas, raquetas, etc.): 50%. Por este motivo el gerente de la empresa solicitó al Departamento de Informática que modifique los precios de acuerdo a los porcentajes antes mencionados.

Generar un nuevo fichero de productos para cumplir con la solicitud del gerente.

Al terminar el proceso informar cantidad total de artículos de cada tipo hay y total general.

Corte de Control

ACLARACION: Los campos sombreados en la representación de los archivos indican los campos por los que estos se encuentran ORDENADOS.

2.10. Del archivo general de alumnos de la U.T.N. - Facultad Regional Resistencia, ordenado por carrera, se desea emitir la cantidad de alumnos correspondientes a cada carrera y el total general.

2.11. Se dispone de un archivo con datos de los alumnos de la U.T.N. con la siguiente información para cada alumno:

Sexo	Carrera	NroLegajo	FechaIngreso	TotalMateriasAprobadas
------	---------	-----------	--------------	------------------------

Se desea un listado con el siguiente detalle por sexo, carrera y total general

- a) Total de alumnos que ingresaron en el año 2009 con más de 20 materias aprobadas.
- b) Total de alumnos que ingresaron en el año 2009 con menos de 20 materias aprobadas.

2.12. Dada una secuencia con información de población de UNA PROVINCIA:

Departamento	Ciudad	Cantidad de habitantes
--------------	--------	------------------------

Generar una secuencia con información de los departamentos de esa provincia, cuyo registro tenga el siguiente formato:

Departamento	Cantidad de habitantes
--------------	------------------------

2.13. Un Casino de la región ha notado un incremento en los costos de las reparaciones de tragamonedas en sus sucursales. Por ello solicitó un informe con la cantidad de reparaciones y sus costos, discriminados según tragamonedas, modelo, marca, sucursal y total general.

Se dispone de un archivo REPARACIONES, con el siguiente formato. Cada registro representa la reparación de un tragamonedas, tener en cuenta que puede existir más de una reparación por tragamonedas.

Cod Sucursal	Marca	Modelo	Cod tragamonedas	Fecha reparacion	Costo reparación
--------------	-------	--------	------------------	------------------	------------------

Realice el algoritmo para emitir el informe con los totales solicitados

2.14. Dados el siguiente fichero con datos de un censo de la ciudad de Resistencia:

Radio	Manzana	Nº de vivienda	Condición de la vivienda	Cantidad de Habitantes
-------	---------	----------------	--------------------------	------------------------

Condición puede ser : Muy Buena, Buena o Mala.

Obtener los siguientes totales de personas que habitan en viviendas cuya condición es muy buena: total en la ciudad y totales por Radio y Manzana).

2.15. El organismo del cual dependen las escuelas de un distrito lleva un archivo con los registros para todos los alumnos de nivel secundario de ese distrito:

Escuela	Año	División	Nombre	Cant. Inasist.
---------	-----	----------	--------	----------------

Se introduce como dato el número de distrito y la cantidad de días de clase dictados en el año.

El archivo está ordenado por número de escuela, año y división.

Los alumnos que superan el 20% de las inasistencias están en situación de LIBRES, de lo contrario son REGULARES.

Se desea conocer:

Para cada división: **Cantidad de alumnos**

Para cada año: **Cantidad total de alumnos libres**

Cantidad de alumnos regulares

Cantidad total de alumnos

Todas las escuelas: **Cantidad total de alumnos**

Porcentaje de alumnos libres.

Promedio de inasistencias por alumnos.

Modelo de listado a emitir:

LISTA DE ALUMNOS REGULARES		
Distrito N°:-----		
Nombre	Porcent. Inasist.	Situación
Pérez	28%	Libre
Solá	16%	Regular
-----	----	-----
Total alumnos Div. A	-----	
Gómez	10%	Regular
-----	----	-----
-----	----	-----
-----	----	-----
Total alumnos Div. B	-----	
Total alumnos libres 1º Año	-----	
Total alumnos regul. 1º Año	-----	
Total alumnos 1º Año	-----	
-----	-----	-----
Total alumnos libres 2º Año	-----	
Total alumnos regul. 2º Año	-----	
Total alumnos 2º Año	-----	

TOTALES ESCUELA		
Total de alumnos		
Porcentaje de alumnos libres		
Promedio de inasistencias por alumnos		
Idem para las demás escuelas		

Mezcla

2.16. Construir un algoritmo que a partir de un fichero de películas nuevas conteniendo:

P_nuevas

NRO_PELICULA	TITULO	GENERO	CANT_COPIAS	FECHA_ESTRENO
--------------	--------	--------	-------------	---------------

y otro fichero de películas existentes, ambos ordenados por película,

Películas

NRO_PELICULA	TITULO	GENERO	CANT_COPIAS	FECHA_ESTRENO
--------------	--------	--------	-------------	---------------

Genere un único archivo (con el mismo formato de los ficheros de entrada) que contenga todas las películas. Considerar que hay un solo registro por película y no se repiten entre ficheros.

2.17. La Secretaría Académica de la Facultad lanza un proyecto para incentivar a aquellos alumnos que realizaron el Cursillo de Ingreso a la Universidad y no lograron aprobarlo en los turnos de Agosto y Febrero, de manera de brindarles apoyo académico con el fin de que, en el Cursillo del año siguiente puedan aprobar los exámenes necesarios e ingresar a la Universidad.

Para esto, dicha Secretaría necesita crear un archivo donde se encuentren todos los aspirantes que realizaron el Cursillo de Ingreso en ambos turnos, y no lograron aprobarlo.

Los datos correspondientes a cada uno de los turnos del Cursillo dictado están almacenados en dos archivos (uno para cada turno), los cuales presentan el siguiente formato:

ASPIRANTES (ordenado por Nro. de DNI)

Nro. de DNI	ApeyNom	Carrera	F. Nac	Email	ColegioSec	FechaInscripcion	Aprobado (SI/NO)
-------------	---------	---------	--------	-------	------------	------------------	------------------

UD debe realizar un algoritmo que permita mezclar los archivos Aspirantes (de Agosto y Febrero) y generar un archivo SEGUIMIENTO con el siguiente formato:

SEGUIMIENTO (ordenado por Nro. de DNI)

Nro. de DNI	ApeyNom	Email	ColegioSec
-------------	---------	-------	------------

Al finalizar el proceso informar la cantidad de aspirantes que se grabaron en el archivo SEGUIMIENTO.

2.18. Un supermercado desea conocer la totalidad de unidades existentes de cada artículo a fin de hacer un control de stock y decidir si se deben comprar nuevas unidades o redistribuir la mercadería existente.

El supermercado posee dos sucursales, cada una de las cuales envió su información en un fichero con el siguiente formato:

Cod_prod	Tipo	Marca	Descripción	Cant_unidades
----------	------	-------	-------------	---------------

Escribir un algoritmo que permita obtener un único fichero de salida, con el mismo formato, que contenga la información solicitada y además, emita un listado con los siguientes datos:

Cod_prod	Tipo	Marca	Descripción	Cant Sucursal 1	Cant. Sucursal 2	Total de Unidades
----------	------	-------	-------------	-----------------	------------------	-------------------

Actualización

2.19 En una Empresa Farmacéutica se posee un archivo Mae_Remedios (ordenado por Clave: Farmacia + Medicamento), el que se actualiza semanalmente, a través de la información que se encuentra cargada en un archivo de Movimientos (ordenado por Clavem: Farmacia + Medicamento, y Codmov), de la siguiente forma:

- Si Clave (Mae_Remedios) es menor que Clavem (Movimientos), simplemente se transfieren los datos del Maestro a la salida y se graban.
- Si Clave (Mae_Remedios) es igual a Clavem (Movimientos) y el Codmov es 1, se considera error y se lista un mensaje indicando el tipo de error; pero si el Codmov es 2, entonces es un remedio que deja de fabricarse y se transfiere el registro al archivo de Remedios vencidos (Rem_Venc) ; pero si el Codmov es 3, se modifica la cantidad actual con la cantidad recibida.
- Si Clave (Mae_Remedios) es mayor que Clavem (Movimientos) y el Codmov es 1, se incorpora el remedio a nuestro Vademecum, considerando que la cantidad recibida configura la cantidad actual y la fecha-vence es 30 días posterior a la fecha actual; pero si el Codmov es 2 o 3 se considera error y se deben producir los correspondientes mensajes de error.

Se considera que sólo existe un registro de movimiento para cada registro del maestro.

MAE_REMEDIOS = MAESTRO_ACTUAL
REGR = REGA

Farmacia	Medicamento	Cant-actual	Fecha-vence
----------	-------------	-------------	-------------

MOVIMIENTOS
REGV

Farmacia	Medicamento	Codmov	Cant-recibida
----------	-------------	--------	---------------

REM_VENC
REGE

Medicamento	Cant-vencida
-------------	--------------

2.20. Se desea generar una secuencia, a partir de la secuencia del problema 2.01, donde se actualice la fecha de aprobación del último examen, para ello se cuenta con información del último turno de examen, según el siguiente detalle:

Fichero EXAMENES

Nro. de Legajo	Carrera	Cód. Materia	Fecha de examen	Nota
----------------	---------	--------------	-----------------	------

Se debe considerar que no necesariamente todos los alumnos se presentan en cada turno de examen, y que un alumno puede presentarse a rendir más de una materia en un mismo turno. Supónganse ambas secuencias ordenadas por Número de Legajo.

2.21. En un práctico para la Facultad un grupo de alumnos debe implementar una Red Social llamada UTNBook. Para lo cual debe utilizar los siguientes archivos:

Amigos (ordenado por Cod_usuario y Cod_Amigo)

Cod_usuario	Cod_amigo	Fecha_amistad	Mensaje_muro
-------------	-----------	---------------	--------------

Cada registro indica la fecha desde que los usuarios son amigos y el último mensaje que un amigo ha escrito en el muro del usuario.

Notificaciones (ordenado por Cod_usuario y Cod_Amigo)

Cod_usuario	Cod_amigo	Cod_mov	Fecha_amistad	Mensaje_muro
-------------	-----------	---------	---------------	--------------

Periódicamente se debe actualizar el archivo Amigos, con el fin de que refleje las amistades que posee cada usuario. En el archivo notificaciones pueden existir tres tipos de acciones: la solicitud de una amistad (Cod_mov = "A"); la eliminación de una amistad (Cod_mov = "B"); o la información de un mensaje que un amigo ha escrito en el muro del usuario (Cod_mov = "M"). Considerar que la eliminación de una amistad implica la baja física de un registro y que hay un solo registro de Notificaciones por cada registro de Amigo.

2.22. IDEM ejercicio 2.22 pero agregando al enunciado: "Al finalizar el proceso se desea conocer: el usuario que posee más amigos."

2.23. Una empresa de distribución de energía eléctrica posee un archivo maestro con los siguientes datos de sus clientes (la fecha de última lectura son las mediciones realizadas hasta el mes de mayo del 2014):

Id Casa | Fecha de ultima lectura | Cantidad de lecturas | Promedio de lectura | Tipo de consumidor |
Ordenado por ID casa.

El campo tipo de consumidor puede ser:

- A (cuando el promedio de lectura es menor a 20K),
- B (cuando el promedio de lectura es menor a 40K) o
- C (cuando el promedio de lectura es igual o supera los 40K).

Cuenta además con el siguiente archivo con datos de las mediciones realizadas en los dos últimos años (desde junio del 2014 hasta junio del 2016)

Id Casa | Fecha de medición | Consumo |
Ordenado por ID casa y Fecha de medición.

Se pide actualizar el archivo maestro con la información de las lecturas hasta el mes de enero del año 2015 inclusive. Los campos que se deben actualizar son: la fecha de ultima lectura, cantidad de lecturas, el promedio y modificar (en caso de ser necesario) el tipo de consumidor.

En caso en que no exista el ID casa (una conexión nueva) se tiene que crear en el maestro y actualizar los otros campos.

2.24. Una obra social lleva el control de los costos que le insume cada afiliado. Considerando como costos, los pagos que la misma debe realizar (en conceptos de honorarios, pagos a farmacias, etc) por los distintos servicios que el afiliado utiliza.

Para esto la empresa cuenta con un archivo CostosPorAfiliado, en el cual se registran la cantidad de atenciones y/o servicios que solicita el empleado y el costo total (en pesos) que la empresa debe pagar por estos.

Este archivo está conformado por registros con el siguiente formato, y está ordenado Cod_Afiliado

CostosPorAfiliado

Cod_afiliado	Fecha_alta	Fecha_baja	Cant_servicios	Costo
--------------	------------	------------	----------------	-------

Cada viernes en la organización, se lleva a cabo un proceso de actualización del archivo costosPorAfiliados. Para poder realizarlo se cuenta con un archivo ServiciosSemanales, ordenado por cod_afiliado y fecha_at, en el cual se registran los servicios que solicitaron los afiliados durante cada semana. Cada registro de este archivo presenta el siguiente formato,

ServiciosSemanales

Cod_afiliado	Fecha_at	Cod-servicio	Costo
--------------	----------	--------------	-------

En el archivo serviciosSemanales puede existir más de un registro por cada afiliado.

Para realizar el proceso de actualización se deben tener en cuenta las siguientes consideraciones.

- Si algún Cod_afiliado de serviciosSemanales, no se encuentra en ningún registro del archivo costoPorAfiliado, y el primer Cod_servicio asociado al mismo es igual a 001 entonces se trata de un nuevo afiliado que fue dado de alta.
- Si Cod_afiliado de serviciosSemanales es igual al de costoPorAfiliados y el Cod_servicio es igual a 000 se trata de un afiliado que fue dado de baja y se considera como fecha de baja el valor que reside en fecha_at. Si en cambio, el Cod_servicio tiene algún otro valor este representa un servicio o atención que solicitó el afiliado; por lo tanto deben contabilizarse la cantidad de servicios que solicitó; como así también los costos que estos insumen.
- Si algún Cod_afiliado de costoPorAfiliados no se encuentra en el archivo serviciosSemanales, se trata de un afiliado que no utilizó los servicios en la semana que se realiza la actualización.
- Cualquier otro caso distinto a los considerados anteriormente se toma como un error, y deben ser informados por pantalla; indicando claramente el motivo del error.

2.25. IDEM 2.22 pero suponiendo que hay mas de un registro de notificaciones por cada registro de amigos.

3. Tratamiento de Archivos Indexados.

3.01. Dado un fichero secuencial de Facturas, ordenado por Nro. de Cliente y Nro. de Factura, con la siguiente estructura:

FACTURAS

NRO_CLI	NRO_FACT	FECHA	IMPORTE
---------	----------	-------	---------

Se desea un listado con el siguiente detalle:

Nro. Cliente Nombre Cliente Total Facturado Cantidad de Facturas

Los datos del cliente se encuentran en un fichero indexado por Nro. de Cliente, que tiene la siguiente estructura:

CLIENTES

NRO_CLI	NOMBRE	D.N.I.	C.U.I.T	DOMICILIO
---------	--------	--------	---------	-----------

3.02. Una empresa dispone de un fichero secuencial con datos de sus empleados, ordenado por número de sucursal, y categoría, con los siguientes datos:

NRO_SUCURSAL	CATEGORÍA (A-B-C)	NOMBRE_EMPLEADO	COD_CURSO	TÉCNICO (SI-NO)
--------------	-------------------	-----------------	-----------	-----------------

y un fichero con datos de cursos, indexado por código de curso:

COD_CURSO	NOMBRE	FECHA	CANT_HORAS
-----------	--------	-------	------------

Emitir un listado informando:

- para cada empleado: sucursal, categoría, nombre del empleado y nombre del curso que debe realizar.
- por sucursal, categoría y toda la empresa:
 - total empleados técnicos
 - total empleados no técnicos
 - total empleados

3.03. Los automovilistas pasan por el peaje del Puente Gral. Belgrano y deben pagar según su categoría, pero además, si ya han pasado previamente dentro del día tienen pase libre.

Teniendo en cuenta el archivo siguiente, construya el algoritmo que realice lo que corresponda: genere el comprobante, indicando el importe a pagar o emita un mensaje indicando que ya pasó anteriormente. Además indique cuáles deberían ser los datos de entrada.

