

CADP 2023

Práctica 0 – Introducción a la programación

1. Implemente un programa que lea por teclado dos números enteros e imprima en pantalla los valores leídos en orden inverso. Por ejemplo, si se ingresan los números 4 y 8, debe mostrar el mensaje: *Se ingresaron los valores 8 y 4*
2. Modifique el programa anterior para que el mensaje de salida muestre la suma de ambos números:
 - a. Utilizando una variable adicional
 - b. Sin utilizar una variable adicional

3. Implemente un programa que lea dos números reales e imprima el resultado de la división de los mismos con una precisión de dos decimales. Por ejemplo, si se ingresan los valores 4,5 y 7,2, debe imprimir: *El resultado de dividir 4,5 por 7,2 es 0,62*

Recuerde que para imprimir en pantalla números reales puede utilizar la notación: `writeln(X:Y:Z)` donde X es el número a imprimir, Y es el ancho (en cantidad de caracteres) que debe ocupar la impresión, y Z es la cantidad de decimales. Por ejemplo, sea el número $\pi=3.141592654$:

```
3.14 => writeln(pi,1,2);
```

```
3.14 => writeln(pi,8,2); (hay 4 espacios delante del 3, para completar 8 caracteres de ancho)
```

```
3,1415 => writeln(pi,1,4);
```

4. Implemente un programa que lea el diámetro **D** de un círculo e imprima:
 - a. El radio (R) del círculo (la mitad del diámetro).
 - b. El área del círculo. Para calcular el área de un círculo debe utilizar la fórmula $\pi \times R^2$
 - c. El perímetro del círculo. Para calcular el perímetro del círculo debe utilizar la fórmula $D \times \pi$ (o también $\pi \times R \times 2$).
5. Un kiosquero debe vender una cantidad **X** de caramelos entre **Y** clientes, dividiendo cantidades iguales entre todos los clientes. Los que le sobren se los quedará para él.
 - a. Realice un programa que lea la cantidad de caramelos que posee el kiosquero (X), la cantidad de clientes (Y), e imprima en pantalla un mensaje informando la cantidad de caramelos que le corresponderá a cada cliente, y la cantidad de caramelos que se quedará para sí mismo.
 - b. Imprima en pantalla el dinero que deberá cobrar el kiosquero si cada caramelo tiene un valor de \$1.60.

6. Realice un programa que informe el valor total en pesos de una transacción en dólares. Para ello, el programa debe leer el monto total en dólares de la transacción, el valor del dólar al día de la fecha y el porcentaje (en pesos) de la comisión que cobra el banco por la transacción. Por ejemplo, si la transacción se realiza por 10 dólares, el dólar tiene un valor 189,32 pesos y el banco cobra un 4% de comisión, entonces el programa deberá informar:

La transacción será de 1968,93 pesos argentinos

(resultado de multiplicar $10 \times 189,32$ y adicionarle el 4%)

CADP 2023

Práctica 1 – Estructuras de control: if y while

1. Realizar un programa que lea 2 números enteros desde teclado e informe en pantalla cuál de los dos números es el mayor. Si son iguales debe informar en pantalla lo siguiente: “Los números leídos son iguales”.
2. Realizar un programa que lea un número real e imprima su valor absoluto. El valor absoluto de un número X , se escribe $|X|$ y se define como:
 $|X| = X$ cuando X es mayor o igual a cero
 $|X| = -X$ cuando X es menor a cero

3. Realizar un programa que lea 3 números enteros y los imprima en orden descendente.
Por ejemplo, si se ingresan los valores 4, -10 y 12, deberá imprimir: 12 4 -10

4. Realizar un programa que lea un número real X . Luego, deberá leer números reales hasta que se ingrese uno cuyo valor sea exactamente el doble de X (el primer número leído).
5. Modifique el ejercicio anterior para que, luego de leer el número X , se lean **a lo sumo** 10 números reales. La lectura deberá finalizar al ingresar un valor que sea el doble de X , o al leer el décimo número, en cuyo caso deberá informarse “No se ha ingresado el doble de X ”.

6. Realizar un programa que lea el número de legajo y el promedio de cada alumno de la facultad. La lectura finaliza cuando se ingresa el legajo -1, que no debe procesarse.
Por ejemplo, se lee la siguiente secuencia:

33423
8.40
19003
6.43
-1

En el ejemplo anterior, se leyó el legajo 33422, cuyo promedio fue 8.40, luego se leyó el legajo 19003, cuyo promedio fue 6.43, y finalmente el legajo -1 (para el cual no es necesario leer un promedio).

Al finalizar la lectura, informar:

- a. La cantidad de alumnos leída (en el ejemplo anterior, se debería informar 2).
- b. La cantidad de alumnos cuyo promedio supera 6.5 (en el ejemplo anterior, se debería informar 1).
- c. El porcentaje de alumnos destacados (alumnos con promedio mayor a 8.5) cuyo legajo sean menor al valor 2500 (en el ejemplo anterior se debería informar 0%).

7. Realizar un programa que lea el código, el precio actual y el nuevo precio de los productos de un almacén. La lectura finaliza al ingresar el producto con el código 32767, el cual debe procesarse. Para cada producto leído, el programa deberá indicar si el nuevo precio del producto supera en un 10% al precio anterior.

Por ejemplo:

- Si se ingresa el código 10382, con precio actual 40, y nuevo precio 44, deberá imprimir: “el aumento de precio del producto 10382 no supera el 10%”
- Si se ingresa el código 32767, con precio actual 30 y nuevo precio 33,01, deberá imprimir: “el aumento de precio del producto 32767 es superior al 10%”

8. Realizar un programa que lea tres caracteres, e informe si los tres eran letras vocales o si al menos uno de ellos no lo era. Por ejemplo, si se leen los caracteres “a e o” deberá informar “Los tres son vocales”, y si se leen los caracteres “z a g” deberá informar “al menos un carácter no era vocal”.

9. Realizar un programa que lea un carácter, que puede ser “+” (suma) o “-” (resta); si se ingresa otro carácter, debe informar un error y finalizar. Una vez leído el carácter de suma o resta, deberá leerse una secuencia de números enteros que finaliza con 0. El programa deberá aplicar la operación leída con la secuencia de números, e imprimir el resultado final.

Por ejemplo:

- Si se lee el carácter “-” y la secuencia 4 3 5 -6 0, deberá imprimir: $2 (4 - 3 - 5 - (-6))$
- Si se lee el carácter “+” y la secuencia -10 5 6 -1 0, deberá imprimir $0 (-10 + 5 + 6 + (-1))$

CADP 2023

Práctica 1 (parte 2) – Estruc. de control: for y repeat-until

1. Realizar un programa que lea 10 números enteros e informe la suma total de los números leídos.
 - a. Modifique el ejercicio 1 para que además informe la cantidad de números mayores a 5.
2. Realice un programa que lea 10 números e informe cuál fue el mayor número leído. Por ejemplo, si se lee la secuencia:

3 5 6 2 3 10 98 8 -12 9

deberá informar: "El mayor número leído fue el 98"

- a. Modifique el programa anterior para que, además de informar el mayor número leído, se informe el número de orden, dentro de la secuencia, en el que fue leído. Por ejemplo, si se lee la misma secuencia:

3 5 6 2 3 10 98 8 -12 9

deberá informar: "El mayor número leído fue el 98, en la posición 7"

3. Realizar un programa que lea desde teclado la información de alumnos ingresantes a la carrera Analista en TIC. De cada alumno se lee nombre y nota obtenida en el módulo EPA (la nota es un número entre 1 y 10). La lectura finaliza cuando se lee el nombre "Zidane Zinedine", que debe procesarse. Al finalizar la lectura informar:
 - La cantidad de alumnos aprobados (nota 8 o mayor) y
 - la cantidad de alumnos que obtuvieron un 7 como nota.
4. Realizar un programa que lea 1000 números enteros desde teclado. Informar en pantalla cuáles son los dos números mínimos leídos.
 - a. Modifique el ejercicio anterior para que, en vez de leer 1000 números, la lectura finalice al leer el número 0, el cual **debe** procesarse.
 - b. Modifique el ejercicio anterior para que, en vez de leer 1000 números, la lectura finalice al leer el número 0, el cual **no debe** procesarse
5. Realizar un programa que lea números enteros desde teclado. La lectura debe finalizar cuando se ingrese el número 100, el cual debe procesarse. Informar en pantalla:
 - El número máximo leído.
 - El número mínimo leído.
 - La suma total de los números leídos.

6. Realizar un programa que lea información de 200 productos de un supermercado. De cada producto se lee código y precio (cada código es un número entre 1 y 200). Informar en pantalla:
 - Los códigos de los dos productos más baratos.
 - La cantidad de productos de más de 16 pesos con código par.
7. Realizar un programa que lea desde teclado información de autos de carrera. Para cada uno de los autos se lee el nombre del piloto y el tiempo total que le tomó finalizar la carrera. En la carrera participaron 100 autos. Informar en pantalla:
 - Los nombres de los dos pilotos que finalizaron en los dos primeros puestos.
 - Los nombres de los dos pilotos que finalizaron en los dos últimos puestos.
8. Un local de ropa desea analizar las ventas realizadas en el último mes. Para ello se lee por cada día del mes, los montos de las ventas realizadas. La lectura de montos para cada día finaliza cuando se lee el monto 0. Se asume un mes de 31 días. Informar la cantidad de ventas por cada día, y el monto total acumulado en ventas de todo el mes.
 - a. Modifique el ejercicio anterior para que además informe el día en el que se realizó la mayor cantidad de ventas.

Guía opcional de actividades adicionales

Aclaración: Las actividades de esta guía tienen por objetivo integrar todos los temas vistos hasta ahora en la práctica. Se aconseja realizar estas actividades una vez completados todos los ejercicios de la práctica 1 partes 1 y 2.

Actividad 1: Revisando Inversiones

Realizar un programa que analice las inversiones de las empresas más grandes del país. Para cada empresa se lee su código (un número entero), la cantidad de inversiones que tiene, y el monto dedicado a cada una de las inversiones. La lectura finaliza al ingresar la empresa con código 100, que debe procesarse.