PEAJE indexado por patente, fecha

PATENTE	FECHA	CATEGORÍA	COSTO
XXX-NNN	8N	N	XXX,XX

Tarifas vigentes por categoría:

Categoría	1	2	3	4
Costo	1000	1900	2800	3700

3.04. Crear un algoritmo que simule el trabajo de una caja de supermercado. El algoritmo debe permitir imprimir el ticket de compra y realizar el descuento de stock del producto. Al generar el comprobante del ticket debe guardar los datos en los archivos TICKET y DETALLE_TICKET (el cliente es: "consumidor final" y el NroTicket se genera automáticamente, mediante la función OBTENER_TICKET).

Archivos:

PRODUCTOS (indexado por CodProd)

CodProd	Nombre	CantStock	Precio
---------	--------	-----------	--------

TICKET (indexado por NroTicket)

NroTicket	Fecha	Cliente
-----------	-------	---------

DETALLE_TICKET (indexado por Nro_Ticket)

NroTicket	CodProd1	Cantidad1	CodProd2	Cantidad2	CodProd3	Cantidad3
-----------	----------	-----------	----------	-----------	----------	-----------

Formato de impresión del ticket:

Empresa: _____ CUIT: ____ - ____ - ____ Fecha: ____/____/____
 Cliente: _____

Producto	cantidad	Subtotal
_____	_____	_____
.....
Total: _____		

3.05. Para poder comprar dólares en una entidad bancaria al precio oficial (\$1055), el beneficiario, debe tener un ingreso promedio en los últimos 12 meses equivalente a dos veces el sueldo mínimo vital y móvil (el cual en la actualidad es de \$296.832 por mes). Luego, con el sueldo del mes actual, solo se permite comprar por un importe no superior al 30% del mismo. Por ejemplo: si una persona tiene un sueldo de \$300.000, desde septiembre del año pasado, al día de hoy esa persona cumple la condición para comprar y puede comprar dólares por un monto máximo de \$90.000 (equivalente a U\$S 85,30).

Para ello se cuenta con dos archivos indexados:

- El archivo que contiene la cabecera del sueldo (Clave: DNI, Período y NroRecibo):

DNI: N(8)	Periodo: N(6)	NroRecibo: N(15)	AyN: AN(50)	Empresa: AN(50)
-----------	---------------	------------------	-------------	-----------------

El periodo está representado por 6 caracteres numéricos dispuestos de forma de año/mes (aaaamm). El número de recibo (NroRecibo) es único. Una persona puede tener varios recibos de sueldo.

- Y el archivo que contiene los datos de cada recibo (Clave: NroRecibo):

NroRecibo: N(15)	Monto N(15,2)
------------------	---------------

Dado el escenario descripto, se pide escribir dos algoritmos:

- Algoritmo 1, debe permitir:

a) Que el empleado del Banco ingrese un número de documento de algún interesado en comprar dólares, y le devuelva si está habilitado o no para comprar y, en caso positivo, cuál es el monto máximo en pesos que se le autoriza.

b) Si el interesado desea comprar, solicite el monto en pesos que destinará a la compra, el cual deberá ser descontado de su cuenta. Los datos de la cuenta están en un archivo indexado con la siguiente estructura (indexado por DNI):

Dni: N(8)	NroCuenta: N(25)	Saldo: N(15,2)
-----------	------------------	----------------

c) los puntos a) y b) se repiten hasta que el operador (empleado del Banco) indique que desea finalizar.

- Algoritmo 2, debe permitir:

a) Procesar peticiones de compra, de acuerdo a un archivo de entrada de peticiones, evaluando si es posible realizar la operación o no. Si no es posible, indicar cual es el motivo: 1 – No tiene el ingreso promedio suficiente, o 2 – Pide más del 30 % de su sueldo actual.

Peticiones:

DNI: N(8)	CantSoli: N(15,2)
-----------	-------------------

b) El resultado de la evaluación se debe grabar en un nuevo archivo de salida con el siguiente formato:

Salida:

DNI: N(8)	CantSoli: N(15,2)	Pudo: 'Si'/'No'	Error: 1..2
-----------	-------------------	-----------------	-------------

3.06. Una Municipalidad debe liquidar las patentes de su parque automotor para el cuarto trimestre del año e imprimir un padrón de cobros y deudas, con cortes de importe por grupo, categoría y año de fabricación.

Los archivos son:
AUTOS, secuencial ordenado por CLAVE

GRUPO	CATEGORIA	AÑO FAB	NRO PATENTE	DNI	AYNOM	DOMIC
2 N	1 .. 50	4 N	8 N	8 N	25 AN	30 AN
CLAVE						

DEUDAS, indexado por CLAVED

NRO PATENTE	AÑO DEUDA	TRIMESTRE	IMPORTE
8 N	4 N	1 N	5 enteros 2 dec
CLAVED			

Para el trimestre actual, la cuota a abonar viene en el siguiente archivo:

CUOTAS, indexado por CLAVEC

GRUPO	CATEGORIA	AÑO FAB	IMPORTE
2 N	1 .. 50	4 N	5 enteros 2 dec
CLAVEC			

Antes de imprimir el renglón correspondiente a cada nro de patente se debe verificar si existen deudas pendientes, en cuyo caso se sumaran todos los importes adeudados y se consignarán en la columna de deudas.

PADRON:

CLAVE	DNI	AYNOM	DOMIC	DEUDA	4to TRIMESTRE
XXXXXXXXXX	XXXXX	XXXXXXXXX	XXXXXXXXX	\$ XXX,XX	\$ XXX,XX
			TOTAL	\$ XXX,XX	\$ XXX,XX

3.07. Crear un algoritmo que imprima un reporte como el que se indica, incluyendo totales por Obra Social y Clínica de liquidaciones a médicos. Los archivos que intervienen son:

LIQUIDACIONES (secuencial, ordenado por O.S. y Clínica)

O.S.	Clínica	NroLeg	Mes	Año	Bruto	Desc AFIP	Desc DGR
------	---------	--------	-----	-----	-------	-----------	----------

Neto = Bruto – (Desc AFIP + Desc DGR)

OBRAS SOCIALES (indexado por CodOs)

CodOs	Nombre
-------	--------

CLÍNICAS (indexado por CodCli)

CodCli	Nombre
--------	--------

MÉDICOS (indexado por NroLeg)

NroLeg	ApyNom	Especialidad	DNI
--------	--------	--------------	-----

Reporte:

Obra Social: _____

Clínica: _____

Médicos

Nro. Legajo	Nombre	Neto
_____	_____	\$ _____

.....

Total Clínica..... \$ _____

.....

Total Obra Social..... \$ _____

3.08. Los alumnos que desarrollaban el proyecto UTNBook (ejercicio 2.23) han decidido modificar parte de la aplicación para que esta tenga mayor interacción con el usuario. Es así que decidieron eliminar el archivo de

Notificaciones, provocando así que la inserción nuevos amigos, la eliminación de amistades y la escritura en los muros de otras personas esté a cargo del usuario.

Con estas consideraciones trabajaron con el siguiente archivo:

Amigos (Indexado por Cod_usuario y Cod_Amigo)

Cod_usuario	Cod_amigo	Fecha_amistad	Mensaje_muro
-------------	-----------	---------------	--------------

El proceso para agregar o eliminar amigos y escribir en los muros de estos es ahora el siguiente:

El usuario, al iniciar la sesión, ingresa su Código de Usuario y el Código del amigo. Luego, puede seleccionar tres opciones distintas: Agregar Amigo; Eliminar Amigo y Escribir en el Muro.

Si selecciona Agregar Amigo se realizan las acciones necesarias para incorporar esta nueva amistad al usuario, y en caso de no poder realizarse, se muestra un mensaje por pantalla explicando el motivo que imposibilita la amistad. (por ej. Cod_Usuario inexistente, Amistad entre Usuario y Amigo ya existente, etc.).

En cambio, si selecciona Eliminar Amigo el algoritmo deberá eliminar la amistad en caso de existir, en caso contrario informar el error por pantalla.

Por último, al seleccionar Escribir en el Muro el usuario debe ingresar el mensaje que desea escribir a su amigo y luego el algoritmo deberá registrar este mensaje. De ser necesario informar algún error que pueda ocurrir (por ej. Amistad entre el Usuario y Amigo inexistente, etc).

3.09. Pepsico S.A.I.C. desea que Ud. realice el algoritmo para poner al corriente los saldos de sus clientes y el stock de la empresa. Para ello cuentan con:

Clientes (indexado por ClienteID)

ClienteID	ClienteNombre	ClienteCUIT	ClienteSaldo
-----------	---------------	-------------	--------------

Detalle_Clientes (indexado por NroOperacion)

NroOperacion	FechaOperacion	ClienteID	Importe	FacturaNumero
--------------	----------------	-----------	---------	---------------

Ventas(ordenado por VentaNumero)

VentaNumero	VentaFecha	FacturaNumero	ProductoID	CantidadVendida	ClienteID
-------------	------------	---------------	------------	-----------------	-----------

Productos(indexado por Productoid)

Productoid	CantidadStock	ProductoDetalle	CostoUnitario
------------	---------------	-----------------	---------------

Se debe actualizar el saldo del cliente y además agregar el detalle de la compra que figura en el archivo de Ventas al archivo Detalle_Clientes, por cada venta realizada se deberá descontar la cantidad vendida del stock.

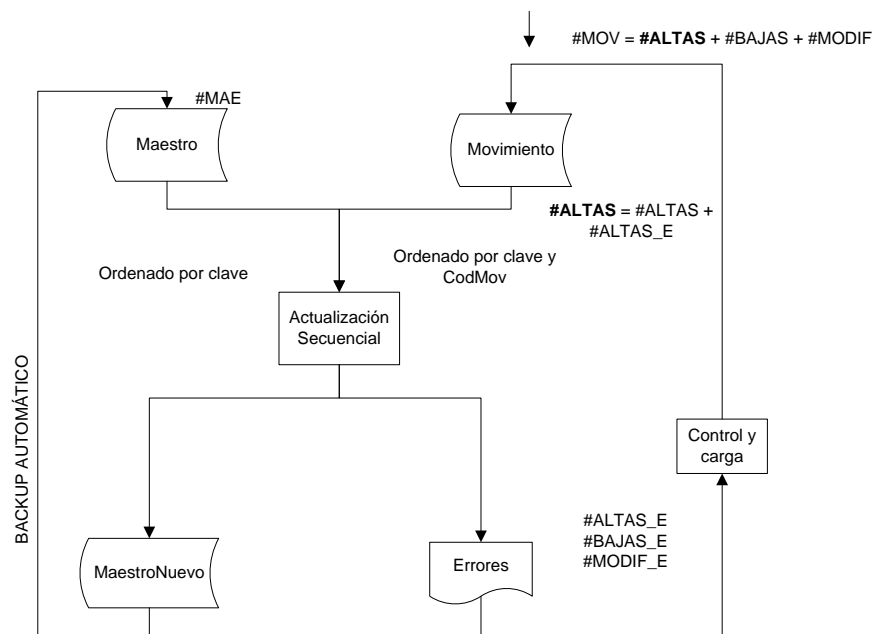
Por final del proceso se desea saber el total de productos vendidos, y un listado de los clientes con su saldo actualizado.

COMPLEMENTOS TEÓRICOS

1 - FICHEROS SECUENCIALES

TIPO	PROCESO	CARACTERÍSTICAS
INDIVIDUALES Un proceso es individual cuando existe un único Fichero de Entrada y 1 o ningún Fichero de Salida.	Genérico	Es el proceso de carga o generación, tiene como objetivo crear un archivo consistente y de ser posible, congruente grueso.
	Emisión	Tiene como objetivo la salida impresa de datos. Son listadores cuando emiten listados como se ingresan sin más que títulos. Se considera padrón cuando los datos ingresados están ordenados y se emiten además totales finales.
	Estadísticos	Recorrido del archivo para contabilizar elementos, utilizando una tabla (memoria interna) y al finalizar emitir un cuadro de resumen.
	Corte De Control	Son padrones, pero poseen totales parciales. Es requisito obligatorio que el archivo de entrada esté ordenado por clave compleja.
MULTIPLES Un proceso es múltiple cuando existen 2 o más Ficheros de Entrada y 1 o más Ficheros de Salida.	Mezcla	Ficheros de entrada: por lo menos dos. Ficheros de salida: uno (resultado de la combinación de los dos de entrada).
	Actualización Secuencial Por Lotes	Cero o más registros del fichero movimiento por cada registro del maestro
	Actualización Secuencial Unitaria	Cero o un (como máximo) registro del fichero movimiento por cada registro del maestro.

Esquema de Actualización Secuencial



Referencias

- #MAE: número de registros existentes en el archivo maestro.
- #MOV : número de registros existentes en el archivo movimiento, compuesto por:
 - #ALTAS + #BAJAS + #MODIF
- #MN: número de registros grabados en el nuevo archivo maestro, compuesto por:
 - #MAE - #BAJAS + #ALTAS
- #ALTAS: cantidad de altas correctas.
- #BAJAS: cantidad de bajas correctas.
- #MODIF: cantidad de modificaciones correctas.
- #ALTAS_E: cantidad de altas erróneas
- #BAJAS_E: cantidad de bajas erróneas.
- #MODIF_E: cantidad de modificaciones erróneas.

Características: Lento, seguro, diferido, batch.

Algoritmos de Procesos con Ficheros

PROCESOS INDIVIDUALES

• CORTE DE CONTROL

Ejemplo para una clave de 3 niveles, clave=

PROVINCIA	DEPARTAMENTO	CIUDAD
Clave 3	Clave 2	Clave 1

ACCION Corte ES

```

Inicializar_totalizadores;
Inicializar_adquisición;
Leer_registro;
    Reg1 := Clave1; Reg2 := Clave2; Reg3 := Clave3      * Resguardar claves *
    MIENTRAS NO FDA(Arch) HACER
        VER_CORTE
        TRATAR_REGISTRO
        Leer_registro;
    FMientras;
    Corte_3;      * corte de mayor nivel *
    Emitir_totales;

```

FACCIÓN.

Acción Ver_corte es

```

SI Clave3 <> Reg3
    ENTONCES Corte3
        SINO SI Clave2 <> Reg2
            ENTONCES Corte2
                SINO SI Clave1 <> Reg1
                    ENTONCES Corte1
                Fsi;
            Fsi;
        Fsi;
    FSI;

```

Facción.

Acción Corte_n es

```

Corte n-1      * llama al corte de nivel inmediato inferior *
Emitir totales n
Acumular totales al nivel inmediato superior * tot n+1 := tot n+1 + tot n *
Reinicializar totales de este nivel      * tot n := 0 *
Reg n := Clave n

```

Facción.

Ejercicio ejemplo de Corte de Control:

Se desea conocer la composición del parque automotor nacional. Para ello se cuenta con información de los vehículos de todo el país, según el siguiente detalle:

- Provincia
- Departamento
- Ciudad
- Patente
- Modelo
- Tipo

Se desea sacar un listado con el siguiente detalle por ciudad, departamento, provincia y total nacional:

	TOTAL	% del TOTAL
Vehículos de más de 5 años	xxxxx	xx,xx %
Vehículos de 5 años o menos	xxxxx	xx,xx %
Total de vehículos	xxxxx	100,00 %

Acción Corte_Control es

AMBIENTE*Constantes*

aa = 2025;

Variables

automot = registro

prov, dpto, ciudad: AN (15);

pat : entero;

mode: 1900 .. 2100;

tipo: AN (5);

Fregistro;

arch: archivo de automot;

aut: automot;

c_totmas, c_totmen: entero;

d_totmas, d_totmen: entero;

p_totmas, p_totmen: entero;

t_totmas, t_totmen: entero;

res_prov: AN (15)

res_dpto: AN (15)

res_ciudad: AN (15)

Subacción print_tot(totmas,totmen: entero) es;

Esc("Vehículos de más de 5 años ',totmas,totmas*100/(totmas+totmen));

Esc("Vehículos de 5 años o menos',totmen,totmen*100/(totmas+totmen));

Esc("Total de vehículos ',totmas+totmen,' 100.00');

FSubacción;

Subacción corte_ciudad es;

Escribir ('PROVINCIA: ',res_prov,' - DEPARTAMENTO: ',res_dpto);

Escribir(' - CIUDAD: ',res_ciudad);

print_tot(c_totmas,c_totmen);

d_totmas:=d_totmas + c_totmas;

d_totmen:=d_totmen + c_totmen;

res_ciudad:=aut.ciudad;

c_totmas:=0; c_totmen:=0;

FSubaccion;

```

Subacción corte_dpto es;
  corte_ciudad;
  Esc('PROVINCIA: ',res_prov,' - TOTAL DEPARTAMENTO: ',res_dpto);
  print_tot(d_totmas,d_totmen);
  p_totmas:=p_totmas + d_totmas;
  p_totmen:=p_totmen + d_totmen;
  res_dpto:=aut.dpto;
  d_totmas:=0; d_totmen:=0;
FSubacción ;

```

```

Subacción corte_prov es;
  corte_dpto;
  Escribir('TOTAL PROVINCIA: ',res_prov);
  print_tot(p_totmas,p_totmen);
  t_totmas:=t_totmas + p_totmas;
  t_totmen:=t_totmen + p_totmen;
  res_prov:=aut.prov;
  p_totmas:=0; p_totmen:=0;
F Subacción ;

```

ALGORITMO

```

Abrir (arch);
Leer (arch,aut)
Res_prov:=aut.prov;
Res_dpto := aut.dpto;
Res_ciudad := aut.ciudad
t_totmas:=0; t_totmen:=0; c_totmas:=0; c_totmen:=0;
d_totmas:=0, d_totmen:=0; p_totmas:=0, p_totmen:=0;

Mientras no FDA (arch) hacer
  Si (aut.prov <> res_prov)
    Entonces corte_prov
    Sino Si (aut.dpto <> res_dpto)
      Entonces corte_dpto
      Sino Si (aut.ciudad <> res_ciudad)
        Entonces corte_ciudad
      Fsi;
    Fsi;
  Si ((aa- aut.mode) > 5) Entonces c_totmas:=c_totmas + 1
    Sino c_totmen:=c_totmen + 1;
  Fsi;
Fcon;
Leer (arch,automot)
Fmientras;

corte_prov;
Escribir('TOTAL NACIONAL');
Esc('Vehículos de más de 5 años ',totmas,totmas*100/(totmas+totmen));
Esc('Vehículos de 5 años o menos',totmen,totmen*100/(totmas+totmen));
Esc('Total de vehículos      ',totmas+totmen,' 100.00');
Cerrar (arch);
Facción.

```

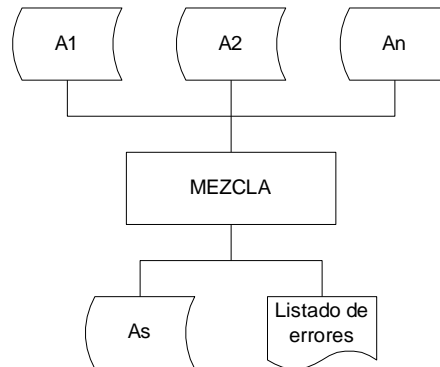
PROCESOS MÚLTIPLES

- Se aplica la “**técnica de apareo**”
- Las estructuras de los ficheros deben tener un **elemento común**: la “clave de apareo” o “campo clave”.
- Los ficheros deben estar **ordenados** por el “campo clave”.

• MEZCLA

Ficheros de entrada: por lo menos dos.

Ficheros de salida: uno (resultado de la combinación de los dos de entrada).



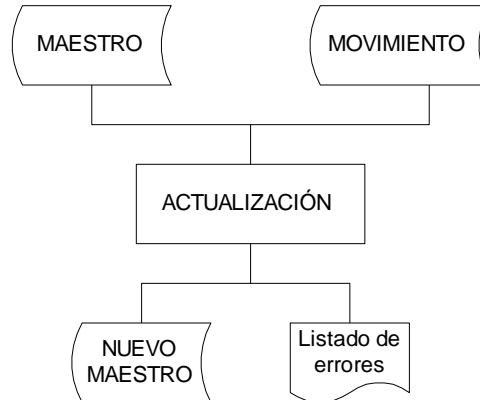
Tipos de Mezcla:

	DIRECTA	INDIRECTA
Formato de los registros de los Ficheros de Entrada	igual	distinto
Formato de los registros del Fichero de Salida	igual al de los ficheros de entrada	igual a alguno de los ficheros de entrada o una combinación de estos
Cantidad de registros del Fichero de Salida	es igual a la sumatoria de las cantidades de los registros de los ficheros de entrada: $\# \text{ As} = \sum_{i=1}^n \# \text{ A}_i$	no es posible predecir

- ACTUALIZACIÓN

Ficheros de entrada: por lo menos dos: MAESTRO Y MOVIMIENTOS.

Ficheros de salida: como mínimo uno.



Tipos de Actualización:

POR LOTES	UNITARIA
varios registros de movimiento para un registro maestro.	un registro movimiento para un registro maestro.

Tipos de Ficheros Maestros:

- Histórico
- Común o Normal

Tipos de Ficheros de Movimientos:

- Actualización General o Combinado.
- Actualización Parcial o Individual.

Tipos de Movimientos:

- Alta
- Baja
- Modificación

Algoritmo Actualización unitaria

Utilizando ciclo incluyente

ACCION ACT_UNIT ES

Algoritmo

Abrir_Archivos

Leer_Maestro

Leer_Movimiento

MIENTRAS (Clave_Mae <> High Value) o (Clave_Mov <> High Value) HACER

 SI Clave_Mae = Clave_Mov

 ENTONCES Proceso_iguales

 SINO SI Clave_Mae < Clave_Mov

 ENTONCES Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

 Leer_Maestro

 SINO Proceso_distintos

 Fsi;

FSi;

Fmientras;

Cerrar Archivos

FACCIÓN.

Acción Leer_Movimiento es

 LEER(Arch_mov, Reg_mov)

 SI FDA(Arch_mov)

 ENTONCES Clave_mov := High_value

 FSi;

FAcción.

Acción Leer_Maestro es

 LEER(Arch_mae, Reg_mae)

 SI FDA(Arch_mae)

 ENTONCES Clave_mae := High_value

 FSi;

FAcción.

Acción Proceso_iguales es

 SI Reg_mov.Cod_mov = 'ALTA'

 ENTONCES ESCRIBIR('Error alta no existe')

 Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

 SINO SI Reg_mov.Cod_Mov = 'MODIF'

 ENTONCES Proceso_modif_maestro

 Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

 SINO * eliminación lógica *

 Marcar_registro

 Reg_sal := Reg_mae

 ESCRIBIR(Arch_sal, Reg_sal)

 Fsi;

FSi;

Leer_Maestro

Leer_Movimiento

FAcción.

Acción Proceso_distintos es

```

    SI Reg_mov.Cod_Mov = 'BAJA'
    ENTONCES ESCRIBIR('Error baja no existe')
    SINO SI Reg_mov.Cod_Mov = 'MODIF'
    ENTONCES ESCRIBIR('Error modificación no existe')
    SINO * Asigna campo por campo, porque Reg_sal y Reg_mov tienen distinto formato *
        Reg_sal.clave := Reg_mov.clave
        Reg_sal.campo1 := Reg_mov.campo1
        Reg_sal.campo2 := Reg_mov.campo2
        .....
        Reg_sal.campoN := Reg_mov.campoN
        Reg_sal.Marca_baja := ''
        ESCRIBIR(Arch_sal, Reg_sal)

```

Fsi;

Fsi;

Leer_Movimiento

FAcción.

Acción Proceso_modif_maestro es

```

    Si Reg_Mov.campo1 <> ''
    entonces Reg_mae.campo1 := Reg_mov.campo1
    fsi;
    Si Reg_Mov.campo2 <> ''
    entonces Reg_mae.campo2 := Reg_mov.campo2
    fsi;
    Si Reg_Mov.campo3 <> ''
    entonces Reg_mae.campo3 := Reg_mov.campo3
    fsi;
    * ... y así sucesivamente para todos los campos del registro... *
    FAcción.

```

Acción Marcar_registro es

```

    Reg_mae.Marca_baja := '*' * en vez de asterisco, se puede asignar la fecha del día,
                             o cualquier otro dato, según el problema *

```

FAcción.

Algoritmo Actualización por lotes

Utilizando ciclo incluyente, se considera un archivo de movimientos consistente y se realizan bajas lógicas.

Consideraciones sobre el ambiente:

formato1 = registro Clave = registro campo1: campo2: freg; campo3:..... campoN:.... Marca_baja:..... freg;	formato2 = registro Clave = registro campo1: campo2: freg; campo3:..... campoN:.... cod_mov : (A,B,M) freg;
---	---

Reg_mae, Reg_sal, Aux: formato1;
 Reg_Mov: formato2;

ACCION ACT_POR_L ES

Algoritmo

```

Abrir_Archivos
Leer_Maestro
Leer_Movimiento
MIENTRAS (Clave_Mae <> High Value) o (Clave_Mov <> High Value) HACER
  SI Clave_Mae < Clave_Mov                                * Maestro sin Movimiento *
    ENTONCES Reg_sal := Reg_mae
    ESCRIBIR(Arch_sal, Reg_sal)
    Leer_Maestro
  SINO SI Clave_Mae = Clave_Mov                            * Maestro con Movimiento *
    ENTONCES Aux := Reg_mae
    MIENTRAS (Clave_Mae = Clave_Mov) HACER
      Proceso_Movim
      Leer_Movimiento
      FMientras
      Reg_sal := Aux
      ESCRIBIR(Arch_sal, Reg_sal)
      Leer_Maestro
    SINO * Movimiento sin Maestro ~ 1 Alta y 0-1 Modific. y/o Bajas *
      * Asigna campo por campo, porque Aux y Reg_mov
      tienen distinto formato *
      Aux.clave := Reg_mov.clave
      Aux.campo1 := Reg_mov.campo3
      Aux.campo2 := Reg_mov.campo4
      .....
      Aux.campoN := Reg_mov.campoN
      Aux.Marca_baja := ''
      Leer_Movimiento
      MIENTRAS (Clave_Aux = Clave_Mov) HACER
        Proceso_Movim
        Leer_Movimiento
      FMientras
      Reg_sal := Aux
      ESCRIBIR(Arch_sal, Reg_sal)
  
```

```

        FSI;
    Fmientras;

    Cerrar(Arch_sal)
    FACCIÓN.

Acción Leer_Movimiento es
    LEER(Arch_mov, Reg_mov)
    SI FDA(Arch_mov)
        ENTONCES Clave_mov := High_value
    FSI;
    FAcción.

Acción Leer_Maestro es
    LEER(Arch_mae, Reg_mae)
    SI FDA(Arch_mae)
        ENTONCES Clave_mae := High_value
    FSI;
    FAcción.

Acción Proceso_Movim es
    * Comparar la información del Registro de Movimientos y, de acuerdo a los valores que
      traigan, y alterar los contenidos del Registro Auxiliar (Aux). *
    SI Cod_Mov = 'M' * Modificación *
        ENTONCES Proceso_modif_maestro
            SINO SI Cod_Mov = 'B' * eliminación lógica *
                ENTONCES Marcar_registro
            Fsi;
    Fsi;

    FAcción.

Acción Proceso_modif_maestro es
    Si Reg_Mov.campo3 <> ' ' ** OJO! campo1 y campo2 NO, porque son claves **
        entonces Aux.campo3 := Reg_mov.campo3
    fsi;
    Si Reg_Mov.campo4 <> ' '
        entonces Aux.campo4 := Reg_mov.campo4
    fsi;
    .....
    * ... y así sucesivamente para todos los campos del registro... *
    FAcción.

Acción Marcar_registro es
    Aux.Marca_baja:= '*' * en vez de asterisco, se puede asignar la fecha del día,
                        o cualquier otro dato, según el problema *
    FAcción.

```

FICHEROS INDEXADOSActualización Indexada Unitaria

- Archivo maestro indexado por clave
- Se ingresan movimientos por terminal

ACCION ACT_INDEX_UNIT ES

Ambiente

Valido= ("A", "B", "M")

Arch_mae: archivo de reg_mae indexado por clave_mae.

Cod_mov: AN(1);

Algoritmo

Abrir E/S (arch_mae)

Escribir ('Por favor ingrese la clave a procesar y el código de movimiento (A: incorporaciones, B: bajas, M: modificaciones) Para finalizar ingrese cualquier otra letra.')

Leer(clave, cod_mov)

MIENTRAS cod_mov en valido hacer

Reg_mae.clave_mae := clave

Leer (arch_mae, Reg_mae)

SI No existe

ENTONCES SI cod_mov = 'B'

ENTONCES ESCRIBIR('Error baja no existe')

SINO SI cod_mov = 'M'

ENTONCES ESCRIBIR('Error modificación no existe')

SINO ** Ingresar por teclado los datos correspondientes a la nueva clave **

Leer(Reg_mae.campo1)

Leer(Reg_mae.campo2)

.....

Leer(Reg_mae.campon)

ESCRIBIR(arch_mae, Reg_mae)

FSI;

FSI;

SINO SI cod_mov = 'A'

ENTONCES ESCRIBIR('Error clave existe, alta no es posible')

SINO SI cod_mov = 'M'

ENTONCES ** Ingresar por teclado los datos que se desean modificar **

Leer(campo1)

SI campo1 <> ""

ENTONCES Reg_mae.campo1 := campo1

FSI

Leer(campo2)

SI campo2 <> ""

ENTONCES Reg_mae.campo2 := campo2

FSI

.....

Leer(campon)

SI campon <> ""

```
        ENTONCES Reg_mae.campoN := campoN
        FSI
        RE-ESCRIBIR(arch_mae, Reg_mae)

    SINO  **{baja lógica}**
        Marcar Registro Maestro
        RE-ESCRIBIR(arch_mae, Reg_mae)

        ** o {baja física}**
        BORRAR(arch_mae, Reg_mae)

    FSI;
    FSI;
    FSI;
    Escribir ('Por favor ingrese la clave a procesar y el código de movimiento (A: incorporaciones,
    B: bajas, M: modificaciones) Para finalizar ingrese cualquier otra letra.')

    Leer(clave, cod_mov)

Fin_Mientras

Cerrar(Arch_mae)

FACCIÓN.
```

Ejercicios Complementarios T.P. Nro. 2**Secuencias de Texto**

1. Dada una secuencia TEXTO construir un algoritmo que cuente e informe por oración cuántas palabras terminan con la sílaba "LA".
2. Dada una secuencia de caracteres que contiene un texto compuesto por oraciones, crear una secuencia de salida que contenga sólo las palabras que comiencen con la sílaba "MA". Informar para cada oración cuántas palabras cumplen con la condición.
3. Dada una secuencia de caracteres que contiene un texto compuesto por oraciones; crear una secuencia de salida que contenga solo las palabras con 4 letras, informar para cada oración cuantas palabras cumplen con la condición. El fin de cada oración esta dado por un "."
4. Dada una secuencia de texto de entrada, desarrollar un algoritmo que produzca una secuencia oración de salida; la que contendrá palabras numéricas formadas con los dígitos impares que contenga la palabra correspondiente de entrada . Ej. Si la entrada viniera "A125EB%" en la salida debería ir "15". Al final del proceso informar cuál es el promedio de palabras por oración encontradas en la secuencia de entrada.
5. Escribir un algoritmo que: dado un fichero de texto que contiene letras y números, permita distinguir palabras de enteros de hasta 5 cifras y desglosarlos en dos secuencias que contengan respectivamente las palabras pares y las impares.
6. Escribir un algoritmo que permita desglosar una secuencia que contiene monotonías crecientes (series de elementos consecutivos ordenados crecientemente; cada serie está constituida por un elemento como mínimo), en dos secuencias de salida, de tal modo que tales monotonías sean copiadas alternativamente en una de las secuencias resultado o en otra.
7. Dada una secuencia texto de entrada, desarrollar un algoritmo que produzca una secuencia oración de salida; la que contendrá palabras numéricas formadas con los dígitos pares que contenga las palabras correspondientes de entrada. Ej.: entrada (q12w34e58) → salida (248). Al final del proceso informar cual es el promedio de palabras por oración encontradas en la secuencia de entrada.
8. Escribir un algoritmo que a partir de una secuencia de palabras dada, genere dos secuencias de salida. Una de ellas contendrá solo las palabras de la secuencia original que comienzan con vocal y la otra será una secuencia numérica en la que se almacenarán las longitudes de las palabras que cumplieron tal condición. Por final de proceso se deberá informar la cantidad de palabras de ambas secuencias de palabras y la longitud de la palabra más larga que cumplió la condición enunciada.
9. Escribir un algoritmo que a partir de una secuencia de palabras dada, genere una secuencia de igual tipo conteniendo solo las palabras de la secuencia original que NO comienzan con vocal . Por final de proceso se deberá informar la cantidad de palabras de ambas secuencias de palabras y el porcentaje de palabras que no cumplieron con la condición enunciada.
10. Escribir un algoritmo que permita efectuar la operación de diferencia entre dos secuencias (ordenadas crecientes), esto es, que genere el conjunto de los elementos no comunes a las dos secuencias.
11. Escribir un algoritmo que dado dos ficheros de texto (caracter, palabra, oraciones) denominadas TEX1 y TEX2 genere otro fichero oración, mediante la composición de palabras en posiciones pares dentro de las oraciones del TEX1 y palabras en posiciones impares dentro del TEX2. Al final del proceso emitir los siguientes contadores :
 - a) Cantidad de palabras del archivo de salida.
 - b) Promedio de palabras por oración de TEX1 Y TEX2.
12. Dada una secuencia de caracteres con datos de personas con el siguiente formato:

Nombre|Domicilio|DNI (8 caracteres)Región (1-2-3)Nombre|Domicilio|DNI (8 caracteres)Región (1-2-3).....*
(marca de fin)

DNI, Región y Nombre de la siguiente persona no están separados por ningún carácter, Región tiene un solo carácter.