El programa deberá informar:

- Para cada empresa, el monto promedio de sus inversiones
- Código de la empresa con mayor monto total invertido
- Cantidad de empresas con inversiones de más de \$50000

Por ejemplo:

Ingrese un código de empresa: 33

Ingrese la cant. de inversiones: 4

Ingrese el monto de cada inversión: 33200 56930 24500.85 10000

Resultado del análisis: Empresa 33 Monto promedio 31157,71

Ingrese un código de empresa: 41

Ingrese la cant. de inversiones: 3

Ingrese el monto de cada inversión: 102000.22 53000 12000

Resultado del análisis: Empresa 41 Monto promedio 55666,74

Ingrese un código de empresa: 100

Ingrese la cant. de inversiones: 1

Ingrese el monto de cada inversión: 84000.34

Resultado del análisis: Empresa 100 Monto promedio 84000.34

(Fin de la lectura)

La empresa 41 es la que mayor dinero posee invertido (\$167000.22).

Hay 3 empresas con inversiones por más de \$50000

Actividad 2: procesamiento de las autoevaluaciones de CADP

La cátedra de CADP está analizando los resultados de las autoevaluaciones que realizaron los alumnos durante el cuatrimestre. Realizar un programa que lea, para cada alumno, su legajo, su condición (I para INGRESANTE, R para RECURSANTE), y la nota obtenida en cada una de las 5 autoevaluaciones. Si un alumno no realizó alguna autoevaluación en tiempo y forma, se le cargará la nota -1. La lectura finaliza al ingresar el legajo -1. Por ejemplo, si la materia tuviera dos alumnos, un ingresante y un recursante, la lectura podría ser así:

Legajo: 19003

Condición: R

Notas: 8 10 6 -1 8

Legajo 21020

Condición: I

Notas: 4 0 6 10 -1

Legajo -1

(Fin de la lectura)

Una vez ingresados todos los datos, el programa debe informar:

- Cantidad de alumnos INGRESANTES en condiciones de rendir el parcial y porcentaje sobre el total de alumnos INGRESANTES.
- Cantidad de alumnos RECURSANTES en condiciones de rendir el parcial y porcentaje sobre el total de alumnos RECURSANTES.
- Cantidad de alumnos que aprobaron todas las autoevaluaciones
- Cantidad de alumnos cuya nota **promedio** fue mayor a 6.5 puntos
- Cantidad de alumnos que obtuvieron cero puntos en **al menos** una autoevaluación.
- Código de los dos alumnos con mayor cantidad de autoevaluaciones con nota 10 (diez)
- Código de los dos alumnos con mayor cantidad de autoevaluaciones con nota 0 (cero)

Nota: recuerde que, para poder rendir el EXAMEN PARCIAL, el alumno deberá obtener “Presente” en al menos el 75% del total de las autoevaluaciones propuestas. Se considera “Presente” la autoevaluación que se entrega en tiempo y forma y con al menos el 40% de respuestas correctas.

Actividad 3: Ventas de tanques de agua

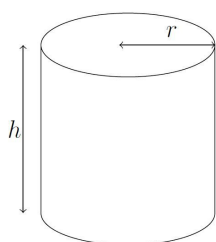
Un fabricante de tanques de agua está analizando las ventas de sus tanques durante el 2020. La empresa fabrica tanques a medida, que pueden ser rectangulares (tanques ‘R’) o cilíndricos (tanques ‘C’).

- De cada tanque R se conoce su ancho (A), su largo (B) y su alto (C)
- De cada tanque C se conoce su radio y su alto

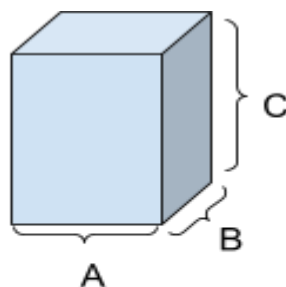
Todas las medidas se ingresan en metros. Realizar un programa que lea la información de los tanques vendidos por la empresa. La lectura finaliza al ingresar un tanque de tipo ‘Z’. Al finalizar la lectura, el programa debe informar:

- Volumen de los dos mayores tanques vendidos
- Volumen promedio de todos los tanques cilíndricos vendidos
- Volumen promedio de todos los tanques rectangulares vendidos
- Cantidad de tanques cuyo alto sea menor a 1.40 metros
- Cantidad de tanques cuyo volumen sea menor a 800 metros cúbicos

Recordar: las fórmulas para el cálculo de volumen (V) del cilindro y del paralelepípedo rectangular son:



$$V = \pi \cdot r^2 \cdot h$$



$$V = A \cdot B \cdot C$$

CADP 2023

Práctica 2 (parte 1) – Funciones y procedimientos

Aclaración importante: si bien en esta práctica se presentan soluciones utilizando variables globales, el objetivo de las mismas es comprender su funcionamiento y mostrar algunos de los problemas asociados con su uso. A la hora de resolver ejercicios en las siguientes prácticas, NO se deberán utilizar variables globales.

1. Dado el siguiente programa, indicar qué imprime.

```
program alcance1;  
var a,b: integer;  
procedure uno;  
var b: integer;  
begin  
  b := 3;  
  writeln(b);  
end;  
begin  
  a:= 1;  
  b:= 2;  
  uno;  
  writeln(b, a);  
end.
```

2. Dado el siguiente programa, indicar qué imprime.

```
program alcance2;  
var a,b: integer;  
procedure uno;  
begin  
  b := 3;  
  writeln(b);  
end;  
begin  
  a:= 1;  
  b:= 2;  
  uno;  
  writeln(b, a);  
end.
```

3. Dado el siguiente programa, indicar cuál es el error y su causa.

```
program alcance3;  
var a: integer;  
procedure uno;  
var b: integer;
```

```

begin
  b:= 2;
  writeln(b);
end;
begin
  a:= 1;
  uno;
  writeln(a, b);
end.

```

4. Dado los siguientes programas, explicar la diferencia.

<pre> program alcance4a; var a,b: integer; procedure uno; begin a := 1; writeln(a); end; begin a:= 1; b:= 2; uno; writeln(b, a); end. </pre>	<pre> program alcance4b; procedure uno; begin a := 1; writeln(a); end; var a,b: integer; begin a:= 1; b:= 2; uno; writeln(b, a); end. </pre>
--	--

5. Dado el siguiente programa, indicar cuál es el error.

```

program alcance4;
function cuatro: integer;
begin
  cuatro:= 4;
end;
var a: integer;
begin
  cuatro;
  writeln(a);
end.

```

6. a. Realice un módulo que lea de teclado números enteros hasta que llegue un valor negativo. Al finalizar la lectura el módulo debe imprimir en pantalla cuál fue el número par más alto.
- b. Implemente un programa que invoque al módulo del inciso a.

7. Dado el siguiente programa:

```
program alcanceYFunciones;
var
  suma, cant : integer;
function calcularPromedio : real
var
  prom : real;
begin
  if (cant = 0) then
    prom := -1
  else
    prom := suma / cant;
  end;
begin { programa principal }
  readln(suma);
  readln(cant);
  if (calcularPromedio <> -1) then begin
    cant := 0;
    writeln('El promedio es: ' , calcularPromedio)
  end;
else
  writeln('Dividir por cero no parece ser una buena idea');
end.
```

- a) La función calcularPromedio calcula y retorna el promedio entre las variables globales suma y cant, pero parece incompleta. ¿qué debería agregarle para que funcione correctamente?
- b) En el programa principal, la función calcularPromedio es invocada dos veces, pero esto podría mejorarse. ¿cómo debería modificarse el programa principal para invocar a dicha función una única vez?
- c) Si se leen por teclado los valores 48 (variable suma) y 6 (variable cant), ¿qué resultado imprime el programa? Considere las tres posibilidades:
 - i) El programa original
 - ii) El programa luego de realizar la modificación del inciso a)
 - iii) El programa luego de realizar las modificaciones de los incisos a) y b)

8. Dado el siguiente programa:

```
program anidamientos;
  procedure leer;
  var
    letra : char;
  function analizarLetra : boolean
  begin
    if (letra >= 'a') and (letra <= 'z') then
      analizarLetra := true;
    else
      if (letra >= 'A') and (letra <= 'Z') then
        analizarLetra := false;
      end; { fin de la funcion analizarLetra }
    begin
      readln(letra);
      if (analizarLetra) then
        writeln('Se trata de una minúscula')
      else
        writeln('Se trata de una mayúscula');
      end; { fin del procedure Leer}
  end;
var
  ok : boolean;
begin { programa principal }
  leer;
  ok := analizarLetra;
  if ok then
    writeln('Gracias, vuelva prontoss');
end.
```

- a) La función **analizarLetra** fue declarada como un submódulo dentro del procedimiento leer. Pero esto puede traer problemas en el código del programa principal.
- i) ¿qué clase de problema encuentra?
 - ii) ¿cómo se puede resolver el problema para que el programa compile y funcione correctamente?
- b) La función **analizarLetra** parece incompleta, ya que no cubre algunos valores posibles de la variable letra.
- i) ¿De qué valores se trata?
 - ii) ¿Qué sucede en nuestro programa si se ingresa uno de estos valores?
 - iii) ¿Cómo se puede resolver este problema?

CADP 2023

Práctica 2 (parte 2) – Funciones, procedimientos y parámetros

1. Responda las preguntas en relación al siguiente programa:

```
program Ejercicio3;
  procedure suma(num1: integer; var num2:integer);
  begin
    num2 := num1 + num2;
    num1 := 0;
  end;
var
  i, x : integer;
begin
  read(x); { Leo la variable x }
  for i:= 1 to 5 do
    suma(i,x);
    write(x); { imprimo las variable x }
  end.
```

- ¿Qué imprime si se lee el valor 10 en la variable x ?
- ¿Qué imprime si se lee el valor 10 en la variable x y se cambia el encabezado del *procedure* por:
`procedure suma(num1: integer; num2:integer);`
- ¿Qué sucede si se cambia el encabezado del *procedure* por:
`procedure suma(var num1: integer; var num2:integer);`

2. Responda la pregunta en relación al siguiente programa:

```
Program ejercicio4;
procedure digParesImpares(num : integer; var par, impar : integer);
var
  dig: integer;
begin
  while (num <> 0) do begin
    dig:= num mod 10;
    if((dig mod 2)= 0) then
      par := par + 1
    else
      impar:= impar +1;
    num := num DIV 10;
  end;
end;
var
  dato, par, impar, total, cant : integer;
begin
  par := 0;
```

```

    impar := 0;
    repeat
        read(dato);
        digParesImpares(dato,par,impar);
    until (dato = 100);
    writeln('Pares: ',par, 'Ímpares:', impar);
end.

```

a. ¿Qué imprime si se lee la siguiente secuencia de valores? 250, 35, 100

3. Encontrar los 6 errores que existen en el siguiente programa. Utilizar los comentarios entre llaves como guía, indicar en qué línea se encuentra cada error y en qué consiste:

```

1.    program ejercicio5;
2.        { suma los números entre a y b, y retorna el resultado en c }
3.    procedure sumar(a, b, c : integer)
4.    var
5.        suma : integer;
6.    begin
7.
8.        for i := a to b do
9.            suma := suma + i;
10.         c := c + suma;
11.    end;
12.    var
13.        result : integer;
14.    begin
15.        result := 0;
16.        readln(a); readln(b);
17.        sumar(a, b, 0);
18.        write('La suma total es ',result);
19.        { averigua si el resultado final estuvo entre 10 y 30 }
20.        ok := (result >= 10) or (result <= 30);
21.        if (not ok) then
22.            write ('La suma no quedó entre 10 y 30');
23.    end.

```

4. El siguiente programa intenta resolver un enunciado. Sin embargo, el código posee 5 errores. Indicar en qué línea se encuentra cada error y en qué consiste el error.

Enunciado: Realice un programa que lea datos de 130 programadores Java de una empresa. De cada programador se lee el número de legajo y el salario actual. El programa debe imprimir el total del dinero destinado por mes al pago de salarios, y el salario del empleado mayor legajo.

```
1. program programadores;
2.   procedure leerDatos(var legajo: integer; salario : real);
3.   begin
4.     writeln('Ingrese el nro de legajo y el salario');
5.     read(legajo);
6.     read(salario);
7.   end;
8.   procedure actualizarMaximo(nuevoLegajo:integer; nuevoSalario:real; var maxLegajo:integer);
9.   var
10.    maxSalario : real;
11.   begin
12.     if (nuevoLegajo > maxLegajo) then begin
13.       maxLegajo:= nuevoLegajo;
14.       maxSalario := nuevoSalario
15.     end;
16.   end;
17.   var
18.    legajo, maxLegajo, i : integer;
19.    salario, maxSalario : real;
20.   begin
21.     sumaSalarios := 0;
22.     for i := 1 to 130 do begin
23.       leerDatos(salario, legajo);
24.       actualizarMaximo(legajo, salario, maxLegajo);
25.       sumaSalarios := sumaSalarios + salario;
26.     end;
27.     writeln('En todo el mes se gastan ', sumaSalarios, ' pesos');
28.     writeln('El salario del empleado más nuevo es ',maxSalario);
29.   end.
```

5. a. Realizar un módulo que reciba un par de números (numA,numB) y retorne si numB es el doble de numA.

b. Utilizando el módulo *realizado* en el inciso a., realizar un programa que lea secuencias de pares de números hasta encontrar el par (0,0), e informe la cantidad total de pares de números leídos y la cantidad de pares en las que numB es el doble de numA.

Ejemplo: si se lee la siguiente secuencia: (1,2) (3,4) (9,3) (7,14) (0,0) el programa debe informar los valores 4 (cantidad de pares leídos) y 2 (cantidad de pares en los que numB es el doble de numA).

6. Realizar un programa modularizado que lea datos de 100 productos de una tienda de ropa. Para cada producto debe leer el precio, código y tipo (pantalón, remera, camisa, medias, campera, etc.). Informar:

- Código de los dos productos más baratos.
- Código del producto de tipo “pantalón” más caro.
- Precio promedio.

7. a. Realizar un módulo que reciba como parámetro un número entero y retorne la cantidad de dígitos que posee y la suma de los mismos.

b. Utilizando el módulo anterior, realizar un programa que lea una secuencia de números e imprima la cantidad total de dígitos leídos. La lectura finaliza al leer un número cuyos dígitos suman exactamente 10, el cual debe procesarse.

8. Realizar un programa modularizado que lea secuencia de números enteros. La lectura finaliza cuando llega el número 123456, el cual no debe procesarse. Informar en pantalla para cada número la suma de sus dígitos pares y la cantidad de dígitos impares que posee.

9. Realizar un programa modularizado que lea información de alumnos de una facultad. Para cada alumno se lee: número de inscripción, apellido y nombre. La lectura finaliza cuando se ingresa el alumno con número de inscripción 1200, que debe procesarse. Se pide calcular e informar:

- Apellido de los dos alumnos con número de inscripción más chico.
- Nombre de los dos alumnos con número de inscripción más grande.
- Porcentaje de alumnos con nro de inscripción par.

10. Realizar un programa modularizado que lea una secuencia de caracteres y verifique si cumple con el patrón **A\$B#**, donde:

- **A** es una secuencia de sólo letras vocales
- **B** es una secuencia de sólo caracteres alfabéticos sin letras vocales
- los caracteres **\$** y **#** seguro existen

Nota: en caso de no cumplir, informar que parte del patrón no se cumplió.

11. Realizar un programa modularizado que lea una secuencia de caracteres y verifique si cumple con el patrón **A%B***, donde:

- **A** es una secuencia de caracteres en la que no existe el carácter '**\$**'.
- **B** es una secuencia con la misma cantidad de caracteres que aparecen en **A** y en la que aparece a lo sumo 3 veces el carácter '@'.
- Los caracteres **%** y ***** seguro existen

Nota: en caso de no cumplir, informar que parte del patrón no se cumplió.

12. a. Realizar un módulo que calcule el rendimiento económico de una plantación de soja. El módulo debe recibir la cantidad de hectáreas (ha) sembradas, el tipo de zona de siembra (1: zona muy fértil, 2: zona estándar, 3: zona árida) y el precio en U\$S de la tonelada de soja; y devolver el rendimiento económico esperado de dicha plantación.

Para calcular el rendimiento económico esperado debe considerar el siguiente rendimiento por tipo de zona:

Tipo de zona	Rendimiento por ha
1	6 toneladas por ha
2	2,6 toneladas por ha
3	1,4 toneladas por ha

b. ARBA desea procesar información obtenida de imágenes satelitales de campos sembrados con soja en la provincia de Buenos Aires. De cada campo se lee: localidad, cantidad de hectáreas sembradas y el tipo de zona (1, 2 ó 3). La lectura finaliza al leer un campo de 900 ha en la localidad 'Saladillo', que debe procesarse. El precio de la soja es de U\$S320 por tn. Informar:

- La cantidad de campos de la localidad Tres de Febrero con rendimiento estimado superior a U\$S 10.000.
- La localidad del campo con mayor rendimiento económico esperado.
- La localidad del campo con menor rendimiento económico esperado.
- El rendimiento económico promedio.

EJERCICIOS ADICIONALES

1. Dado el siguiente programa:

```
program Ejercicio1_ad;
  procedure intercambio(var num1,num2 : integer);
  var
    aux : integer;
  begin
    aux := num1;
    num1 := num2;
    num2 := aux;
  end;
  procedure sumar(num1 : integer; var num2 : integer);
  begin
    num2 := num1 + num2;
  end;
var
  i, num1, num2 : integer;
begin
  read(num1);
  read(num2);
  for i := 1 to 3 do begin
    intercambio(num1,num2);
    sumar(i,num1);
  end;
  writeln(num1);
end.
```

- a. ¿Qué imprime si se leen los valores num1=10 y num2=5 ?
b. ¿Qué imprime si se leen los valores num1=5 y num2=10 ?

2. Realice un programa modularizado que lea 10 pares de números (X,Y) e informe, para cada par de números, la suma y el producto de todos los números entre X e Y.

Por ejemplo, dado el par (3,6), debe informar:

“La suma es 18” (obtenido de calcular 3+4+5+6)

“El producto es 360” (obtenido de calcular 3*4*5*6)

3. Realizar un programa modularizado que lea información de 200 productos de un supermercado. De cada producto se lee código y precio (cada código es un número entre 1 y 200). Informar en pantalla:
- Los códigos de los dos productos más baratos.
 - La cantidad de productos de más de 16 pesos con código par.

- 4.** a. Realizar un módulo que reciba como parámetro el radio de un círculo y retorne su diámetro y su perímetro.
- b. Utilizando el módulo anterior, realizar un programa que analice información de planetas obtenida del Telescopio Espacial Kepler. De cada planeta se lee su nombre, su radio (medido en kilómetros) y la distancia (medida en años luz) a la Tierra. La lectura finaliza al leer un planeta con radio 0, que no debe procesarse. Informar:
- Nombre y distancia de los planetas que poseen un diámetro menor o igual que el de la Tierra (12.700 km) y mayor o igual que el de Marte (6.780 km).
 - Cantidad de planetas con un perímetro superior al del planeta Júpiter (439.264 km).
- 5.** En la “Práctica 1 - Ejercicios Adicionales” se resolvieron 3 problemas complejos sin utilizar módulos. Al carecer de herramientas para modularizar, esos programas resultaban difíciles de leer, de extender y de depurar.
- a) Analice sus soluciones a dichos problemas, e identifique:
- ¿Qué porciones de su código podrían modularizarse? en qué casos propondría una estructura de módulos anidada?
 - ¿Qué tipo de módulo (función o procedimiento) conviene utilizar en cada caso? Existe algún caso en los que sólo un tipo de módulo es posible?
 - ¿Qué mecanismos de comunicación conviene utilizar entre los módulos propuestos?
- b) Implemente nuevamente los 3 programas, teniendo en cuenta los módulos propuestos en el inciso anterior

CADP 2023

Práctica 3 – Registros

1. Dado el siguiente programa:

```
program Registros;
type
    str20 = string[20];
    alumno = record
        codigo : integer;
        nombre : str20;
        promedio : real;
    end;
procedure leer(var alu : alumno);
begin
    writeln('Ingrese el código del alumno');
    read(alu.codigo);
    if (alu.codigo <> 0) then begin
        writeln('Ingrese el nombre del alumno'); read(alu.nombre);
        writeln('Ingrese el promedio del alumno'); read(alu.promedio);
    end;
end;
{ declaración de variables del programa principal }
var
    a : alumno;
{ cuerpo del programa principal }
begin
    ...
end.
```

- a. Completar el programa principal para que lea información de alumnos (código, nombre, promedio) e informe la cantidad de alumnos leídos. La lectura finaliza cuando ingresa un alumno con código 0, que no debe procesarse. Nota: utilizar el módulo leer.
- b. Modificar al programa anterior para que, al finalizar la lectura de todos los alumnos, se informe también el nombre del alumno con mejor promedio.

2. El registro civil de La Plata ha solicitado un programa para analizar la distribución de casamientos durante el año 2019. Para ello, cuenta con información de las fechas de todos los casamientos realizados durante ese año.