Generar una secuencia de caracteres de salida que contenga todos los DNI que comiencen con dígito par y pertenezcan a la Región 1.

Al final informar cantidad de personas de cada región y porcentaje de las mismas sobre el total de personas.

Archivos: Corte de control

Una empresa dispone de un archivo de registros con los siguientes datos de sus empleados ordenados por planta:

Nombre	Categoría	Planta
--------	-----------	--------

La categoría está codificada del 1 al 4. La fábrica cuenta con dos plantas A y B.

A continuación se presenta el listado de empleados con su categoría y lugar de trabajo:

Nombre	Categoría	Planta	Nombre	Categoría	Planta
Perez	3	A	Casas	2	B
Gutierrez	2	A	Portales	2	B
Aranda	1	A	Morales	3	B
Molina	1	A	Arce	2	B
Salerno	1	A	Cortezs	1	B
Mendez	4	A	Milán	4	B
Corrales	4	A	Davis	1	B
Derqui	3	A	Días	1	B
Biondi	4	A	Vera	2	B
Nogueira	1	A	Basualdo	1	B

PROCESOS CON CORTE DE CONTROL

1.Una empresa dispone de un archivo de registros con datos de sus empleados

Nombre	Categoría
--------	-----------

El archivo está ordenado por categoría

Obtener el listado con el siguiente diseño:

LISTA DE PERSONAL
Categoría 1
Aranda
Molina
Salerno

Categoría 2
Gutierrez
Casas

2.Una empresa dispone de un archivo de registros con datos de sus empleados:

Reg.	Nombre	Categoría	Planta
------	--------	-----------	--------

(Tomar los datos del comienzo del trabajo)

El archivo está ordenado por categorías.

Imprimir el siguiente Listado:

LISTADO DE PERSONAL	
Categoría 1	
Nombre	Planta
Aranda	B
Molina	A
Salerno	A
-----	----
Categoría 2	
Nombre	Planta
Gutierrez	B
Casas	B
-----	----
-----	-----

3. El archivo de la empresa está ahora ordenado por planta. Los empleados cobran los siguientes sueldos según la categoría.

1- \$1.500 2- \$2.000 3- \$3.500 4- \$5.000

Imprimir el siguiente listado considerando la categoría de cada empleado:

LISTADO DE SUELDOS	
Planta A	
Nombre	Sueldo
Aranda	1500
Molina	1500
Salerno	5000
-----	-----
Planta B	
Nombre	Sueldo
Perez	3500
Gutierrez	2000
Aranda	1500
-----	-----

4. El archivo se encuentra ordenado por planta y dentro de cada planta ordenado por categoría.

Se supone en este caso que los sueldos que corresponden a cada categoría deben ser introducidos como dato por el teclado

PLANILLA DE EMPLEADOS		
Planta A		
	Categoría 1	Sueldo:-----
	Molinari	
	Salerno	

	Categoría 2	Sueldo:-----
	Arce	

Planta B		
	Categoría 1	Sueldo:-----
	Aranda	
	Davis	

5. Una escuela dispone de un archivo con los registros de sus alumnos:

Nombre	Año	División	Nota1	Nota2	Nota3
--------	-----	----------	-------	-------	-------

Año: 1 a 5

División: A, B, C, etc.

Nota1, Nota2, Nota3: Notas correspondientes a cada uno de los tres cuatrimestres (0 a 10).

Estos registros están ordenados en forma decreciente por el AÑO y dentro de cada año están ordenados en forma creciente por la división.

Obtener el siguiente listado:

LISTA DE PROMEDIOS		
5° Año		
División "A"		
	Nombre	Promedio
	-----	-----
	-----	-----
División "B"		
	Nombre	Promedio
	-----	-----
	-----	-----
4° Año		
División "A"		
	Nombre	Promedio
	-----	-----
	-----	-----
División "B"		
	Nombre	Promedio
	-----	-----
	-----	-----
	-----	-----

TOTALES Y SUBTOTALES

6. Una empresa dispone de un archivo de registros con los siguientes datos de los pasajeros que volaron en el día:

Nombre	Nro. De vuelo
--------	---------------

El archivo está ordenado por n° de vuelo. Imprimir la cantidad De pasajeros que viajaron en cada vuelo y la cantidad total de pasajeros que viajaron ese día por la compañía:

CANTIDAD DE PASAJEROS	
Vuelo 067:	83
Vuelo 123:	123
Vuelo 451:	321
-----	-----
-----	-----
Total:	1432

7. Reconsidere el problema 1. Obtener la salida:

CANTIDAD DE EMPLEADOS	
Categoría 1	-----
Categoría 2	-----
Categoría 3	-----
Categoría 4	-----
Total	-----

8. Reconsidere el problema 3.

La compañía debe enviar a cada planta el dinero para abonar los sueldos.

Para ello requiere el siguiente listado.

TOTALES DE SUELDOS POR PLANTA	
Planta A:	-----\$
Planta B:	-----\$
Planta C:	-----\$
-----	-----
Total:	-----\$

9. Reconsidere el problema 4. La compañía desea conocer los totales de los sueldos por planta discriminados por categoría, para ello se requiere el siguiente listado:

TOTALES DE SUELDO POR PLANTA Y CATEGORÍA	
Planta A	
Categoría 1:	-----\$
Categoría 2:	-----\$
-----	-----
Total planta A	-----\$
Planta B	
Categoría 1:	-----\$
Categoría 2:	-----\$
-----	-----
Total planta B	-----\$
-----	-----
-----	-----
Total de Sueldos:	-----\$

EMISIÓN DE INFORMES POR CORTE DE CONTROL

Problemas integrados

10.Una empresa con varias sucursales que dispone de un archivo de registros de sus empleados con los siguientes datos:

CodSuc	CodEmpl	HorasDía
--------	---------	----------

CodSuc: código de sucursal
 CodEmpl: código de empleado
 HorasDía: horas trabajadas en el día

Teniendo en cuenta que se abona un premio a aquellos empleados que hayan trabajado durante el mes más de 20 días y que cumplan simultáneamente con:

- No menos de 180 horas trabajadas, el premio será de \$200.
- Más de 170 horas y menos de 180 horas, el premio será de \$150.

Realizar el diagrama que leyendo el archivo de datos determine e imprima:

- Código de sucursal, Código de empleado y premio de los empleados que cumplan con las disposiciones anteriores,
- Importe total de los premios que deben pagar cada sucursal,
- Importe total que la empresa pagará.

11. Una empresa que vende lubricantes registra en un archivo de registros:

NroCli	NroLubr	CantVen	PU
--------	---------	---------	----

NroCli: N° de Cliente
 NroLubr: N° de Lubricantes
 CantVen: Cantidad vendida
 PU: Precio por Unidad

Toda esta información está ordenada por N° de Cliente.
 Determinar e imprimir:

- El N° de Cliente que compró más en pesos
- Cuantos Clientes compraron más de 100 latas en total
- Cantidad de latas vendidas de lubricante 31667 entre todos los clientes.

12. Se realiza una encuesta en distintas provincias. Por cada persona que responde se genera un registro con:

- Código de la provincia
- Edad
- Código de los estudios cursados: 1. Primarios; 2. Secundarios; 3. Universitarios; 4. Terciarios; 5. Ninguno

La información se ordena por provincia y se graba en un archivo.

Se pide:

- Cantidad de personas con estudios primarios
- Cantidad de personas con estudios secundarios
- Para cada provincia:
 1. Código de la provincia
 2. Total de encuestados
- Código de la provincia que tuvo mayor cantidad de personas sin ningún estudio.

13. En una empresa se preparan registros por cada venta individual según se indica:

Mes	Importe
-----	---------

Estos registros, están grabados en un archivo, y están ordenados por mes. Se pide calcular:

- Total y promedio de ventas de cada mes
- Promedio anual de ventas
- Total anual de ventas

Imprimir además la siguiente planilla con los títulos que se indican:

EMPRESA NNN		
Mes	Total	Promedio
Xx	Xx	Xxx.xx
Xx	Xx	Xxx.xx
Promedio anual de ventas		xxx.xx
Total anual		xxxx.xx

14. En una empresa se emiten comprobantes por cada venta que se realiza.

Cada comprobante da origen a un registro de un archivo con la siguiente información:

Año	Mes	Día	Importe
-----	-----	-----	---------

A fin de año se procesa el archivo que está ordenado secuencialmente por mes y por día.

Se pide la planilla siguiente:

EMPRESA NNN				
Año	Mes	Día	Importe	
2000	01	01	xxx.xx	
2000	01	01	xxx.xx	
Total diario				
2000	01	01	xxx.xx	

(días restantes)				

Total Mensual			xxx.xx	

(meses restantes)				

Total Anual			xxxx.xx	

Cuando se produce el corte;

Por día : dar el total diario.

Por cambio de mes : dar el total mensual.

Al final : total diario, mensual y anual.

15. Un banco tiene varias sucursales, cada una de las cuales tiene un gran número de cuentas corrientes. Al final de cada mes se desea conocer el capital existente en:

- Cada cuenta corriente
- Cada sucursal
- Total en el banco.

Para ello se prepara un archivo de registros clasificado por sucursal y por número de cuenta corriente:

Sucursal	CtaCte	Importe	Código
----------	--------	---------	--------

Puede haber más de un registro con el mismo número de cuenta corriente.

CODIGO puede tomar dos valores: D - depósito, E - Extracción.

La planilla que se pide es la siguiente

EMPRESA NNN			
SUCURSAL	CUENTA CORRIENTE	IMPORTE	CODIGO
01	1007	1000	D
01	1007	2000	D
Total Cta. Cte.			3000
01	2054	2000	E
Total Cta.Cte.			-2000

(demás Ctas. Ctes.)			

Total sucursal			xxxx

(demás sucursales)			

Total banco			xxx

16. Una empresa de venta de accesorios de automóviles comercializa 100 artículos como máximo. Para controlar sus exigencias y sus ventas genera un archivo con la siguiente información por cada artículo: Un primer registro por cada artículo con:

NroArt	Cant
--------	------

NROART: número de artículo - CANT: cantidad de unidades en existencia

A continuación un lote de registros, uno por cada venta, con:

NroArt	Cant
--------	------

NROART: número de artículo - CANT: cantidad de unidades de cada venta.

Habrà, por lo menos, un artículo con sus ventas.
Puede ser que de algún artículo no haya ventas.

Imprimir:

- Saldo final de cada artículo y su número.
- Al final del proceso indicar el N° de artículo
- Si de algún artículo se vende más de lo que hay en existencia detener el proceso con un texto aclaratorio (sin imprimir el mínimo).
- Indicar si se procesaron más de 100 artículos
- Calcular el promedio de las cantidades vendidas por artículo exclusivamente para los artículos que tienen ventas.

17. Se conocen

- Las tarifas particular y comercial del KW.
- El consumo mínimo de ambas categorías

Además se dispone de un archivo de usuarios en el que cada registro contiene

NroZona	Nombre	Calle	Nro	TipoUsuario	Consumo
---------	--------	-------	-----	-------------	---------

Los usuarios están ordenados por zona "P" o "C"

Se pide:

- Listar para cada zona de aquellos usuarios que superen el consumo mínimo lo siguiente:
 - Tipo de Usuario
 - Nombre
 - Calle y Número
 - Consumo e importe
- Informar para cada zona:
 - Nombre y consumo promedio de los usuarios particulares
 - La dirección del usuario comercial cuyo consumo haya sido el máximo registrado en esa zona.
 - El total facturado y la cant. De usuarios de cada categoría que abonen el mínimo.
- Informar además al final del listado:
 - Zona con mayor cantidad de usuarios comerciales
 - Zona que registra el menor consumo.