- a) Analizar y definir un tipo de dato adecuado para almacenar la información de la fecha de cada casamiento.
- b) Implementar un módulo que lea una fecha desde teclado y la retorne en un parámetro cuyo tipo es el definido en el inciso a).
- c) Implementar un programa que lea la fecha de todos los casamientos realizados en 2019. La lectura finaliza al ingresar el año 2020, que no debe procesarse, e informe la cantidad de casamientos realizados durante los

meses de verano (enero, febrero y marzo) y la cantidad de casamientos realizados en los primeros 10 días de cada mes. Nota: utilizar el módulo realizado en **b)** para la lectura de fecha.

- 3.** El Ministerio de Educación desea realizar un relevamiento de las 2400 escuelas primarias de la provincia de Bs. As, con el objetivo de evaluar si se cumple la proporción de alumnos por docente calculada por la UNESCO para el año 2015 (1 docente cada 23,435 alumnos). Para ello, se cuenta con información de: CUE (código único de establecimiento), nombre del establecimiento, cantidad de docentes, cantidad de alumnos, localidad. Se pide implementar un programa que procese la información y determine:

- Cantidad de escuelas de La Plata con una relación de alumnos por docente superior a la sugerida por UNESCO.
- CUE y nombre de las dos escuelas con mejor relación entre docentes y alumnos.

El programa debe utilizar:

- a) Un módulo para la lectura de la información de la escuela.
- b) Un módulo para determinar la relación docente-alumno (esa relación se obtiene del cociente entre la cantidad de alumnos y la cantidad de docentes).

- 4.** Una compañía de telefonía celular debe realizar la facturación mensual de sus 9300 clientes con planes de consumo ilimitados (clientes que pagan por lo que consumen). Para cada cliente se conoce su código de cliente y cantidad de líneas a su nombre. De cada línea se tiene el número de teléfono, la cantidad de minutos consumidos y la cantidad de MB consumidos en el mes. Se pide implementar un programa que lea los datos de los clientes de la compañía e informe el monto total a facturar para cada uno. Para ello, se requiere:

- a. Realizar un módulo que lea la información de una línea de teléfono.
- b. Realizar un módulo que reciba los datos de un cliente, lea la información de todas sus líneas (utilizando el módulo desarrollado en el inciso a.) y retorne la cantidad total de minutos y la cantidad total de MB a facturar del cliente.

Nota: para realizar los cálculos tener en cuenta que cada minuto cuesta \$3,40 y cada MB consumido cuesta \$1,35.

- 5.** Realizar un programa que lea información de autos que están a la venta en una concesionaria. De cada auto se lee: marca, modelo y precio. La lectura finaliza cuando se ingresa la marca "ZZZ" que no debe procesarse. La información se ingresa ordenada por marca. Se pide calcular e informar:

- a. El precio promedio por marca.
- b. Marca y modelo del auto más caro.

- 6.** Una empresa importadora de microprocesadores desea implementar un sistema de *software* para analizar la información de los productos que mantiene actualmente en *stock*. Para ello, se conoce la siguiente información de los microprocesadores: marca (Intel, AMD, NVidia, etc), línea (Xeon, Core i7, Opteron, Atom, Centrino, etc.), cantidad de *cores* o núcleos de procesamiento (1, 2, 4, 8), velocidad del reloj (medida en Ghz) y tamaño en nanómetros (nm) de los transistores (14, 22, 32, 45, etc.). La información de los microprocesadores se lee de forma consecutiva por marca de procesador y la lectura finaliza al ingresar un procesador con 0 *cores* (que no debe procesarse). Se pide implementar un programa que lea información de los microprocesadores de la empresa importadora e informe:

- Marca y línea de todos los procesadores de más de 2 *cores* con transistores de a lo sumo 22 nm.
 - Las dos marcas con mayor cantidad de procesadores con transistores de 14 nm.
 - Cantidad de procesadores *multicore* (de más de un *core*) de Intel o AMD, cuyos relojes alcancen velocidades de al menos 2 Ghz.
7. Realizar un programa que lea información de centros de investigación de Universidades Nacionales. De cada centro se lee su nombre abreviado (ej. LIDI, LIFIA, LINTI), la universidad a la que pertenece, la cantidad de investigadores y la cantidad de becarios que poseen. La información se lee de forma consecutiva por universidad y la lectura finaliza al leer un centro con 0 investigadores, que no debe procesarse. Informar:
- Cantidad total de centros para cada universidad.
 - Universidad con mayor cantidad de investigadores en sus centros.
 - Los dos centros con menor cantidad de becarios.
8. La Comisión Provincial por la Memoria desea analizar la información de los proyectos presentados en el programa Jóvenes y Memoria durante la convocatoria 2020. Cada proyecto posee un código único, un título, el docente coordinador (DNI, nombre y apellido, email), la cantidad de alumnos que participan del proyecto, el nombre de la escuela y la localidad a la que pertenecen. Cada escuela puede presentar más de un proyecto. La información se ingresa ordenada consecutivamente por localidad y, para cada localidad, por escuela. Realizar un programa que lea la información de los proyectos hasta que se ingrese el proyecto con código -1 (que no debe procesarse), e informe:
- Cantidad total de escuelas que participan en la convocatoria 2018 y cantidad de escuelas por cada localidad.
 - Nombres de las dos escuelas con mayor cantidad de alumnos participantes.
 - Título de los proyectos de la localidad de Daireaux cuyo código posee igual cantidad de dígitos pares e impares.

EJERCICIOS ADICIONALES

9. Realizar un programa que lea información de los candidatos ganadores de las últimas elecciones a intendente de la provincia de Buenos Aires. Para cada candidato se lee: localidad, apellido del candidato, cantidad de votos obtenidos y cantidad de votantes de la localidad. La lectura finaliza al leer la localidad 'Zárate', que debe procesarse. Informar:

- El intendente que obtuvo la mayor cantidad de votos en la elección.
- El intendente que obtuvo el mayor porcentaje de votos de la elección.

10. Un centro de investigación de la UNLP está organizando la información de las 320 especies de plantas con las que trabajan. Para cada especie se ingresa su nombre científico, tiempo promedio de vida (en meses), tipo de planta (por ej. árbol, conífera, arbusto, helecho, musgo, etc.), clima (templado, continental, subtropical, desértico, etc.) y países en el mundo donde se las encuentra. La información de las plantas se ingresa ordenada por tipo de planta y, para cada planta, la lectura de países donde se las encuentra finaliza al ingresar el país 'zzz'. Al finalizar la lectura, informar:

- El tipo de planta con menor cantidad de plantas.
- El tiempo promedio de vida de las plantas de cada tipo.
- El nombre científico de las dos plantas más longevas.
- Los nombres de las plantas nativas de Argentina que se encuentran en regiones con clima subtropical.
- El nombre de la planta que se encuentra en más países.

11. Una compañía de vuelos internacionales está analizando la información de todos los vuelos realizados por sus aviones durante todo el año 2019. De cada vuelo se conoce el código de avión, país de salida, país de llegada, cantidad de kilómetros recorridos y porcentaje de ocupación del avión. La información se ingresa ordenada por código de avión y, para cada avión, por país de salida. La lectura finaliza al ingresar el código 44. Informar:

- Los dos aviones que más kilómetros recorrieron y los dos aviones que menos kilómetros recorrieron.
- El avión que salió desde más países diferentes.
- La cantidad de vuelos de más de 5.000 km que no alcanzaron el 60% de ocupación del avión.
- La cantidad de vuelos de menos de 10.000 km que llegaron a Australia o a Nueva Zelanda.

12. En la “Práctica 1 - Ejercicios Adicionales” se resolvieron 3 problemas complejos sin utilizar módulos. Al carecer de herramientas para modularizar, esos programas resultaban difíciles de leer, de extender y de depurar. En la “Práctica 2 parte 2 - Ejercicios adicionales” se adaptaron los 3 problemas para utilizar módulos, y así organizar mejor el programa. Ahora podemos incluir los registros y así seguir mejorando nuestros programas. Para cada caso, analice:

- ¿qué entidades del programa conviene representar como registros?
- ¿qué atributos de cada entidad deben incluirse en los registros?

- ¿qué cambios deben realizarse en los módulos implementados en la práctica 2 para aprovechar los nuevos tipos de datos? ¿Conviene seguir utilizando los mismos módulos en todos los casos?

Una vez realizado el análisis, modifique los 3 problemas, utilizando registros para representar los datos del programa. Al finalizar cada problema, compare la solución usando registros y módulos con la solución sin registros y con módulos (práctica 2), y con la solución sin registros ni módulos (práctica 1).

- ¿Qué diferencias observa?
- ¿Qué similitudes encuentra?

CADP 2023

Práctica 4 (parte 1) - Vectores

1. Dado el siguiente programa:

```

1. program sumador;
2. type
3.   vnums = array [1..10] of integer;
4. var
5.   numeros : vnums;
6.   i : integer;
7. begin
8.   for i:=1 to 10 do {primer bloque for}
9.     numeros[i] := i;
10.
11.   for i := 2 to 10 do {segundo bloque for}
12.     numeros[i] := numeros[i] + numeros [i-1]
13. end.

```

- ¿Qué valores toma la variable **numeros** al finalizar el primer bloque **for**?
- Al terminar el programa, ¿con qué valores finaliza la variable **números**?
- Si se desea cambiar la línea 11 por la sentencia: **for i:=1 to 9 do** ¿Cómo debe modificarse el código para que la variable **números** contenga los mismos valores que en 1.b)?
- ¿Qué valores están contenidos en la variable **numeros** si la líneas 11 y 12 se reemplazan por:

```

for i:=1 to 9 do
  numeros[i+1] := numeros[i];

```

2. Dado el siguiente programa, complete las líneas indicadas, considerando que:

- El módulo **cargarVector** debe leer números reales y almacenarlos en el vector que se pasa como parámetro. Al finalizar, debe retornar el vector y su dimensión lógica. La lectura finaliza cuando se ingresa el valor 0 (que no debe procesarse) o cuando el vector está completo.
- El módulo **modificarVectorySumar** debe devolver el vector con todos sus elementos incrementados con el valor n y también debe devolver la suma de todos los elementos del vector.