Archivos: Actualización

1. Dado un Archivo Maestro secuencial con datos de usuarios de una empresa proveedora de energía eléctrica con la siguiente estructura:

Nro_U	Nombre	Domicilio	Tipo_U	Consumo prom
-------	--------	-----------	--------	--------------

(a - b - c)

el cual se actualiza, mes a mes, a partir de las novedades (altas, modificaciones y bajas) almacenadas en otro archivo secuencial de Movimientos. Puede haber más de una novedad para cada usuario y las bajas son físicas.

a) Diseñe el ambiente, con todas las estructuras de datos necesarias para realizar el proceso de actualización. Tenga en cuenta que los nombres de variables se ajusten al algoritmo del punto b).

b) Dado los siguientes algoritmos, marque los errores que encuentre y desarrolle las subacciones usadas en el mismo que están subrayadas.

b.1)

Algoritmo

Abrir_Archivos
Leer_Maestro
Leer_Movimiento

MIENTRAS (Reg_Mae.Nro_U <> High Value) o (Reg_Mov. Nro_U <> High Value) HACER

SI Reg_Mae. Nro_U < Reg_Mov. Nro_U

ENTONCES

ESCRIBIR(Arch_sal, Reg_sal)

Leer_Maestro

SINO SI Reg_Mae.Nro_U = Reg_Mov. Nro_U

ENTONCES

MIENTRAS Reg_Mae.Nro_U = Reg_Mov.Nro_U HACER

Proceso_Movim

Leer_Movimiento

FMientras

Reg_sal := Aux

ESCRIBIR(Arch_sal, Reg_sal)

Leer_Maestro

SINO

Aux.Nro_U := Reg_Mov.Nro_U; Aux.Nombre := Reg_Mov.Nombre;

Aux.Domicilio := Reg_Mov.Domicilio; Aux.Tipo_U := Reg_Mov.Tipo_U;

Aux.Consumo_Prom := Reg_Mov.Consumo_prom;

Leer_Movimiento

MIENTRAS Aux = Reg_Mov.Nro_U HACER

Proceso_Movim

FMientras

Reg_sal := Aux

ESCRIBIR(Arch_sal, Reg_sal)

FSI;

FSI;

Fmientras;

Cerrar(Arch_sal)

b.2)

Algoritmo


```

Abrir_Archivos
Leer_Maestro
Leer_Movimiento
MIENTRAS (Reg_Mae.Nro_U <> High Value) o (Reg_Mov. Nro_U <> High Value) HACER
  SI Reg_Mae. Nro_U < Reg_Mov. Nro_U
    ENTONCES Reg_sal := Reg_mae
      ESCRIBIR(Arch_sal, Reg_sal)
  SINO SI Reg_Mae.Nro_U = Reg_Mov. Nro_U
    ENTONCES Aux := Reg_mae
      MIENTRAS Reg_Mae.Nro_U = Reg_Mov HACER
        Proceso_Movim
        FMientras
        Reg_sal := Aux
        ESCRIBIR(Arch_sal, Reg_sal)
        Leer_Maestro
      SINO
        Leer_Movimiento
        MIENTRAS Aux.Nro_U = Reg_Mov.Nro_U HACER
          Proceso_Movim
          Leer_Movimiento
          FMientras
          Reg_sal := Aux
          ESCRIBIR(Arch_sal, Reg_sal)
        FSI;
      FSI;
    FMientras;
  Cerrar(Arch_sal)

```

b.3)

Algoritmo

```

Abrir_Archivos
Leer_Maestro
Leer_Movimiento
MIENTRAS (.....<> High Value) o (Reg_Mov. Nro_U <> High Value) HACER
  SI Reg_Mae. Nro_U < Reg_Mov. Nro_U
    ENTONCES Reg_sal := Reg_mae
      RE-ESCRIBIR(Arch_sal, Reg_sal)
      Leer_Maestro
  SINO SI Reg_Mae.Nro_U = Reg_Mov. Nro_U
    ENTONCES Aux := Reg_mae
      MIENTRAS Reg_Mae.Nro_U = Reg_Mov.nro_U HACER
        Proceso_Movim
        Leer_Movimiento
        FMientras
        Reg_sal := Aux
        ESCRIBIR(Arch_sal, Reg_sal)
      SINO
        Aux.Nro_U := Reg_Mov.Nro_U; Aux.Nombre := Reg_Mov.Nombre;
        Aux.Domicilio := Reg_Mov.Domicilio; Aux.Tipo_U := Reg_Mov.Tipo_U;
        Aux.Consumo_Prom := Reg_Mov.Consumo_prom;

        Leer_Movimiento

```

```
MIENTRAS Aux = Reg_Mov.Nro_U HACER
    Proceso_Movim
    Leer_Movimiento
FMientras
Reg_sal := Aux
ESCRIBIR(....., Reg_sal)
    Leer_Maestro
    FSI;
FSI;
Fmientras;
Cerrar(Arch_sal)
```

Mezcla

1. Suponer que los datos del ejercicio 2.01 se encuentran en 3 secuencias por separado, correspondientes a ISI, IQ y IEM, todas ordenadas por Nro. de Legajo. Generar un algoritmo que permita fusionar estas en una sola secuencia, respetando el orden y la estructura.

Algoritmos y
Estructuras de
Datos

Práctico 3

Tratamiento de Arreglos
Procesos Estadísticos
Registros de Longitud Variable por Ocurrencia
Complemento Teórico
Ejercicios Complementarios

2025

TRABAJO PRACTICO Nº 3**1. Tratamiento de Arreglos**

01. Para cada una de las consignas siguientes, genere un algoritmo que permita solucionarla (3 algoritmos) usando un arreglo de 100 números enteros:

- Almacenar 100 números.
- Localizar el número de mayor valor y el de menor valor, informar sus valores y sus posiciones.
- Contar y sumar todos los números pares.

02. Genere un único algoritmo que resuelva las 3 consignas del ejercicio anterior.

03. Dados un vector de reales a de n elementos, con $n \in \mathbb{N}_0$, y un número real $x \geq 0$, calcular el valor y tal que:

$$y = \sum_{i=1}^n (a_i * x^i)$$

04. Considerando un arreglo de 50 números enteros, confeccione un algoritmo para resolver las siguientes consignas:

- Modificar el arreglo dado, de modo que todos sus elementos sean múltiplos de 3.
- Crear otro arreglo que contenga los números que no cumplieron la condición.
- Informar cuántos números cumplieron la condición.

05. Dados 2 vectores: A: arreglo [1 .. 30] de reales y B: arreglo [1 .. 30] de reales, ambos ordenados de forma creciente, escribir un algoritmo que realice la mezcla de ambos para obtener otro vector C: arreglo [1 .. 60] de reales, también ordenado de forma creciente.

06. Escribir un algoritmo que permita cargar un arreglo de N nombres, considerando que cada nombre debe tener entre 1 y 10 caracteres.

07. Escribir un algoritmo que permita localizar un nombre en un arreglo de N nombres, ordenados alfabéticamente. Cada nombre puede tener, como máximo, 10 caracteres. Escriba por lo menos dos algoritmos que permitan solucionar el problema; especifique cuál de las formas considera más eficiente y por qué.

08. Repita el ejercicio anterior, pero suponiendo que se precisa localizar todos los nombres que comienzan con una letra dada.

09. Se posee un arreglo de 200 libros con el siguiente formato:

NRO_LIBRO | TITULO | AUTOR | CANT_HOJAS ; ordenado por AUTOR y se presentan las siguientes premisas:

- Se necesita saber que libros se poseen de "Nicklaus Wirth".
- Se necesita saber en qué posición se encuentra "Algoritmos + Estructuras de Datos=Programa".
- Se necesita saber cual es el libro de "Nicklaus Wirth" de mayor volumen.

10. Dado un arreglo de 50 elementos, cada uno de los cuales tiene los siguientes datos: Código de localidad y Lluvia caída en un año. Escribir un algoritmo que permita saber dada una localidad, cuanto llovió en el año. Aplicar el método más adecuado considerando que el arreglo esta ordenado por Código de localidad.

11. Dado un arreglo de 100 elementos, que contiene la siguiente información sobre videos: Título de la película, Nombre del Director, Categoría de película, Cantidad de personas que la vieron, Alquilado (si/no); y está ordenado por el Título de la película, diseñe un algoritmo que, ingresando una categoría, liste todas las películas que pertenecen a dicha categoría.

12. A partir del arreglo de videos descrito en el ejercicio anterior, diseñe un algoritmo que permita atender un pedido de alquiler, para lo cual debe verificar si es posible o no y, cuando corresponda, actualizar la cantidad de personas que vieron dicha película.

13. Se precisa ordenar un arreglo de N alumnos de mayor a menor, de acuerdo a la cantidad de materias aprobadas. Cada elemento del arreglo contiene Nro. de Legajo y Cantidad de materias aprobadas. Escriba por lo menos dos algoritmos que permitan solucionar el problema; especifique cuál de las formas considera más eficiente y por qué.

14. El mes que viene se realizará en Buenos Aires el desfile “Alta Moda 2005”, el cual reúne a los diseñadores más reconocidos del país. Para organizar las pasadas, se dispone de un arreglo por diseñador, con la siguiente información: nombre de el/la modelo y altura. La directora del evento necesita que se imprima cada lista ordenada de acuerdo a la altura de el/la modelo. Escribir un algoritmo que permita ingresar los datos de cada diseñador e imprimirlos de acuerdo a lo solicitado.

15. Se precisa ordenar un arreglo de enteros de menor a mayor, eliminando los números repetidos.

16. Una empresa que comercializa una cierta cantidad de artículos diferentes desea confeccionar un ranking de ventas de los mismos, a partir de una secuencia que contiene:
NRO_ARTICULO | TIPO | CANT_VENDIDA, ordenado por artículo. Escribir un algoritmo que emita el ranking deseado en orden decreciente por cantidad.

17. Se precisa generar una secuencia con los datos de los 10 videos más vistos de una categoría, a partir de una secuencia de entrada de 200 registros que contiene el Título de la película, Nombre del Director, Categoría de película, Cantidad de personas que la vieron y que está ordenada por el Título de la película.

18. Se precisa diseñar una agenda electrónica, donde se archivará el nombre, la dirección y el teléfono de hasta 50 personas. Diseñe un algoritmo que permita efectuar consultas, modificaciones, eliminaciones de los datos de una persona y agregados de nuevas personas (sólo será posible incorporar una persona si hay menos de 50 archivadas en la agenda). Los datos se hallan almacenados en una secuencia. Considere que la agenda siempre debe mantenerse ordenada alfabéticamente.

19. Escribir un algoritmo que emita cuál es y dónde está ubicado el mayor elemento de cada fila de una matriz, e imprima un mensaje si todos los mayores se encuentran en la misma columna.

20. Dadas dos matrices A y B, cuadradas, de 5 x 5, con números enteros, cargar una matriz C, de 5 x 5 teniendo en cuenta las siguientes condiciones: la diagonal principal completar con ceros, en las posiciones que están por encima de la diagonal principal, copiar los correspondientes elementos de la matriz A y en las posiciones que están por debajo de la diagonal principal, copiar los elementos correspondientes de la matriz B.

21. Dada una matriz de 6 x 6 de enteros, cuya última fila y columna contienen ceros, calcular la suma de cada fila y guardar en la última celda de la misma; y la suma de cada columna y guardar en la última celda de la misma. Calcular también el total general y guardar en la posición (6,6).

22. Dada una matriz cuadrada de 5 x 5 de números, sumar filas y columnas y guardar en una matriz de 2 x 5, de modo que la fila 1 contenga la suma de cada fila y la fila 2, la suma de cada columna)

23. En un sector de un hospital, donde se encuentran internados 50 pacientes, se toma la temperatura de cada paciente 4 veces al día durante una semana.

Lectura	Dom	Lun	Mar	Mie	Jue	Vie	Sab
1	XX	XX	XX	XX	XX	XX	XX
2	XX	XX	XX	XX	XX	XX	XX
3	XX	XX	XX	XX	XX	XX	XX
4	XX	XX	XX	XX	XX	XX	XX

Se dispone de un arreglo con la información recopilada de todos los pacientes.

Construir un algoritmo que:

- Liste por pantalla las temperaturas máxima y mínima de cada paciente, indicando el día y lectura en la que ocurrieron.
- Genere un nuevo arreglo que contenga la temperatura promedio por día de cada paciente.

24. Se cuenta con información acerca de los cajeros de un Supermercado, el cual se halla estructurado en 10 cajas registradoras. Dicha información está registrada en una secuencia que contiene Apellido y Nombre del empleado, número de caja que tiene asignada, importe facturado y horario de facturación; la secuencia está ordenada alfabéticamente por Apellido y Nombre. Se solicita una estadística de los importes facturados, discriminado por número de caja y franja de horas y además los montos totales, según el siguiente formato:

Cajas	Franja de horas					Total x Cajas
	8-10	10-12	12-16	16-18	18-20	
1
.....
10
Total x horas

25. Una Fábrica que posee 4 plantas de producción en nuestro país y compra materia prima a 3 proveedores distintos, desea realizar un control de los montos totales correspondientes a compras realizadas en cada planta durante el último año, discriminados por proveedor y por mes.

Para ello dispone de un archivo con los datos de las facturas correspondientes.

(Aclaración: el archivo no está ordenado por ningún criterio)

COMPRAS

Nro_Factura	Proveedor (A, B, C)	Fecha (dd/mm/aaaa)	Nro_planta (1..4)	Importe
-------------	------------------------	-----------------------	----------------------	---------

26. Se dispone de un archivo secuencial de facturas de una empresa de energía eléctrica, correspondientes a un año, con el siguiente formato:

NroFact	NroUsuario	Zona	Fecha	Consumo (en Kw.)
---------	------------	------	-------	------------------

Se desea obtener un cuadro estadístico que informe los consumos y los importes totales facturados por mes discriminados por zona, con el siguiente formato:

	Zona A	Zona B	Zona C	Zona D	Total por Mes
Enero					
Febrero					
....					
Diciembre					
Total por Zona					Total General

La ciudad está dividida en 4 Zonas (A,B,C,D) , la tarifa por Kw. que cobra la empresa en cada una de ellas es: Zona A: \$ 0,05, Zona B: \$ 0,07, Zona C: \$ 0,09 y Zona D: \$ 0,13.

27. Se desea efectuar una estadística de ventas. Se cuenta para ello con una secuencia de las facturas emitidas, las cuales son identificadas por un Número; dicha secuencia contiene información relativa al cliente: su Número y Zona a la cual pertenece, como así también el Tipo de mercadería entregada, la Cantidad de unidades, el total gravado, el total exento de IVA y el valor del IVA. Construya un algoritmo que emita por zona, y dentro de la zona por tipo de mercadería el total de unidades vendidas, el total gravado, el total exento y el total de IVA, y además un total general con la misma información. Hay 9 zonas y 4 tipos de mercadería.

28. La Municipalidad de Resistencia desea obtener una estadística de los valores (en \$) de los terrenos de la ciudad, discriminados por zona y ubicación dentro de la manzana (en esquina, interna, etc.), y los totales por zona y ubicación.

El valor de cada final de cada terreno es igual a:

$$\text{Valor del Terreno} = \text{Superficie terreno (en M2)} * \text{valor del M2} * \text{coeficiente de incremento}$$

Para ello cuenta con la siguiente información:

a) un archivo de los terrenos con los siguientes datos:

Nro_terreno	Zona	Ubicación	Superficie_terreno (en M ²)
-------------	------	-----------	--

Zona: codificadas de A a F

Ubicación: codificada de 1 a 10

b) un arreglo V que contiene los valores del M² por zona.

29. Dada una secuencia de texto que contiene información sobre mascotas, se desea generar un archivo de mascotas, teniendo en cuenta lo siguiente:

En el texto vienen los siguientes datos: nombre de la mascota, encerrado ente asteriscos (*), inmediatamente a continuación una palabra de tres letras que identifica el tipo de documento y separado por un blanco, 8 posiciones numéricas que corresponden al número de documento del dueño, seguidas por un símbolo numeral (#) y, a continuación, la dirección de la persona, finalizando con un signo de fin de interrogación (?).

El final del texto está indicado por un símbolo de admiración (!).

Se desconoce la longitud de la dirección, así como la del nombre de la mascota.

El formato del registro de salida es

Tipo documento	Numero	Nombre_mascota	Dirección
N(1)	An(8)	An(30)	An(40)

Si la sigla es DNI, el tipo es 0, si dice LCE el tipo es 1, si dice LEN el tipo es 2, para PAS es 3, para cualquier otra cadena de letras el tipo es 4.

Al final del proceso informar cuantos registros se grabaron en el archivo de salida.

30. Teniendo en cuenta el ejercicio 1.20 del TP2: “Se posee 2 secuencias (S1 y S2) con las cuales se desea generar una nueva secuencia (SAL) donde se intercalen las palabras de las secuencias de entrada, de la siguiente manera: copiar de S1 aquellas palabras que empiezan y terminan con la misma letra y de S2 aquellas palabras que posean al menos un dígito numérico y además estén en posición par.”, escribir un algoritmo que lo resuelva, considerando que las palabras tienen como máximo 10 caracteres.

COMPLEMENTOS TEÓRICOS

Algoritmos de Procesos con Arreglos:

BÚSQUEDA**Búsqueda lineal**Algoritmo

Leer(Busco)

Para $i := 1$ a N hacer

Si $A[i] = \text{Busco}$ entonces

 Escribir ("El elemento", Busco, "está en la posición", i)

Fsi;

Fpara;

BÚSQUEDA LINEAL CON CENTINELAAlgoritmo1

Leer(Busco) $i := 1$

Mientras ($i < N$) y $A[i] \neq \text{Busco}$ hacer

$i := i + 1$

Fmientras;

Si $A[i] = \text{Busco}$ entonces

 Escribir ("El elemento", Busco, "se encontró en la posición", i)

sino

 Escribir ("No encontrado")

Fsi;

Algoritmo2

Leer(Busco); $i := 1$; Encuentro := Falso

Mientras ($i < N$) y No Encuentro hacer

 Si $A[i] = \text{Busco}$ entonces

 Encuentro := Verdadero

 Posición := i

Fsi;

$i := i + 1$

Fmientras;

Si Encuentro entonces

 Escribir ("El elemento", Busco, "se encontró en la posición", posición)

sino

 Escribir ("No encontrado")

Fsi;

Búsqueda binariaAlgoritmo

Leer(Busco); $iz := 1$; $de := n$;

$cen := (iz + de) \text{ div } 2$; *** o $cen := \text{trunc}((iz + de)/2)$;

Mientras ($iz < de$) y ($\text{nom}[cen] \neq \text{Busco}$) hacer

 Si ($\text{nom}[cen] > \text{Busco}$)

 entonces $de := cen - 1$

 sino Si ($\text{nom}[cen] < \text{Busco}$)

 entonces $iz := cen + 1$;

 Fsi;

Fsi;


```

    cen:=trunc (iz+de) div 2;
  Fmientras;
  Si (nom[cen] = Busco)
    entonces Escribir ('El elemento buscado está en la posición ',cen)
    sino Escribir ('El elemento buscado no está en el arreglo');
  Fsi;

```

ORDENAMIENTO

Inserción directa ** Ordena de menor a mayor **

ALGORITMO

```

  Cargar;
  Para i := 2 a n hacer
    x:=a[i]; j:=i-1;
    Mientras (j > 0) y ( x < a[j] ) hacer
      a[j+1]:=a[j];
      j:=j-1;
    Fmientras;
    a[j+1]:=x;
  Fpara;
  Facción.

```

Selección directa ** Ordena de mayor a menor **

ALGORITMO

```

  Cargar;
  Para i := 1 a n-1 hacer
    x:=a[i]; max:=i;
    Para j := i+1 a n hacer
      Si ( x < a[j] ) Entonces
        max:=j ;
        x:=a[j];
    Fsi;
    a[max]:=a[i];
    a[i]:=x;
  Fpara;
  Facción.

```

Intercambio directo - burbuja ** Ordena de mayor a menor **

ALGORITMO

```

  Cargar;
  Bandera := Falso;
  Mientras No Bandera hacer
    Bandera := Verdadero;
    Para j:=1 a n-1 hacer
      Si ( a[j] < a[j+1] ) Entonces
        x:=a[j];
        a[j]:=a[j+1];
        a[j+1]:=x;
    Bandera := Falso;
  Fsi;
  Fpara;
  FMientras;
  Facción.

```

Ejercicios Complementarios T.P. Nro. 3

1. Escribir un algoritmo que tome como entrada 2 arreglos A y B de enteros, de longitud n y devuelva como salida otro vector C, conteniendo los componentes de B para los que los correspondientes de A sean negativos. Además emitir el vector C e informar el valor y la posición del elemento mayor de C.

2. Dado un vector con los siguientes datos de libros: Autor, Título, Cantidad de hojas, en donde un autor puede tener varios libros, determinar cuál es el Título del libro con mayor cantidad de hojas, para cada Autor, generando el siguiente listado:

Autor Título Cant. de hojas

Considerar que el arreglo está ordenado por Autor y posee 50 elementos.

3. En el Laboratorio de I.S.I. se desea emitir un listado de los totales de horas ociosas de cada uno de los equipos (son 20 en total), ordenados en forma decreciente. Para ello se tiene la siguiente información almacenada en un fichero secuencial:

cód_equipo	descrip	fecha_falla			cant_hs_ociosas
		aaaa	mm	dd	

- Considerar que el fichero se encuentra ordenado por código de equipo.
- Considerar que el fichero está desordenado.

4. Se dispone de una secuencia con datos de alumnos de la Facultad: Nro. legajo, apellido y nombre, domicilio, fecha ingreso, cant. de materias aprobadas, promedio, carrera, edad.

a. Se desea emitir una estadística de cantidad de alumnos con más de 20 materias aprobadas, por carreras y grupos de edades, con el siguiente formato:

Carreras \ Edades	18 - 25	26 - 30	31 o más	Total por Carrera
ISI				
IQ				
IEM				
Totales por Grupos de Edades				

b. Se desea emitir una estadística de cantidad de alumnos con promedio mayor o igual que 7, por carreras y grupos de edades, con el mismo formato que se usó en 4.a.

5. Dada una matriz de N x N confeccionar un algoritmo para cargarla de la siguiente forma:

- Cada $a[i,j] = i + j$
- Los elementos de la última fila $a[N,j] = \text{Sumatoria de } a[i,j]$, donde i varía entre 1...(n-1)
- Los elementos de la última columna $a[i,N] = \text{Sumatoria de } a[i,j]$, donde j varía entre 1...(n-1)
- La posición $a[N,N] = \text{Sumatoria de } a[i,j]$, donde $i = j$

Ejemplo:

2	3	4	9
3	4	5	12
4	5	6	15
9	12	15	12

6. Se lleva a cabo una encuesta a fin de anticipar los posibles porcentajes de votos que obtendrán cada uno de los 4 partidos políticos de mayor peso en las elecciones. Los datos solicitados a los encuestados son: partido al que

votara, edad, sexo y partido al que votó en las elecciones anteriores. Las respuestas se tabulan de acuerdo a los siguientes criterios:

- Partidos: P1, P2, P3, P4, Otro, En Blanco, Indeciso
- Edad: 18-25, 26-35, 36-45, 46-55, 56-65, +65
- Sexo: Femenino, Masculino

Diseñe un algoritmo que, utilizando un arreglo de 4 dimensiones (partido al que votara, edad, sexo, partido al que votó), permita responder a las siguientes consultas:

- Cantidad de personas de cierta edad que votarán a un Partido dado.
- Cantidad de personas de un sexo dado que votarán a un determinado Partido.
- Cantidad de personas de cierta edad que votaron a un determinado Partido y actualmente votarán a otro Partido dado.
- Cantidad de personas de un sexo dado que votaron a un determinado Partido y actualmente votarán a otro Partido dado.

7. Diseñe un algoritmo que basándose en los datos del ejercicio anterior construya los siguientes cuadros:

- Partido actual/Partido anterior
- Partido actual/Edad
- Partido actual/Edad/Sexo

8. Con la secuencia de facturas del ejercicio 26, se precisa generar otra secuencia actualizada, a partir de una secuencia de Descuentos a clientes, la cual solamente contiene el Número correspondiente al cliente, su Razón Social, y Porcentaje de descuento. Dicho porcentaje se fija a través de una tabla que establece la relación entre el total (con IVA) de las compras realizadas en el mes, considerado por rangos, y un determinado descuento del que gozará el cliente en el mes siguiente. Se desea además contar con un listado de los descuentos establecidos para el mes. Construya un algoritmo que satisfaga los requerimientos enunciados; considere que la secuencia de facturas es consistente, es decir no hay facturas de clientes no registrados, pero pueden existir clientes que no hayan efectuado compras en el mes.

Algoritmos y
Estructuras de
Datos

Práctico 4

Funciones y Procedimientos
Tratamiento de Listas.
Ejercicios Complementarios
Complemento Teórico

2021

TRABAJO PRÁCTICO N° 4

Tratamiento de Listas

1. Diseñar un algoritmo para acceder, eliminar o insertar el k-ésimo elemento de una lista (siendo k una posición dada). Si la lista está vacía o si el valor de k esta fuera del rango del índice de la lista, invocar al procedimiento de ERROR. En cualquier otro caso, efectuar el procedimiento solicitado.

2. Se dispone de una lista simplemente encadenada de números enteros, diseñar un algoritmo que a partir de ella genere otra lista conteniendo los nodos cuyos datos terminan en cero; dichos elementos, deberán ser eliminados de la lista original. Se asume que la lista está cargada, y que el algoritmo recibe como parámetro de entrada la dirección del primer elemento.

3. Dada una lista simplemente encadenada de números diseñar un algoritmo que calcule en forma independiente: La suma de los números impares. y la suma de los números pares.

4. Se dispone de una lista simplemente encadenada cuyos registros están ordenados en forma ascendente por una clave de tipo entero; diseñar un algoritmo que invierta el orden de la lista.

5. Dada una lista simplemente encadenada que contiene datos de todas las provincias de la República Argentina: nombre, capital, cantidad total de habitantes y cantidad de analfabetos, y está ordenada en forma decreciente por número de habitantes analfabetos, generar otras tres listas que contengan el nombre, la capital y el porcentaje de analfabetos de las Provincias que respondan a las siguientes restricciones.

L1: $\leq 10\%$ analfabetos

L2: 16 a 25 % analfabetos

L3: $\geq 26\%$ analfabetos

6. En el restaurante ÑOQUIS se está pensando en una solución informática para el soporte de datos del nuevo sistema de atención a clientes. Se han decidido por LISTAS por su dinamismo en cuanto a la cantidad de elementos. Diseñe un algoritmo que realice las siguientes funciones:

- Añadir cliente al ser atendido (lista simple ordenada por Nombre del Cliente).
- Registrar su consumo (Acumular el Total Consumido en valores de montos).
- Realizar el cobro (emitir ticket con Nombre, Fecha, Número de Mesa y Total).
- Eliminar del listado de atención.

La información almacenada debe mantenerse ordenada por Nombre del cliente.

7. Genere un algoritmo que recorra una secuencia texto y genere una lista simplemente encadenada con la frecuencia de utilización de cada letra. La lista debe mantenerse ordenada alfabéticamente y al final informar cual fue la frecuencia de cada letra y cuáles fueron la de mayor y menor frecuencia.

8. Escribir un algoritmo que permita buscar, insertar o borrar un elemento identificado con una clave determinada en una lista circular simplemente encadenada.

9. Dada una lista circular de 8 elementos que contienen, cada uno, un valor numérico entero para encriptación, Encriptar (NODO)

Multiplicador (N5)	Prox (Puntero)
-----------------------	-------------------

encriptar un texto ingresado por teclado (arreglo de caracteres de 255 elementos como máximo). El texto encriptado debe almacenarse en una lista de salida simplemente encadenada y, por último mostrar por pantalla.

La encriptación se realiza según las siguientes pautas:

- 1- Dada la posición del carácter dentro del texto, buscar en la lista circular de encriptación el multiplicador correspondiente, de acuerdo a su posición lógica en la lista. Por ejemplo: como solo hay 8

multiplicadores, si se ingresa un arreglo de 10 caracteres, para encriptar los últimos dos se utilizarían los multiplicadores 1 y 2.

- 2- Se usa la función ASCII(Carácter) para convertir cada carácter del arreglo en un entero.
- 3- Luego se lo multiplican los valores obtenidos en los 2 pasos anteriores.
- 4- Por último se descompone el resultado obtenido en 3 en cada uno de sus dígitos, se los suma y se guarda en la lista de salida. Ej: si en el paso 3 me dio 123, $1+2+3=6$.

10. Se dispone de una lista circular con un conjunto de números naturales. Este conjunto está dividido en 6 grupos. En cada grupo el primer elemento indica la cantidad de números subsiguientes a él que integran dicho grupo. Se desea generar una nueva lista doblemente encadenada en la cual cada nodo contendrá el promedio de cada grupo. Dicha lista debe quedar ordenada en forma ascendente y, al final se debe informar cuál fue el máximo y el mínimo promedio de todo el conjunto de números.

11. En una empresa de servicios motorizados que NO posee una cantidad fija de integrantes, pero siempre es mayor que 1 (uno). Los datos de las personas están almacenados en un archivo (DNI, Nombre, antigüedad) ordenado por DNI. Realizar un algoritmo que permita generar una lista de asignación (usando una estructura de lista doblemente encadenada), ordenada al principio por antigüedad, y un proceso que realice dicha asignación, en cada solicitud, por turno según orden en la lista. Una vez asignado dicha persona se va al final de la lista para esperar su nueva asignación. Al final del día informar cuantas asignaciones tuvo cada integrante.

Personal			
NroPersonal (N5)	Nombre (AN50)	Asignaciones (N5)	Antigüedad (N2)

12. En el nuevo DELIVERY VAMOS-RAPIDO se está diseñando una solución informática para la atención de clientes. Se ha pensado en una LISTA por su dinamismo en cuanto a la cantidad de elementos. Diseñe un algoritmo que realice las siguientes funciones:

- Registrar su pedido en una lista Doble (se agrega en Estado P (Pendiente), y se mantiene ordenado por Nombre del Cliente).
- Realizar el envío (cambia estado del pedido de P (pendiente) a E (Enviado)).
- Eliminar del listado de atención una vez recibido el cobro (al atender el cobro del cadete que hizo el trabajo).
- Datos a almacenar por pedido: Nombre, Dirección, Teléfono y Total.

La información almacenada debe mantenerse ordenada por Nombre del cliente.

13. Dada una lista doblemente encadenada de enteros, ordenada en forma creciente, escriba un algoritmo que elimine las entradas repetidas. El algoritmo debe verificar al inicio que la lista que se pasa como argumento está ordenada; en caso afirmativo realizar la depuración y emitir un mensaje de "ÉXITO"; en caso contrario, emitir un mensaje de "ERROR".

14. Se desea ingresar una serie de números, e imprimirlos en el orden contrario al de ingreso. Diseñar un algoritmo que satisfaga tal requerimiento, utilizando la estructura de datos más apropiada.

15. Supóngase que, utilizando una lista encadenada, se implementa una cola Q (estructura "FIFO", es decir, "primero en entrar, primero en salir"). Diseñar un algoritmo que permita insertar o extraer un elemento, comprobando siempre si la cola está vacía o no.

16. Supóngase que, utilizando una lista encadenada, se implementa una pila P (estructura "LIFO", es decir, "último en entrar, primero en salir"). Diseñar un algoritmo que permita insertar o extraer un elemento, comprobando siempre si la pila está vacía o no.

17. Teniendo en cuenta el ejercicio 1.20 del TP2: "Se posee 2 secuencias (S1 y S2) con las cuales se desea generar una nueva secuencia (SAL) donde se intercalen las palabras de las secuencias de entrada, de la siguiente manera: copiar de S1 aquellas palabras que empiezan y terminan con la misma letra y de S2 aquellas palabras que posean al menos un dígito numérico y además estén en posición par.", escribir un algoritmo que lo resuelva, teniendo en cuenta que conoce la estructura de listas.

Complementos Teóricos

Escribir un algoritmo que permita buscar, insertar o borrar un elemento identificado con una clave determinada en una **lista circular simplemente encadenada**.

Acción LISTA_CIRCULAR es
AMBIENTE

elto= Registro
valor: entero;
sig: puntero a elto;

Freg;
PRIMERO: puntero a elto;
DATO: entero;
OP: 1 .. 5;
P, T : puntero a elto;

ALGORITMO

PRIMERO := NIL;

Repetir

Escribir ('* LISTA CIRCULAR *');

Escribir (' 1- BUSCAR ');

Escribir (' 2- INSERTAR');

Escribir (' 3- ELIMINAR');

Escribir (' 4- LISTAR ');

Escribir (' 5- TERMINAR');

Escribir ('* INGRESE OPCION:');

Repetir Leer (OP) Hasta que OP En [1 .. 5];

Si OP En [1 .. 3]

Ent. Escribir ('*** DATO ? ');

Leer (DATO);

Fsi;

Según OP

1: Si PRIMERO = NIL { *Buscar* }

Ent. Escribir ('ERROR, Lista Vacía')

Sino T := PRIMERO;

Mientras (*T.SIG <> PRIMERO) y (*T.VALOR <> DATO) hacer

T := *T.SIG;

Fmientras;

Si (*T.VALOR <> DATO)

Ent. Escribir ('ERROR, ', DATO, ' no está en la Lista')

Sino Escribir ('HALLADO VALOR: ', *T.VALOR);

Fsi;

Fsi;

2: NUEVO(P); { *Insertar* }

*P.VALOR:=DATO;

Si PRIMERO = NIL

Ent. { *es el primer elemento de la lista* }

*P.SIG:=P;

PRIMERO:=P;

Sino { *la lista ya contenía elto., inserta en el 2º lugar, para no modificar el puntero del último elto. }*

*P.SIG := *PRIMERO.SIG;

*PRIMERO.SIG := P;

Fsi;

3: Si PRIMERO = NIL { *Eliminar* }

```

Ent. Escribir ('ERROR, Lista Vacía')
Sino T := PRIMERO;
  Mientras (*T.SIG <> PRIMERO) y (*T.VALOR <> DATO) hacer
    ANT := T; T := *T.SIG;
  Fmientras;
  Si (*T.VALOR <> DATO)
    Ent. Escribir ('ERROR, ', DATO, ' no está en la Lista')
    Sino Si *T.SIG = T
      Ent. { Es el único elemento de la lista }
      PRIMERO:=NIL
    Sino { Lista con más de un elemento }
      Si PRIMERO = T
        Ent. { Hay que eliminar el primer elto. Ojo! Actualizar el puntero del último elto. }
        P:= PRIMERO;
        Mientras *P.SIG <> PRIMERO hacer
          P := *P.SIG;
        Fmientras;
        *P.SIG:= *T.SIG; PRIMERO:=*T.SIG;
      Sino
        *ANT.SIG := *T.SIG;
      Fsi;
    Fsi;
  DISPONER(T);
Fsi;
4: Si PRIMERO = NIL
  Ent. Escribir ('ERROR, Lista Vacía')
  Sino T := PRIMERO;
    Mientras (*T.SIG <> PRIMERO) hacer
      Escribir (*T.VALOR); T := *T.SIG;
    Fmientras;
    Escribir (*T.VALOR);
  Fsi;
Fsegún;
Hasta que OP = 5;
Facción.