<pre> program Vectores; const cant_datos = 150; type vdatos = array[1..cant_datos] of real; procedure cargarVector(var v:vdatos; var dimL : integer); var . . . { completar } begin . . . { completar } end; procedure modificarVectorySumar(var v:vdatos; dimL : integer; n: real; var suma: real); var . . . { completar } </pre>	<pre> { programa principal } var datos : vdatos; i, dim : integer; num, suma : real; begin dim := 0; sumaTotal := 0; cargarVector(...); { completar } writeln('Ingrese un valor a sumar'); readln(num); modificarVectorySumar(...);{completar} writeln('La suma de los valores es: ', suma); writeln('Se procesaron: ',dim, ' números') end. </pre>
---	--

```
begin
. . . { completar }
end;
```

3. Se dispone de un vector con números enteros, de dimensión física dimF y dimensión lógica dimL .
 - a) Realizar un módulo que imprima el vector desde la primera posición hasta la última.
 - b) Realizar un módulo que imprima el vector desde la última posición hasta la primera.
 - c) Realizar un módulo que imprima el vector desde la mitad ($\text{dimL DIV } 2$) hacia la primera posición, y desde la mitad más uno hacia la última posición.
 - d) Realizar un módulo que reciba el vector, una posición X y otra posición Y , e imprima el vector desde la posición X hasta la Y . Asuma que tanto X como Y son menores o igual a la dimensión lógica. Y considere que, dependiendo de los valores de X e Y , podría ser necesario recorrer hacia adelante o hacia atrás.
 - e) Utilizando el módulo implementado en el inciso anterior, vuelva a realizar los incisos a, b y c.
4. Se dispone de un vector con 100 números enteros. Implementar los siguientes módulos:
 - a) **posicion**: dado un número X y el vector de números, retorna la posición del número X en dicho vector, o el valor -1 en caso de no encontrarse.
 - b) **intercambio**: recibe dos valores x e y (entre 1 y 100) y el vector de números, y retorna el mismo vector donde se intercambiaron los valores de las posiciones x e y .
 - c) **sumaVector**: retorna la suma de todos los elementos del vector.
 - d) **promedio**: devuelve el valor promedio de los elementos del vector.
 - e) **elementoMaximo**: retorna la posición del mayor elemento del vector
 - f) **elementoMinimo**: retorna la posición del menor elemento del vector
5. Utilizando los módulos implementados en el ejercicio 4, realizar un programa que lea números enteros desde teclado (a lo sumo 100) y los almacene en un vector. La carga finaliza al leer el número 0. Al finalizar la carga, se debe intercambiar la posición del mayor elemento por la del menor elemento, e informe la operación realizada de la siguiente manera: *“El elemento máximo ... que se encontraba en la posición ... fue intercambiado con el elemento mínimo ... que se encontraba en la posición ...”*.
6. Dado que en la solución anterior se recorre dos veces el vector (una para calcular el elemento máximo y otra para el mínimo), implementar un único módulo que recorra una única vez el vector y devuelva ambas posiciones.
7. Realizar un programa que lea números enteros desde teclado hasta que se ingrese el valor -1 (que no debe procesarse) e informe:
 - a. la cantidad de ocurrencias de cada dígito procesado.
 - b. el dígito más leído.
 - c. los dígitos que no tuvieron ocurrencias.

Por ejemplo, si la secuencia que se lee es: 63 34 99 94 96 -1, el programa deberá informar:

Número 3: 2 veces

Número 4: 2 veces

Número 6: 2 veces

Número 9: 4 veces

El dígito más leído fue el 9.

Los dígitos que no tuvieron ocurrencias son: 0, 1, 2, 5, 7, 8

8. Realizar un programa que lea y almacene la información de 400 alumnos ingresantes a la Facultad de Informática de la UNLP en el año 2020. De cada alumno se lee: nro de inscripción, DNI, apellido, nombre y año de nacimiento. Una vez leída y almacenada toda la información, calcular e informar:
 - a) El porcentaje de alumnos con DNI compuesto sólo por dígitos pares.
 - b) Apellido y nombre de los dos alumnos de mayor edad.
9. Modificar la solución del punto anterior considerando que el programa lea y almacene la información de **a lo sumo** 400 alumnos. La lectura finaliza cuando se ingresa el DNI -1 (que no debe procesarse).
10. Realizar un programa que lea y almacene el salario de los empleados de una empresa de turismo (a lo sumo 300 empleados). La carga finaliza cuando se lea el salario 0 (que no debe procesarse) o cuando se completa el vector. Una vez finalizada la carga de datos se pide:
 - a) Realizar un módulo que incremente el salario de cada empleado en un 15%. Para ello, implementar un módulo que reciba como parámetro un valor real X, el vector de valores reales y su dimensión lógica y retorne el mismo vector en el cual cada elemento fue multiplicado por el valor X.
 - b) Realizar un módulo que muestre en pantalla el sueldo promedio de los empleados de la empresa.
11. El colectivo de fotógrafos ArgenPics desea conocer los gustos de sus seguidores en las redes sociales. Para ello, para cada una de las 200 fotos publicadas en su página de Facebook, cuenta con la siguiente información: título de la foto, el autor de la foto, cantidad de Me gusta, cantidad de clics y cantidad de comentarios de usuarios. Realizar un programa que lea y almacene esta información. Una vez finalizada la lectura, el programa debe procesar los datos e informar:
 - a) Título de la foto más vista (la que posee mayor cantidad de clics).
 - b) Cantidad total de Me gusta recibidas a las fotos cuyo autor es el fotógrafo “Art Vandelay”.
 - c) Cantidad de comentarios recibidos para cada una de las fotos publicadas en esa página.
12. En astrofísica, una galaxia se identifica por su nombre, su tipo (1. elíptica; 2. espiral; 3. lenticular; 4. irregular), su masa (medida en kg) y la distancia en pársecs (pc) medida desde la Tierra. La Unión Astronómica Internacional cuenta con datos correspondientes a las 53 galaxias que componen el Grupo Local. Realizar un programa que lea y almacene estos datos y, una vez finalizada la carga, informe:
 - a) La cantidad de galaxias de cada tipo.
 - b) La masa total acumulada de las 3 galaxias principales (la Vía Láctea, Andrómeda y Triángulo) y el porcentaje que esto representa respecto a la masa de todas las galaxias.
 - c) La cantidad de galaxias no irregulares que se encuentran a menos de 1000 pc.
 - d) El nombre de las dos galaxias con mayor masa y el de las dos galaxias con menor masa.
13. El Grupo Intergubernamental de Expertos sobre el Cambio Climático de la ONU (IPCC) realiza todos los años mediciones de temperatura en 100 puntos diferentes del planeta y, para **cada uno de estos lugares**, obtiene un promedio anual. Este relevamiento se viene realizando desde hace 50 años, y con todos los datos recolectados, el IPCC se encuentra en condiciones de realizar análisis estadísticos. Realizar un programa que lea y almacene los datos correspondientes a las mediciones de los últimos 50 años (la información se ingresa ordenada por año). Una vez finalizada la carga de la información, obtener:
 - a) El año con mayor temperatura promedio a nivel mundial.
 - b) El año con la mayor temperatura detectada en algún punto del planeta en los últimos 50 años.

14. El repositorio de código fuente más grande en la actualidad, GitHub, desea estimar el monto invertido en los proyectos que aloja. Para ello, **dispone** de una tabla con información de los desarrolladores que participan en un proyecto de software, junto al valor promedio que se paga por hora de trabajo:

CÓDIGO	ROL DEL DESARROLLADOR	VALOR/HORA (USD)
1	Analista Funcional	35,20
2	Programador	27,45
3	Administrador de bases de datos	31,03
4	Arquitecto de software	44,28
5	Administrador de redes y seguridad	39,87

Nota: los valores/hora se incluyen a modo de ejemplo

Realizar un programa que procese la información de los desarrolladores que participaron en los 1000 proyectos de software más activos durante el año 2017. De cada participante se conoce su país de residencia, código de proyecto (1 a 1000), el nombre del proyecto en el que participó, el rol que cumplió en dicho proyecto (1 a 5) y la cantidad de horas trabajadas. La lectura finaliza al ingresar el código de proyecto -1, que no debe procesarse. Al finalizar la lectura, el programa debe informar:

- a) El monto total invertido en desarrolladores con residencia en Argentina.
- b) La cantidad total de horas trabajadas por Administradores de bases de datos.
- c) El código del proyecto con menor monto invertido.
- d) La cantidad de Arquitectos de software de cada proyecto.

CADP 2023

Práctica 4 (parte 2) - Vectores

1.
 - a. Dado un vector de enteros de a lo sumo 500 valores, realice un módulo que reciba dicho vector y un valor n y retorne si n se encuentra en el vector o no.
 - b. Modifique el módulo del inciso a. considerando ahora que el vector se encuentra ordenado de manera ascendente.
2. Realice un programa que resuelva los siguientes incisos:
 - a. Lea nombres de alumnos y los almacene en un vector de a lo sumo 500 elementos. La lectura finaliza cuando se lee el nombre 'ZZZ', que no debe procesarse.
 - b. Lea un nombre y elimine la primera ocurrencia de dicho nombre en el vector.
 - c. Lea un nombre y lo inserte en la posición 4 del vector.
 - d. Lea un nombre y lo agregue al vector.

Nota: Realizar todas las validaciones necesarias.

3. Una empresa de transporte de caudales desea optimizar el servicio que brinda a sus clientes. Para ello, cuenta con información sobre todos los viajes realizados durante el mes de marzo. De cada viaje se cuenta con la siguiente información: día del mes (de 1 a 31), monto de dinero transportado y distancia recorrida por el camión (medida en kilómetros).
 - a. Realizar un programa que lea y almacene la información de los viajes (a lo sumo 200). La lectura finaliza cuando se ingresa una distancia recorrida igual a 0 km, que no debe procesarse.
 - b. Realizar un módulo que reciba el vector generado en a) e informe:
 - El monto promedio transportado de los viajes realizados
 - La distancia recorrida y el día del mes en que se realizó el viaje que transportó menos dinero.
 - La cantidad de viajes realizados cada día del mes.
 - c. Realizar un módulo que reciba el vector generado en a) y elimine todos los viajes cuya distancia recorrida sea igual a 100 km.
4. Una cátedra dispone de información de sus alumnos (a lo sumo 1000). De cada alumno se conoce nro de alumno, apellido y nombre y cantidad de asistencias a clase. Dicha información se encuentra ordenada por nro de alumno de manera ascendente. Se pide:
 - a. Un módulo que retorne la posición del alumno con un nro de alumno recibido por parámetro. El alumno seguro existe.
 - b. Un módulo que reciba un alumno y lo inserte en el vector.
 - c. Un módulo que reciba la posición de un alumno dentro del vector y lo elimine.
 - d. Un módulo que reciba un nro de alumno y elimine dicho alumno del vector
 - e. Un módulo que elimine del vector todos los alumnos con cantidad de asistencias en 0.

Nota: Realizar el programa principal que invoque los módulos desarrollados en los incisos previos con datos leídos de teclado.

5. La empresa Amazon Web Services (AWS) dispone de la información de sus 500 clientes monotributistas más grandes del país. De cada cliente conoce la fecha de firma del contrato con AWS, la categoría del monotributo (entre la A y la F), el código de la ciudad donde se encuentran las oficinales (entre 1 y 2400) y el monto mensual acordado en el contrato. La información se ingresa ordenada por fecha de firma de contrato (los más antiguos primero, los más recientes últimos).

Realizar un programa que lea y almacene la información de los clientes en una estructura de tipo vector. Una vez almacenados los datos, procesar dicha estructura para obtener:

- a. Cantidad de contratos por cada mes y cada año, y año en que se firmó la mayor cantidad de contratos
- b. Cantidad de clientes para cada categoría de monotributo
- c. Código de las 10 ciudades con mayor cantidad de clientes
- d. Cantidad de clientes que superan mensualmente el monto promedio entre todos los clientes.

EJERCICIOS ADICIONALES

1. La compañía Canonical Ltd. desea obtener estadísticas acerca del uso de Ubuntu Linux en La Plata. Para ello, debe realizar un programa que lea y almacene información sobre las computadoras con este sistema operativo (a lo sumo 10.000). De cada computadora se conoce: código de computadora, la versión de Ubuntu que utilizan (18.04, 17.10, 17.04, etc.), la cantidad de paquetes instalados y la cantidad de cuentas de usuario que poseen. La información debe almacenarse ordenada por código de computadora de manera ascendente. La lectura finaliza al ingresar el código de computadora -1, que no debe procesarse. Una vez almacenados todos los datos, se pide:
 - a. Informar la cantidad de computadoras que utilizan las versiones 18.04 o 16.04.
 - b. Informar el promedio de cuentas de usuario por computadora.
 - c. Informar la versión de Ubuntu de la computadora con mayor cantidad de paquetes instalados.
 - d. Eliminar la información de las computadoras con código entre 0 y 500.
2. Continuando con los 3 ejercicios adicionales de la **Guía opcional de actividades adicionales**, ahora utilizaremos vectores para almacenar la información ingresada por teclado. Consideraciones importantes:
 - Los datos ingresados por teclado deberán almacenarse en una estructura de tipo vector apropiada. Dado que en ninguno de los ejercicios se indica la cantidad máxima de datos a leer, para poder utilizar un vector asumimos que en todos los casos se ingresarán **a lo sumo 5000 datos** (donde cada dato será o bien una inversión, un alumno o un tanque de agua, según lo indica cada ejercicio).
 - Una vez leídos y almacenados los datos, deberán procesarse (recorrer el vector) para resolver cada inciso. Al hacerlo, deberán reutilizarse los módulos ya implementados en la práctica anterior. En la medida de lo posible, el vector **deberá recorrerse una única vez** para resolver todos los incisos.

CADP 2023

Práctica 5 – Punteros

PARTE CONCEPTUAL

- 1) ¿Qué se define como memoria estática?
- 2) ¿Qué se define como memoria dinámica?
- 3) ¿Qué es una variable de tipo puntero?
- 4) ¿Qué hace la operación “NEW” aplicada en una variable del tipo puntero?
- 5) ¿Qué hace la operación “DISPOSE” aplicada en una variable del tipo puntero?

PARTE PRÁCTICA

Para algunos ejercicios de la parte práctica, se utilizará la función de Pascal “**sizeof**”, que recibe como parámetro una variable de cualquier tipo y retorna la cantidad de bytes que dicha variable ocupa en la memoria principal.

Se presenta la siguiente tabla, que indica la cantidad de bytes que ocupa la representación interna de distintos tipos de datos en un compilador de Pascal típico.

Se recomienda **graficar** cada una de las situaciones planteadas a partir de una prueba de escritorio.

TIPO	CANTIDAD DE BYTES
Entero	2 bytes
Real	4 bytes
Char	1 byte
String	Tantos bytes como indique la longitud del String + 1
Record	La suma de las longitudes de los campos del registro
Puntero	4 bytes
Boolean	1 byte

Tabla de referencia de tamaño de los tipos de datos de Pascal (estos valores pueden variar entre diferentes implementaciones del compilador)

- 1) Indicar los valores que imprime el siguiente programa en Pascal. Justificar mediante prueba de escritorio.

```
program punteros;
type
  cadena = string[50];
  puntero_cadena = ^cadena;
var
  pc: puntero_cadena;
begin
  writeln(sizeof(pc), ' bytes');
  new(pc);
  writeln(sizeof(pc), ' bytes');
  pc^:= 'un nuevo nombre';
  writeln(sizeof(pc), ' bytes');
  writeln(sizeof(pc^), ' bytes');
  pc^:= 'otro nuevo nombre distinto al anterior';
  writeln(sizeof(pc^), ' bytes');
end.
```

- 2) Indicar los valores que imprime el siguiente programa en Pascal. Justificar mediante prueba de escritorio.

```

program punteros;
type
  cadena = string[9];
  producto = record
    codigo: integer;
    descripcion: cadena;
    precio: real;
  end;

  puntero_producto = ^producto;

var
  p: puntero_producto;
  prod: producto;
begin
  writeln(sizeof(p), ' bytes');
  writeln(sizeof(prod), ' bytes');
  new(p);
  writeln(sizeof(p), ' bytes');
  p^.codigo := 1;
  p^.descripcion := 'nuevo producto';
  writeln(sizeof(p^), ' bytes');
  p^.precio := 200;
  writeln(sizeof(p^), ' bytes');
  prod.codigo := 2;
  prod.descripcion := 'otro nuevo producto';
  writeln(sizeof(prod), ' bytes');
end.

```

- 3) Indicar los valores que imprime el siguiente programa en Pascal. Justificar mediante prueba de escritorio.

```

program punteros;
type
  numeros = array[1..10000] of integer;
  puntero_numeros = ^numeros;

var
  n: puntero_numeros;
  num: numeros;
  i: integer;
begin
  writeln(sizeof(n), ' bytes');
  writeln(sizeof(num), ' bytes');
  new(n);
  writeln(sizeof(n^), ' bytes');
  for i:= 1 to 5000 do
    n^[i] := i;
  writeln(sizeof(n^), ' bytes');
end.

```

- 4) Indicar los valores que imprimen los siguientes programas en Pascal. Justificar mediante prueba de escritorio.

```

a) program punteros;
type
  cadena = string[50];
  puntero_cadena = ^cadena;

```



```
var
  pc: puntero_cadena;
begin
  pc^:= 'un nuevo texto';
  new(pc);
  writeln(pc^);
end.

b) program punteros;
type
  cadena = string[50];
  puntero_cadena = ^cadena;
var
  pc: puntero_cadena;
begin
  new(pc);
  pc^:= 'un nuevo nombre';
  writeln(sizeof(pc^), ' bytes');
  writeln(pc^);
  dispose(pc);
  pc^:= 'otro nuevo nombre';
end.

c) program punteros;
type
  cadena = string[50];
  puntero_cadena = ^cadena;

procedure cambiarTexto(pun: puntero_cadena);
begin
  pun^:= 'Otro texto';
end;

var
  pc: puntero_cadena;
begin
  new(pc);
  pc^:= 'Un texto';
  writeln(pc^);
  cambiarTexto(pc);
  writeln(pc^);
end.

d) program punteros;
type
  cadena = string[50];
  puntero_cadena = ^cadena;

procedure cambiarTexto(pun: puntero_cadena);
begin
  new(pun);
  pun^:= 'Otro texto';
end;

var
  pc: puntero_cadena;
begin
  new(pc);
  pc^:= 'Un texto';
  writeln(pc^);
  cambiarTexto(pc);
  writeln(pc^);
end.
```

- 5) De acuerdo a los valores de la tabla que indica la cantidad de bytes que ocupa la representación interna de cada tipo de dato en Pascal, calcular el tamaño de memoria en los puntos señalados a partir de (I), suponiendo que las variables del programa ya están declaradas y se cuenta con 400.000 bytes libres. Justificar mediante prueba de escritorio.

```

Program Almacenamiento_Dinamico;
Type
    TEmpleado = record
        sucursal: char;
        apellido: string[25];
        correoElectrónico: string[40];
        sueldo: real;
    end;
    Str50 = string[50];

Var
    alguien: TEmpleado;
    PtrEmpleado: ^TEmpleado;
Begin

    {Suponer que en este punto se cuenta con 400.000 bytes de memoria disponible (I)}
    Readln( alguien.apellido );

    {Pensar si la lectura anterior ha hecho variar la cantidad de memoria (II)}
    New( PtrEmpleado );

    {¿Cuánta memoria disponible hay ahora? (III)}
    Readln( PtrEmpleado^.Sucursal, PtrEmpleado^.apellido );
    Readln( PtrEmpleado^.correoElectrónico, PtrEmpleado^.sueldo );

    {¿Cuánta memoria disponible hay ahora? (IV)}
    Dispose( PtrEmpleado );

    {¿Cuánta memoria disponible hay ahora? (V)}

end.

```

- 6) Realizar un programa que ocupe 50 KB de memoria en total. Para ello:
- El programa debe utilizar sólo memoria estática
 - El programa debe utilizar el 50% de memoria estática y el 50% de memoria dinámica
 - El programa debe minimizar tanto como sea posible el uso de la memoria estática (a lo sumo, 4 bytes)

- 7) Se desea almacenar en memoria una secuencia de 2500 nombres de ciudades, cada nombre podrá tener una longitud máxima de 50 caracteres.

a) Definir una estructura de datos estática que permita guardar la información leída. Calcular el tamaño de memoria que requiere la estructura.

b) Dado que un compilador de Pascal típico no permite manejar estructuras de datos estáticas que superen los 64 Kb, pensar en utilizar un vector de punteros a palabras, como se indica en la siguiente estructura:

```

Type  Nombre = String[50];
        Puntero = ^Nombre;
        ArrPunteros = array[1..2500] of Puntero;

Var
        Punteros: ArrPunteros;

```

- b.1)** Indicar cuál es el tamaño de la variable Punteros al comenzar el programa.
- b.2)** Escribir un módulo que permita reservar memoria para los 2500 nombres. ¿Cuál es la cantidad de memoria reservada después de ejecutar el módulo? ¿La misma corresponde a memoria estática o dinámica?
- b.3)** Escribir un módulo para leer los nombres y almacenarlos en la estructura de la variable Punteros.