```

Escribir un algoritmo que permita buscar, insertar o borrar un elemento determinado en una **lista doblemente encadenada**.

Acción LISTA_DOBLE es
 AMBIENTE

```

elto= Registro
  ant: puntero;
  valor: entero;
  sig: puntero a elto;

```

```

Freg;
PRIMERO: puntero a elto;
ULTIMO : puntero a elto;
DATO: entero;
OP: 1 .. 5;
P, T: puntero a elto;

```


ALGORITMO

```

PRIMERO := NIL; ULTIMO := NIL;
Repetir
  Escribir ('* LISTA DOBLEMENTE ENCADENADA *');
  Escribir ('  1- BUSCAR ');
  Escribir ('  2- INSERTAR');
  Escribir ('  3- ELIMINAR');
  Escribir ('  4- LISTAR ');
  Escribir ('  5- TERMINAR');
  Escribir ('* INGRESE OPCION:');
  Repetir Leer (OP) Hasta que OP En [1 .. 5];
  Si OP En [1 .. 3]
    Ent. Escribir ('*** DATO ? ');
    Leer (DATO);
  Fsi;
Según OP
  1: Si PRIMERO = NIL
    Ent. Escribir ('ERROR, Lista Vacía')
    Sino T := PRIMERO;
      Mientras (*T.SIG <> NIL) y (*T.VALOR <> DATO) hacer
        T := *T.SIG;
      Fmientras;
      Si (*T.VALOR <> DATO)
        Ent. Escribir ('ERROR, ', DATO, ' no está en la Lista')
        Sino Escribir ('HALLADO VALOR: ', *T.VALOR);
      Fsi;
    Fsi;
  2: NUEVO(P);
    *P.VALOR:=DATO;
    Si PRIMERO = NIL
      Ent. *P.ANT:=NIL;
      *P.SIG:=NIL;
      PRIMERO:=P;
      ULTIMO :=P;
    Sino *ULTIMO.SIG:=P;
      *P.ANT := ULTIMO;
      ULTIMO := P;
      *P.SIG := NIL;
    Fsi;
  3: Si PRIMERO = NIL
    Ent. Escribir ('ERROR, Lista Vacía')
    Sino T := PRIMERO;
      Mientras (*T.SIG <> NIL) y (*T.VALOR <> DATO) hacer
        T := *T.SIG;
      Fmientras;
      Si (*T.VALOR <> DATO)
        Ent. Escribir ('ERROR, ', DATO, ' no está en la Lista')
        Sino Si PRIMERO = ULTIMO
          Ent. PRIMERO:=NIL;
          ULTIMO :=NIL;
        Sino Si PRIMERO = T
          Ent. (*T.SIG).ANT:=NIL;
          PRIMERO:=*T.SIG;
        Sino Si ULTIMO = T
          Ent. (*T.ANT).SIG:=NIL;
          ULTIMO:=*T.ANT;
        Sino (*T.ANT).SIG:=*T.SIG;

```

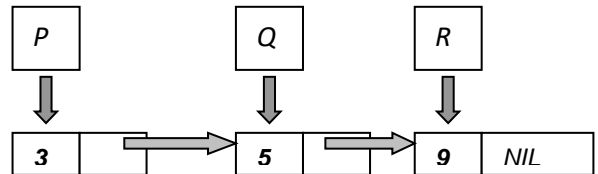
```
                (*T.SIG).ANT:=*T.ANT;
            Fsi;
        Fsi;
    Fsi;
    DISPONER(T);
Fsi;

4: Si PRIMERO = NIL
    Ent. Escribir ('ERROR, Lista Vacía')
    Sino T := PRIMERO;
        Mientras (T <> NIL) hacer
            Escribir (*T.VALOR);
            T := *T.SIG;
        Fmientras;
    Fsi;
Fsegún;
Hasta que OP = 5;
Facción.
```

Ejercicios Complementarios T.P. Nro. 4

1. Dada la siguiente declaración y el siguiente esquema:

P, Q, R: puntero a NODO
 NODO: registro
 DATO: entero
 PROX: puntero a nodo
 Fin registro



P	Q	R	Comentario

¿Qué hacen las siguientes órdenes?

Considerar que se ejecutan secuencialmente.

- 1) $P := P \star \text{PROX}$
- 2) $Q := P$
- 3) $R := P \star \text{PROX}$
- 4) $P \star \text{DATO} := Q \star \text{DATO}$
- 5) $P \star \text{DATO} := (Q \star \text{PROX}) \star \text{DATO}$
- 6) $R \star \text{PROX} := P$

2. Dada una lista simplemente encadenada de números enteros, diseñar un algoritmo que a partir de ella genere otra lista conteniendo los nodos cuyos datos sean múltiplos de 3, dichos elementos deberán ser eliminados de la lista original. Se asume que la lista está cargada y que el algoritmo recibe como parámetro de entrada la dirección del primer elemento.

3. Dada una lista doblemente encadenada que contiene datos de todas las provincias de la República Argentina: nombre, capital, cantidad total de habitantes y cantidad de analfabetos, y está ordenada alfabéticamente por nombre de provincia, se desea generar otra lista simplemente encadenada pero ordenada en orden decreciente por número de habitantes analfabetos.

4. Dada una lista circular con los datos de los socios de un club: número, nombre y condición ('A' = alta, 'B' = baja); hacer un algoritmo que cree otra lista simplemente encadenada con los socios múltiplos de 100, que no estén dados de baja y eliminar de la lista original los socios dados de baja (cond = 'B').

5. Una empresa FARMACÉUTICA, posee una lista simplemente enlazada de pedidos realizados por sus clientes en el último mes. Se desea generar dos grupos con la siguiente información:

- Los pedidos de los clientes deudores.
- Los pedidos de los clientes regulares.

Para ello se cuenta con un archivo "CLIENTES" indexado por **Nro_cliente**.

Deben tenerse en cuenta las siguientes consideraciones:

- La lista de entrada está ordenada por Nro_cliente y pueden existir varios pedidos de un mismo cliente.
- En el caso de recibir un pedido de un cliente que se encuentre dado de baja, debe ser dado de alta automáticamente.
- El total (Monto \$) de Dinero_deuda debe ser actualizado con la suma de los pedidos en Cuenta Corriente. Modificar la condición a Deudor si esa suma es mayor a cero.

Formato lista de entrada

Nro_cliente	Nro_pedido	Precio_pedido	TipoPedido (C:Contado, R: Cuenta Corriente)
-------------	------------	---------------	---

Formato lista salida

Nro_cliente	Total_pedido
-------------	--------------

Formato archivo "Clientes"

Nro_cliente	DNI	Domicilio	Dinero_deuda	Deudor (si/no)	Baja
-------------	-----	-----------	--------------	----------------	------

6. Usando las operaciones de Lista, escribir un procedimiento AGRUPA (PRIM, ULT: puntero a nodo; A:entero) que, dada una lista doblemente encadenada de enteros L agrupe (sume) elementos de tal manera que en L queden sólo elementos mayores o iguales que A. El algoritmo recorre la lista y cuando encuentra un elemento menor empieza a agrupar (sumar) el elemento con los siguientes hasta llegar al valor A o hasta que se acabe la lista (el elemento menor se debe eliminar de la lista.)

Por ejemplo si $L = \{3, 4, 2, 4, 1, 4, 4, 3, 2, 2, 4, 1, 4, 1, 4, 4, 1, 4, 4, 2\}$, entonces $AGRUPA(PRIM, ULT, 10)$ da $L = \{13, 12, 13, 10, 10\}$. En la lista final NO deben quedar elementos menores que A, salvo, eventualmente, el último.

7. La empresa "Remises Yapú S.A." necesita solucionar el problema de asignación de vehículos para sus clientes. La empresa posee 105 autos, y maneja dos colas (estructuras FIFO) una llamada "No Asignados" y otra llamada "Asignados".

En la primera se encuentran todos los autos que no han sido asignados a algún cliente en lo que va del día y en la segunda los autos que ya han sido asignados. La idea de la cola "Asignados" es manejar en forma equitativa la asignación de turnos, es decir que cada vehículo tenga igual oportunidad de obtener un cliente.

Los vehículos se asignan de la siguiente forma:

- Si existen autos "No Asignados", se toma uno de ellos y se lo mueve a la cola de "Asignados".
- Si no existen autos "No Asignados", se toma uno de la cola de "Asignados", y se lo coloca al final de la cola.

Siempre que se asigna un vehículo se debe incrementar en uno un campo "cantidad_de_clientes", a fin de saber cuantos clientes tuvo.

Por final de proceso se requiere saber la cantidad total de clientes de ese día.

Se pide: Confeccionar una subacción que realice la asignación de vehículos y el mantenimiento de las colas, usando listas.

8. Se dispone de una lista doblemente enlazada con un conjunto de números naturales. Este conjunto está dividido en 10 grupos. En cada grupo el primer elemento indica la cantidad de números subsiguientes a él que integran dicho grupo.

Se desea generar una nueva lista circular, simplemente encadenada, en la cual cada nodo contendrá el valor máximo de cada grupo. Dicha lista debe quedar ordenada en forma ascendente y, al final se debe informar cuál es el promedio de todos los máximos.

9. El ciclo de desarrollo de software tradicional, incremental e iterativo cuenta con las siguientes etapas: Captura de Requisitos, Análisis, Diseño, Desarrollo, Implementación y Prueba.

De la última etapa se obtiene un informe que contiene las modificaciones que deben realizarse al software para que cumpla con las funcionalidades para las cuales fue creado inicialmente.

Suponga que cuenta con una lista de proyectos de software de los cuales se tiene la siguiente información:

Proy	Resp	Fch_Ini	C_Error
AN(20)	AN(30)	Fecha	N(3)

El dato Cant_Errores indica la cantidad de errores que deben solucionarse en cada proyecto.

La información respecto a cada Error se almacena en una lista de acuerdo al orden de los proyectos y teniendo en cuenta la cantidad que le corresponden a éste. Por ejemplo si en la Lista 1 el Proyecto A tiene 3 errores, los 3 primeros nodos de la Lista 2 corresponden al Proyecto A. La información que se almacena por Error es la siguiente:

Desc_E	Est
AN(30)	AN(1)

Est – M: En Modificación | O: En Observación | R: Resuelto

Durante el proceso de Control de Proyectos se analiza cada error de cada proyecto, y se le consulta al Responsable el estado del Error (en caso que no haya sido Resuelto) para actualizarlo.

El Responsable debe tener la opción de recorrer todos los proyectos, en el orden almacenados, las veces que considere necesario, luego de analizar el último en la lista.

Suponiendo que se necesita simular el control de los proyectos, creando dos listas para almacenar por un lado **Proyectos** y por otro **Errores**. Y que además se debe informar al final del proceso qué cantidad de proyectos pasaron el control con todos sus errores resueltos. Se le solicita a ud:

- Dado el Algoritmo resuelto pero incompleto, analice la propuesta presentada e indique las sentencias incorrectas, encerrándolas con un círculo, e indique las modificaciones necesarias para que funcione, complete la líneas de puntos y agregue las acciones que falten.
- Desarrolle la acción Control_Proyectos y complete el ambiente si fuera necesario

<p>Accion Ej_1 (PRIM: Puntero a Nodo, PRIM2: Puntero a Nodo2) Es</p> <p>Ambiente NODO = Registro Proy: N(30) Resp: AN(30) Fecha: Fecha C_error: N(3) Prox: Puntero a Nodo FinRegistro NODO1 = Registro Desc_E: AN(30) Est: AN(1) Prox: Puntero FinRegistro P, K: Puntero a Nodo Q : Puntero a Nodo1 Cont, cant_proy: N(3) Opc: caracter</p> <p>Subaccion Cargar_Proyectos es Escribir("Ingrese 'S' para empezar o cualquier otro caracter para salir") Leer(opc) Mientras opc = 'S' hacer</p> <p>..... Leer(*P.Proy, *P.Resp, *P.Fch_Ini, *P.C)</p> <p>Si Entonces P := *P.prox; PRIM:=P; Sino K := PRIM Mientras (*K.prox <> PRIM) hacer K := *K.prox FinMientras *K.prox := P *P.prox := PRIM Fsi; Cont:= 0 Cargar_Errores Escribir("Ingrese 'S' para seguir cargando proyectos o cualquier otro caracter para salir") Leer(opc) Fin Mientras Fin Subaccion</p>	<p>Subacción Cargar_Errores es Mientras Cont <= *P.C_error hacer NUEVO(Q) Escribir(*Q.Desc_E, *P.Est) Si PRIM2 = NIL Entonces *Q.prox := Nil := Q Sino *Q.prox := PRIM2 PRIM2 := Q FinSi Fin Mientras FinSubaccion</p> <p>Algoritmo Cargar_Proyectos; Control_Proyectos; Escribir('La cantidad de proyectos con errores resueltos es de', *P.C_error)</p> <p>Fin Accion</p>
---	--

Algoritmos y
Estructuras de
Datos

Práctico 5

Recursividad
Árboles
Ejercicios Complementarios

2025

TRABAJO PRACTICO Nro. 5

Recursividad

1. Crear una función recursiva para solucionar cada uno de los enunciados siguientes:

1.01. Calcular el factorial de un número positivo n . Tener en cuenta la definición matemática de $n!$:

$$n! = \begin{cases} 1 & \text{si } n = 0 \\ n * (n-1)! & \text{si } n > 0 \end{cases}$$

1.02. Dado un número n como parámetro de entrada, calcular el n -ésimo número de la serie de Fibonacci. Tener en cuenta la siguiente definición:

$$\text{Fib}(n) = \begin{cases} 1 & \text{si } n = 1, n = 2 \\ \text{Fib}(n-1) + \text{Fib}(n-2) & \text{si } n > 2 \end{cases}$$

1.03. Dados dos números: a y b . Calcule la potencia a^b , usando sólo multiplicaciones sucesivas.

1.04. Construir un algoritmo recursivo que permita determinar si los dígitos de un número n dado son todos pares.

1.05. Dados dos números enteros, divídalos (división entera) y muestre el resultado, usando sólo la operación resta.

1.06. Determine recursivamente si un número dado es par o impar, usando sólo la operación resta.

1.07. El algoritmo de Euclides para encontrar el MCD (máximo común divisor) de dos números enteros positivos (m y n) se puede definir recursivamente.

Algoritmo de Euclides: el MCD de dos enteros es el entero mayor que divide a ambos.

Dividendo	Divisor	Cociente	Resto
m	n	q_1	r_1
n	r_1	q_2	r_2
r_1	r_2	q_3	r_3

Cuando el último resto es cero (por ej. $r_3 = 0$), el MCD es el último divisor (en ese caso, r_2).