8) Analice el siguiente programa:

```
1.  program memoria;
2.  type
3.    datos = array [1..20] of integer;
4.    punt = ^datos;
5.
6.  procedure procesarDatos(v : datos; var v2 : datos);
7.  var
8.    i, j : integer;
9.  begin
10.    for i := 1 to 20 do
11.      v2[21 - i] := v[i];
12.    end;
13.
14.  var
15.    vect : datos;
16.    pvect : punt;
17.    i : integer;
18.  begin
19.    for i:= 1 to 20 do
20.      vect[i] := i;
21.    new(pvect);
22.    for i:= 20 downto 1 do
23.      pvect^[i] := 0;
24.    procesarDatos(pvect^, vect);
25.    writeln('fin');
26.  end.
```

Responda: ¿cuánta memoria en total ocupa el programa al ejecutarse? Considere tanto variables estáticas como dinámicas, parámetros y variables locales de los módulos.

- a)** Hasta la sentencia de la línea 18
- b)** Hasta la sentencia de la línea 20
- c)** Hasta la sentencia de la línea 23
- d)** Hasta la sentencia de la línea 11
- e)** Hasta la sentencia de la línea 25

CADP 2023

Práctica 6 – Listas

1. Dado el siguiente programa:

<pre> program JugamosConListas; type lista = ^nodo; nodo = record num : integer; sig : lista; end; procedure armarNodo(var L: lista; v: integer); var aux : lista; begin new(aux); aux^.num := v; aux^.sig := L; L := aux; end; </pre>	<pre> var pri : lista; valor : integer; begin pri := nil; writeln('Ingrese un numero'); read(valor); while (valor <> 0) then begin armarNodo(pri, valor); writeln('Ingrese un numero'); read(valor); end; . . . { imprimir lista } end. </pre>
---	---

- Indicar qué hace el programa.
- Indicar cómo queda conformada la lista si se lee la siguiente secuencia de números: 10 21 13 48 0.
- Implementar un módulo que imprima los números enteros guardados en la lista generada.
- Implementar un módulo que reciba la lista y un valor, e incremente con ese valor cada dato de la lista.

2. Dado el siguiente código que lee información de personas hasta que se ingresa la persona con dni 0 y luego imprime dicha información en el orden inverso al que fue leída, identificar los 9 errores.

<pre> program ejercicio2; type lista = ^nodo; persona = record dni: integer; nombre: string; apellido: string; end; nodo = record dato: persona; sig: lista; end; procedure leerPersona(p: persona); begin read(p.dni); if (p.dni <> 0) then begin read(p.nombre); read(p.apellido); end; end; </pre>	<pre> {Carga la lista hasta que llega el dni 0} procedure generarLista(var l:lista); var p:nodo; begin leerPersona(p); while (p.dni <> 0) do begin agregarAdelante(l,p); end; end; procedure imprimirInformacion(var l:lista); begin while (l <> nil) do begin writeln('DNI: ', l^.dato.dni, 'Nombre:', l^.nombre, 'Apellido:', l^.apellido); l:= l^.sig; end; end; </pre>
--	---

<pre>{Agrega un nodo a la lista} procedure agregarAdelante(l:lista;p:persona); var aux: lista; begin aux^.dato:= p; aux^.sig:= l; l:= aux; end;</pre>	<pre>{Programa principal} var l:lista; begin generarLista(l); imprimirInformacion(l); end.</pre>
---	--

3. Utilizando el programa del ejercicio 1, realizar los siguientes cambios:

- Modificar el módulo *armarNodo* para que los elementos se guarden en la lista en el orden en que fueron ingresados (agregar atrás).
- Modificar el módulo *armarNodo* para que los elementos se guarden en la lista en el orden en que fueron ingresados, manteniendo un puntero al último ingresado.

4. Utilizando el programa del ejercicio 1, realizar los siguientes módulos:

- Máximo*: recibe la lista como parámetro y retorna el elemento de valor máximo.
- Mínimo*: recibe la lista como parámetro y retorna el elemento de valor mínimo.
- Múltiplos*: recibe como parámetros la lista L y un valor entero A, y retorna la cantidad de elementos de la lista que son múltiplos de A.

5. Realizar un programa que lea y almacene la información de productos de un supermercado. De cada producto se lee: código, descripción, stock actual, stock mínimo y precio. La lectura finaliza cuando se ingresa el código -1, que no debe procesarse. Una vez leída y almacenada toda la información, calcular e informar:

- Porcentaje de productos con stock actual por debajo de su stock mínimo.
- Descripción de aquellos productos con código compuesto por al menos tres dígitos pares.
- Código de los dos productos más económicos.

6. La Agencia Espacial Europea (ESA) está realizando un relevamiento de todas las sondas espaciales lanzadas al espacio en la última década. De cada sonda se conoce su nombre, duración estimada de la misión (cantidad de meses que permanecerá activa), el costo de construcción, el costo de mantenimiento mensual y rango del espectro electromagnético sobre el que realizará estudios. Dicho rango se divide en 7 categorías:

- radio;
- microondas;
- infrarrojo;
- luz visible;
- ultravioleta;
- rayos X;
- rayos gamma

Realizar un programa que lea y almacene la información de todas las sondas espaciales. La lectura finaliza al ingresar la sonda llamada "GAIA", que debe procesarse.

Una vez finalizada la lectura, informar:

- El nombre de la sonda más costosa (considerando su costo de construcción y de mantenimiento).
- La cantidad de sondas que realizarán estudios en cada rango del espectro electromagnético.
- La cantidad de sondas cuya duración estimada supera la duración promedio de todas las sondas.
- El nombre de las sondas cuyo costo de construcción supera el costo promedio entre todas las sondas.

Nota: para resolver los incisos a), b), c) y d), la lista debe recorrerse la menor cantidad de veces posible

7. El Programa Horizonte 2020 (H2020) de la Unión Europea ha publicado los criterios para financiar proyectos de investigación avanzada. Para los proyectos de sondas espaciales vistos en el ejercicio anterior, se han determinado los siguientes criterios:

- sólo se financiarán proyectos cuyo costo de mantenimiento no supere el costo de construcción.
- no se financiarán proyectos espaciales que analicen ondas de radio, ya que esto puede realizarse desde la superficie terrestre con grandes antenas.

A partir de la información disponible de las sondas espaciales (la lista generada en ejercicio 6), implementar un programa que:

- a. Invoque un módulo que reciba la información de una sonda espacial, y retorne si cumple o no con los nuevos criterios H2020.
- b. Utilizando el módulo desarrollado en a) implemente un módulo que procese la lista de sondas de la ESA y retorne dos listados, uno con los proyectos que cumplen con los nuevos criterios y otro con aquellos que no los cumplen.
- c. Invoque a un módulo que reciba una lista de proyectos de sondas espaciales e informe la cantidad y el costo total (construcción y mantenimiento) de los proyectos que no serán financiados por H2020. Para ello, utilice el módulo realizado en b.

8. Utilizando el programa del ejercicio 1, modificar el módulo *armarNodo* para que los elementos de la lista queden ordenados de manera ascendente (insertar ordenado).

9. Utilizando el programa del ejercicio 1, realizar los siguientes módulos:

- a. *EstáOrdenada*: recibe la lista como parámetro y retorna *true* si la misma se encuentra ordenada, o *false* en caso contrario.
- b. *Eliminar*: recibe como parámetros la lista y un valor entero, y elimina dicho valor de la lista (si existe). Nota: la lista podría no estar ordenada.
- c. *Sublista*: recibe como parámetros la lista L y dos valores enteros A y B, y retorna una nueva lista con todos los elementos de la lista L mayores que A y menores que B.
- d. Modifique el módulo Sublista del inciso anterior, suponiendo que la lista L se encuentra ordenada de manera ascendente.
- e. Modifique el módulo Sublista del inciso anterior, suponiendo que la lista L se encuentra ordenada de manera descendente.

10. Una empresa de sistemas está desarrollando un *software* para organizar listas de espera de clientes. Su funcionamiento es muy sencillo: cuando un cliente ingresa al local, se registra su DNI y se le entrega un número (que es el siguiente al último número entregado). El cliente quedará esperando a ser llamado por su número, en cuyo caso sale de la lista de espera. Se pide:

- a. Definir una estructura de datos apropiada para representar la lista de espera de clientes.
- b. Implementar el módulo *RecibirCliente*, que recibe como parámetro el DNI del cliente y la lista de clientes en espera, asigna un número al cliente y retorna la lista de espera actualizada.
- c. Implementar el módulo *AtenderCliente*, que recibe como parámetro la lista de clientes en espera, y retorna el número y DNI del cliente a ser atendido y la lista actualizada. El cliente atendido debe eliminarse de la lista de espera.
- d. Implementar un programa que simule la atención de los clientes. En dicho programa, primero llegarán todos los clientes juntos, se les dará un número de espera a cada uno de ellos, y luego se los atenderá de a uno por vez. El ingreso de clientes se realiza hasta que se lee el DNI 0, que no debe procesarse.

11. La Facultad de Informática debe seleccionar los 10 egresados con mejor promedio a los que la UNLP les entregará el premio Joaquín V. González. De cada egresado se conoce su número de alumno, apellido y el promedio obtenido durante toda su carrera.

Implementar un programa que:

- a. Lea la información de todos los egresados, hasta ingresar el código 0, el cual no debe procesarse.
- b. Una vez ingresada la información de los egresados, informe el apellido y número de alumno de los egresados que recibirán el premio. La información debe imprimirse ordenada según el promedio del egresado (de mayor a menor).

12. Una empresa desarrolladora de juegos para teléfonos celulares con Android dispone de información de todos los dispositivos que poseen sus juegos instalados. De cada dispositivo se conoce la versión de Android instalada, el tamaño de la pantalla (en pulgadas) y la cantidad de memoria RAM que posee (medida en GB).

La información disponible se encuentra ordenada por versión de Android. Realizar un programa que procese la información disponible de todos los dispositivos e informe:

- a. La cantidad de dispositivos para cada versión de Android.
- b. La cantidad de dispositivos con más de 3 GB de memoria y pantallas de a lo sumo a 5 pulgadas.
- c. El tamaño promedio de las pantallas de todos los dispositivos.

13. El Portal de Revistas de la UNLP está estudiando el uso de sus sistemas de edición electrónica por parte de los usuarios. Para ello, se dispone de información sobre los 3600 usuarios que utilizan el portal. De cada usuario se conoce su email, su rol (1: Editor; 2. Autor; 3. Revisor; 4. Lector), revista en la que participa y cantidad de días desde el último acceso.