El algoritmo recursivo se puede definir con los siguientes pasos:

- $\text{MCD}(m, n) = n$, si $n \leq m$ y n divide a m
- $\text{MCD}(m, n) = \text{MCD}(n, m)$ si $m < n$
- $\text{MCD}(m, n)$ es $\text{MCD}(n, \text{resto de } m \text{ dividido por } n)$

Para simplificar el algoritmo considerar que siempre $m > n$

1.08. Dado un vector de 10 números enteros, calcular la suma de sus elementos

2. Crear un procedimiento recursivo para solucionar cada uno de los enunciados siguientes:

2.01. Imprimir las cifras de un número n (siendo $n \geq 0$) en orden inverso al original. Por ej.: el inverso de 254 es 452.

2.02. Leer una palabra (una cadena de caracteres) y la cantidad de caracteres y generar su palíndromo. El palíndromo de "Venezuela" es "aleuzeneV".

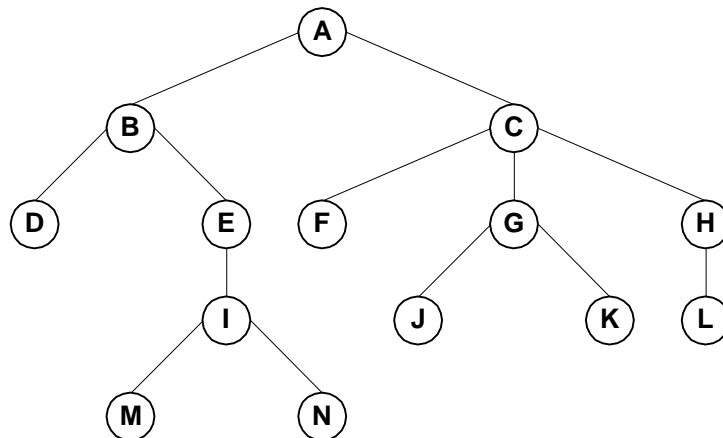
2.03. Dada una lista de nombres ordenada en forma ascendente, construir un procedimiento recursivo que imprima como salida la misma lista, pero en orden descendente, sin modificar la lista original.

2.04. Para convertir un número decimal a binario, simplemente debe dividirse sucesivas veces por dos (2) hasta quedarnos con el cociente cero (0). Todos los restos de las divisiones, tomados en orden inverso, forman el número binario objetivo. Escribir un procedimiento recursivo que, recibiendo como parámetro un número entero positivo, muestre en pantalla el mismo número codificado en binario.

Árboles

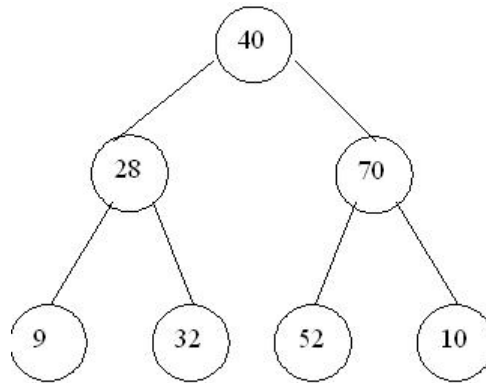
1. Responda las siguientes preguntas para el árbol de la figura:

- ¿Qué nodos son hojas?
- ¿Qué nodo es raíz?
- ¿Cuál es el padre del nodo C?
- ¿Qué nodos son los hijos de C?
- ¿Qué nodos son los antecesores de C?
- ¿Qué nodos son los descendientes de E?
- ¿Cuáles son los hermanos derechos de D y E?
- ¿Qué nodos están a izquierda y qué nodos a derecha de G?
- ¿Cuál es la profundidad del árbol?
- ¿Cuál es la altura del nodo C?



- ¿Cuántos caminos de longitud 3 hay en el árbol representado en la figura anterior?
- Dada la expresión de la siguiente línea, dibujar el árbol equivalente

$$[(x-y)*z] / (m+n**p)$$
- Dado el siguiente árbol:



- Indique el resultado del recorrido post-orden
- Indique el resultado del recorrido en-orden
- Elabore un algoritmo para el recorrido en-orden

5. Para las siguientes expresiones:

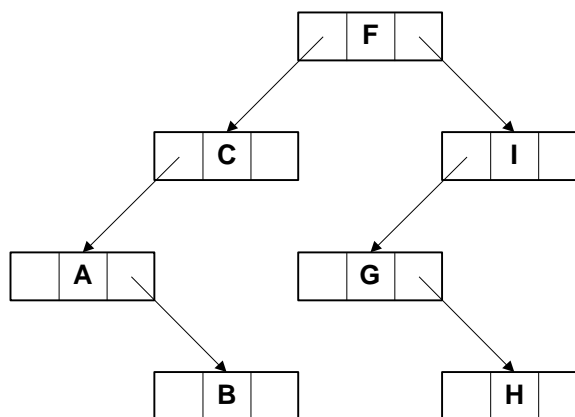
3.1. $(5 + 7) / 8 - (6 * 7) ** 2$

3.2. $3 - 6 + 6 * (8/3)$

3.3. $4 / (8 - 6 * (8 + 3))$

- Grafique el árbol equivalente
- Indique las expresiones resultantes de los recorridos EN-ORDEN, PRE-ORDEN y POST-ORDEN
- Calcule los resultados numéricos que arroja la ejecución de cada uno de los recorridos.

6. Escriba un algoritmo que permita recorrer el siguiente árbol en los tres procedimientos, Realice la prueba de escritorio:



7. Escribir una función recursiva que encuentre el número de nodos de un árbol binario.

8. Escribir una función recursiva que encuentre la altura de un árbol binario.

9. Se dispone de un árbol binario de enteros. Escribir funciones que calculen:

- La suma de sus elementos.
- La suma de sus elementos que son múltiplos de 3.

10. Suponiendo que un árbol está definido como la estructura recursiva de datos:

Arbol=registro

X: entero;

Izq, Dcha: puntero a arbol;

Freg;

Escribir un algoritmo que encuentre un elemento con una clave dada C, y realice una operación P con él.

11. Conviértase la expresión $((a + b) + c * (d + e) + f) * (g + h)$ en

a) expresión prefija

b) expresión postfija.

12. El recorrido en pre-orden de un determinado árbol binario es: **G E A I B M C L D F K J H** y en-orden **I A B E G L D C F M K H J**.

a) Dibujar el árbol binario.

b) Dar el recorrido en post-orden.

c) Diseñar un algoritmo para recorrer el árbol en post-orden.

13. Supongamos que tenemos una función valor tal que dado un valor de tipo char (una letra del alfabeto) devuelve un valor entero asociado a dicho identificador.

Supongamos también la existencia de un árbol de expresión T cuyos nodos hoja son letras del alfabeto y cuyos nodos interiores son los caracteres *, +, -, /.

Diseñar una función que tome como parámetros un nodo y un árbol binario y devuelva el resultado entero de la evaluación de la expresión representada.

EJERCICIOS DE ÁRBOLES BINARIOS DE BÚSQUEDA (ABB)

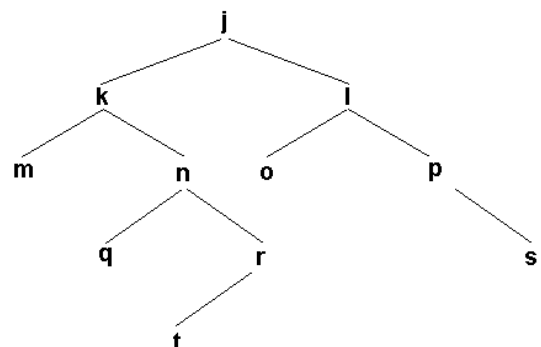
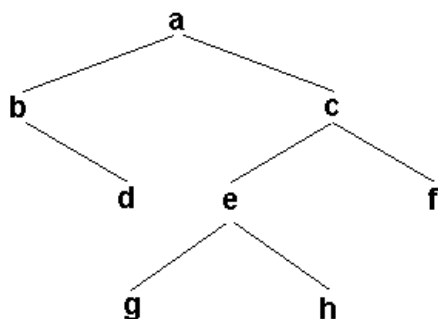
1) ¿Puede reconstruirse de forma única un ABB dado su inorden? ¿Y dados el preorden y el postorden?.

2) Construir un ABB con las claves 50, 25, 75, 10, 40, 60, 90, 35, 45, 70, 42.

3) ¿Bajo qué condiciones puede un árbol ser parcialmente ordenado y binario de búsqueda simultáneamente?. Razonar la respuesta.

EJERCICIOS DE ÁRBOLES EQUILIBRADOS AVL

Dados los siguientes árboles indicar cuál es AVL y cuál no.



Complementos Teóricos**ÁRBOLES EQUILIBRADOS AVL**

Diremos que un árbol binario está equilibrado (en el sentido de Addelson-Velskii y Landis) si, para cada uno de sus nodos ocurre que las alturas de sus dos subárboles difieren como máximo en 1. Los árboles que cumplen esta condición son denominados a menudo árboles AVL.

Ejercicios Complementarios T.P. Nro. 5

1. Indique los estados iniciales, intermedios y finales para cada variable incluida en las siguientes acciones

1.01. Acción Calc_101 es

```
C := TRUNC (ABSO (4/-2) + 3);
B := REDOND (-4,3);
B := B + C;
Escribir (C, B);
Facción.
```

1.02. ALFA vale 2,5; JOTA vale 2;

```
Acción Calc_102 (ALFA, JOTA: entero): entero es
  BETA := ALFA**JOTA;
  ALFA := ALFA + BETA;
  Calc_105 := ALFA;
Facción
```

1.03. H vale -10;

```
Acción Calc_103 (H: entero) es
  NUM := ABSO (H); A := 3;
  NUM2 := TRUNC (NUM/A) ** 2 /A;
  NUM := NUM2 - NUM * 5;
  B := REDOND (NUM/NUM2);
  A := (A + B) - 9 * 2 -1;
  Escribir ( NUM, A, NUM2, B);
Facción
```

2 Efectué los siguientes conjuntos de acciones, resolviendo las invocaciones en función de las consignas del punto 2. Realice la prueba de escritorio.

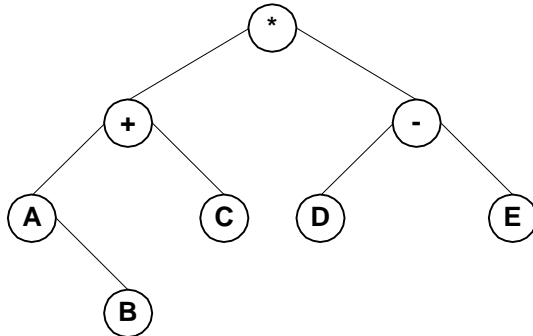
2.01 X:= 7; Y:= 10;
Z:=Calc_102 (X,Y);

2.02 X:= 4, Y:= 3;
K:= Calc_102 (X,Y) * 3 + ABSO(Y-10);

2.03 M:= -20;
Calc_103(M);

2.04. A:=0;
Calc_101;
A:= A + B;

3. Recorrer en los 3 modos el árbol siguiente:



4. El recorrido en orden de un árbol binario produce la secuencia de nodos: dfcgbca. Dibujar el árbol.

5. Escribir una función no recursiva para recorrer un árbol binario en el modo en-orden.

6. Escribir una función no recursiva para recorrer un árbol binario en Post-orden.

Algoritmos y
Estructuras de
Datos

Práctico 6

Complejidad Algorítmica

2025

Supongamos que disponemos de la siguiente definición de tipo: $vector = ARRAY[1..n] OF INTEGER;$ $n = ...;$

Consideramos entonces los procedimientos y funciones siguientes:

a) SUBACCION Algoritmo1(a:vector);

```
i,j:entero;
temp:entero;
PARA i:=1 HASTA n-1 HACER (* 1 *)
    PARA j:=n HASTA i+1 incremento -1 HACER (* 2 *)
        SI a[j-1]>a[j] ENTONCES (* 3 *)
            temp:=a[j-1]; (* 4 *)
            a[j-1]:=a[j]; (* 5 *)
            a[j]:=temp (* 6 *)
        FIN (* 7 *)
    FIN (* 8 *)
FIN (* 9 *)
FIN Algoritmo1;
```

b) SUBACCION Algoritmo2(VAR a:vector;c:ENTERO):entero;

```
VAR inf,sup,i:entero;
inf:=1; sup:=n; (* 1 *)
MIENTRAS (sup>=inf) HACER (* 2 *)
    i:=(inf+sup) DIV 2; (* 3 *)
    SI a[i]=c ENTONCES RETURN i (* 4 *)
    CONTRARIO SI c<a[i] ENTONCES sup:=i-1 (* 5 *)
    CONTRARIO inf:=i+1 (* 6 *)
FIN (* 7 *)
FIN; (* 8 *)
RETURN 0; (* 9 *)
FIN Algoritmo2;
```

c) SUBACCION Euclides(m,n:CARDINAL):CARDINAL;

```
VAR temp:CARDINAL;
MIENTRAS m>0 HACER (* 1 *)
    temp:=m; (* 2 *)
    m:=n MOD m; (* 3 *)
    n:=temp (* 4 *)
END; (* 5 *)
RETURN n (* 6 *)
END Euclides;
```

d) SUBACCION Misterio(n:CARDINAL);

```
VAR i,j,k,s:ENTERO;

s:=0; (* 1 *)
PARA i:=1 HASTA n-1 HACER (* 2 *)
    PARA j:=i+1 HASTA n HACER (* 3 *)
        PARA k:=1 HASTA j HACER (* 4 *)
            s:=s+2 (* 5 *)
        FIN (* 6 *)
    FIN (* 7 *)
FIN (* 8 *)
FIN Misterio;
```

- 1) Calcular sus tiempos de ejecución en el mejor, peor, y caso medio.
- 2) Dar cotas asintóticas para las funciones anteriores.

EJERCICIOS COMPLEMENTARIOS GENERALES

Secuencias de Texto, Arreglos y Listas

1. Dada una secuencia TEXTO se desea un algoritmo que cuente cuántas veces aparece la primera palabra del texto dentro del mismo y dentro de cada oración.
2. Escribir un algoritmo que produzca una secuencia ORACION de salida, formada por palabras en posiciones impares y que finalicen con la letra "N" de una secuencia TEXTO de entrada; en la salida cambiar las vocales por su vocal inmediata siguiente. (Ej. Si es "a", saldrá "e", si es "u" saldrá "a").
3. Escribir un algoritmo que permita desglosar una secuencia de caracteres en dos secuencias que contengan respectivamente los caracteres numéricos y los no numéricos contenidos en la secuencia original. Por final de proceso se deberá informar la cantidad de caracteres de cada una de las secuencias y las frecuencias de aparición de cada dígito ordenadas en forma creciente.
4. Dada una secuencia de texto (carácter, palabra, oración, texto), se desea generar otra secuencia que contenga la cantidad de palabra de longitud par de cada oración, y al final del proceso se emita un ranking decreciente por oración.
5. Dado un texto, efectuar un algoritmo que lo recorra y detecte la frecuencia de vocales por oración y luego liste dichos totales en orden decreciente.
6. Para un archivo texto confeccione un índice de palabras, indicando para cada una, en que renglón se encuentra y con qué ocurrencia. Enunciar literalmente la filosofía seguida en la solución.

Ejemplo: EJERCICIOS DE EXAMENES FINALES DE ALGORITMOS Y ESTRUCTURAS DE DATOS

1. Escribir un algoritmo que dadas dos secuencias oración de entrada, denominadas orac1 y orac2, genere una de salida, de manera tal que se intercalen las palabras de las secuencias de entrada, que cumplan con la siguiente condición:

De orac1 solo me interesan las palabras que comienzan y terminan con una vocal.

De orac2 solo me interesan las palabras de no más de 5 caracteres.

Al final del proceso informar cantidades de palabras de todas las secuencias.

3. Enuncie un trozo de algoritmo que produzca la inserción de un elemento en una lista doble.
4. Escriba un trozo de algoritmo para aplicar una búsqueda binaria sobre un arreglo unidimensional.
5. Clasifique los datos y sus estructuras, y describa en particular las listas lineales.