- a. Imprimir el nombre de usuario y la cantidad de días desde el último acceso de todos los usuarios de la revista *Económica*. El listado debe ordenarse a partir de la cantidad de días desde el último acceso (orden ascendente).
- b. Informar la cantidad de usuarios por cada rol para todas las revistas del portal.
- c. Informar los emails de los dos usuarios que hace más tiempo que no ingresan al portal.

14. La oficina de becas y subsidios desea optimizar los distintos tipos de ayuda financiera que se brinda a alumnos de la UNLP. Para ello, esta oficina cuenta con un registro detallado de todos los viajes realizados por una muestra de 1300 alumnos durante el mes de marzo. De cada viaje se conoce el código de alumno (entre 1 y 1300), día del mes, Facultad a la que pertenece y medio de transporte (1. colectivo urbano; 2. colectivo interurbano; 3. tren universitario; 4. tren Roca; 5. bicicleta). Tener en cuenta que un alumno puede utilizar más de un medio de transporte en un mismo día.

Además, esta oficina cuenta con una tabla con información sobre el precio de cada tipo de viaje.

Realizar un programa que lea la información de los viajes de los alumnos y los almacene en una estructura de datos apropiada. La lectura finaliza al ingresarse el código de alumno -1, que no debe procesarse.

Una vez finalizada la lectura, informar:

- a. La cantidad de alumnos que realizan más de 6 viajes por día
- b. La cantidad de alumnos que gastan en transporte más de \$80 por día
- c. Los dos medios de transporte más utilizados.
- d. La cantidad de alumnos que combinan bicicleta con algún otro medio de transporte.

EJERCICIOS ADICIONALES

15. La cátedra de CADP está organizando la cursada para el año 2019. Para ello, dispone de una lista con todos los alumnos que cursaron EPA. De cada alumno se conoce su DNI, apellido, nombre y la nota obtenida. Escribir un programa que procese la información de alumnos disponible y los distribuya en turnos utilizando los siguientes criterios:

- Los alumnos que obtuvieron al menos 8 en EPA deberán ir a los turnos 1 ó 4.
- Los alumnos que obtuvieron entre 5 y 8 deberán ir a los turnos 2, 3 ó 5.
- Los alumnos que no alcanzaron la nota 5 no se les asignará turno en CADP.

Al finalizar, el programa debe imprimir en pantalla la lista de alumnos para cada turno.

Nota: La distribución de alumnos debe ser lo más equitativa posible.

16. La empresa distribuidora de una app móvil para corredores ha organizado una competencia mundial, en la que corredores de todos los países participantes salen a correr en el mismo momento en distintos puntos del planeta. La app registra, para cada corredor, el nombre y apellido, la distancia recorrida (en kilómetros), el tiempo (en minutos) durante el que corrió, el país y la ciudad desde donde partió, y la ciudad donde finalizó su recorrido. Realizar un programa que permita leer y almacenar toda la información registrada durante la competencia. La lectura finaliza al ingresar la distancia -1. Una vez que se han almacenado todos los datos, informar:

- o La cantidad total de corredores, la distancia total recorrida y el tiempo total de carrera de todos los corredores.
- o El nombre de la ciudad que convocó la mayor cantidad de corredores y la cantidad total de kilómetros recorridos por los corredores de esa ciudad.
- o La distancia promedio recorrida por corredores de Brasil.
- o La cantidad de corredores que partieron de una ciudad y finalizaron en otra ciudad.
- o El paso (cantidad de minutos por km) promedio de los corredores de la ciudad de Boston.

17. Continuando con los 3 ejercicios adicionales de la Guía opcional de actividades adicionales, ahora sumaremos lo aprendido sobre listas para almacenar la información ingresada por teclado. Consideraciones importantes:

- Los datos ingresados por teclado deberán almacenarse en una estructura de tipo lista apropiada.
- Una vez leídos y almacenados los datos, deberán procesarse (recorrer la lista) para resolver cada inciso. Al hacerlo, deberán reutilizarse los módulos ya implementados en las prácticas anteriores. En la medida de lo posible, la lista deberá recorrerse una única vez para resolver todos los incisos.

CADP 2023

Práctica 7 - Repaso

1. Una productora nacional realiza un casting de personas para la selección de actores extras de una nueva película, para ello se debe leer y almacenar la información de las personas que desean participar de dicho casting. De cada persona se lee: DNI, apellido y nombre, edad y el código de género de actuación que prefiere (1: drama, 2: romántico, 3: acción, 4: suspenso, 5: terror). La lectura finaliza cuando llega una persona con DNI 33555444, la cual debe procesarse.
Una vez finalizada la lectura de todas las personas, se pide:
 - a. Informar la cantidad de personas cuyo DNI contiene más dígitos pares que impares.
 - b. Informar los dos códigos de género más elegidos.
 - c. Realizar un módulo que reciba un DNI, lo busque y lo elimine de la estructura. El DNI puede no existir. Invocar dicho módulo en el programa principal.

2. Implementar un programa que lea y almacene información de clientes de una empresa aseguradora automotriz. De cada cliente se lee: código de cliente, DNI, apellido, nombre, código de póliza contratada (1..6) y monto básico que abona mensualmente. La lectura finaliza cuando llega el cliente con código 1122, el cual debe procesarse.
La empresa dispone de una tabla donde guarda un valor que representa un monto adicional que el cliente debe abonar en la liquidación mensual de su seguro, de acuerdo al código de póliza que tiene contratada.
Una vez finalizada la lectura de todos los clientes, se pide:
 - a. Informar para cada cliente DNI, apellido, nombre y el monto completo que paga mensualmente por su seguro automotriz (monto básico + monto adicional).
 - b. Informar apellido y nombre de aquellos clientes cuyo DNI contiene al menos dos dígitos 9.
 - c. Realizar un módulo que reciba un código de cliente, lo busque (seguro existe) y lo elimine de la estructura.

3. Una remisería dispone de información acerca de los viajes realizados durante el mes de mayo de 2020. De cada viaje se conoce: número de viaje, código de auto, dirección de origen, dirección de destino y kilómetros recorridos durante el viaje. Esta información **se encuentra ordenada** por código de auto y para un mismo código de auto pueden existir 1 o más viajes. Se pide:
 - a. Informar los dos códigos de auto que más kilómetros recorrieron.
 - b. Generar una lista nueva con los viajes de más de 5 kilómetros recorridos, ordenada por número de viaje.

4. Una maternidad dispone información sobre sus pacientes. De cada una se conoce: nombre, apellido y peso registrado el primer día de cada semana de embarazo (a lo sumo 42). La maternidad necesita un programa que analice esta información, determine e informe:
 - a. Para cada embarazada, la semana con mayor aumento de peso.
 - b. El aumento de peso total de cada embarazada durante el embarazo.

5. Una empresa de transporte de cargas dispone de la información de su flota compuesta por 100 camiones. De cada camión se tiene: patente, año de fabricación y capacidad (peso máximo en toneladas que puede transportar).
Realizar un programa que lea y almacene la información de los viajes realizados por la empresa. De cada viaje se lee: código de viaje, código del camión que lo realizó (1..100), distancia en kilómetros

recorrida, ciudad de destino, año en que se realizó el viaje y DNI del chofer. La lectura finaliza cuando se lee el código de viaje -1.

Una vez leída y almacenada la información, se pide:

1. Informar la patente del camión que más kilómetros recorridos posee y la patente del camión que menos kilómetros recorridos posee.
2. Informar la cantidad de viajes que se han realizado en camiones con capacidad mayor a 30,5 toneladas y que posean una antigüedad mayor a 5 años al momento de realizar el viaje (año en que se realizó el viaje).
3. Informar los códigos de los viajes realizados por choferes cuyo DNI tenga sólo dígitos impares.

Nota: Los códigos de viaje no se repiten.

6. El Observatorio Astronómico de La Plata ha realizado un relevamiento sobre los distintos objetos astronómicos observados durante el año 2015. Los objetos se clasifican en 7 categorías: 1: estrellas, 2: planetas, 3: satélites, 4: galaxias, 5: asteroides, 6: cometas y 7: nebulosas.

Al observar un objeto, se registran los siguientes datos: código del objeto, categoría del objeto (1..7), nombre del objeto, distancia a la Tierra (medida en años luz), nombre del descubridor y año de su descubrimiento.

A. Desarrolle un programa que lea y almacene la información de los objetos que han sido observados. Dicha información se lee hasta encontrar un objeto con código -1 (el cual no debe procesarse). La estructura generada debe mantener el orden en que fueron leídos los datos.

B. Una vez leídos y almacenados todos los datos, se pide realizar un reporte con la siguiente información:

1. Los códigos de los dos objetos más lejanos de la tierra que se hayan observado.
2. La cantidad de planetas descubiertos por "Galileo Galilei" antes del año 1600.
3. La cantidad de objetos observados por cada categoría.
4. Los nombres de las estrellas cuyos códigos de objeto poseen más dígitos pares que impares.

7. La Facultad de Informática desea procesar la información de los alumnos que finalizaron la carrera de Analista Programador Universitario. Para ello se deberá leer la información de cada alumno, a saber: número de alumno, apellido, nombres, dirección de correo electrónico, año de ingreso, año de egreso y las notas obtenidas en cada una de las 24 materias que aprobó (los aplazos no se registran).

1. Realizar un módulo que lea y almacene la información de los alumnos hasta que se ingrese el alumno con número de alumno -1, el cual no debe procesarse. Las 24 notas correspondientes a cada alumno deben quedar ordenadas de forma descendente.

2. Una vez leída y almacenada la información del inciso 1, se solicita calcular e informar:

- a. El promedio de notas obtenido por cada alumno.
- b. La cantidad de alumnos ingresantes 2012 cuyo número de alumno está compuesto únicamente por dígitos impares.
- c. El apellido, nombres y dirección de correo electrónico de los dos alumnos que más rápido se recibieron (o sea, que tardaron menos años)

3. Realizar un módulo que, dado un número de alumno leído desde teclado, lo busque y elimine de la estructura generada en el inciso 1. El alumno puede no existir.

8. Una entidad bancaria de la ciudad de La Plata solicita realizar un programa destinado a la administración de transferencias de dinero entre cuentas bancarias, efectuadas entre los meses de Enero y Noviembre del año 2018.

El banco dispone de una lista de transferencias realizadas entre Enero y Noviembre del 2018. De cada transferencia se conoce: número de cuenta origen, DNI de titular de cuenta origen, número de cuenta destino, DNI de titular de cuenta destino, fecha, hora, monto y el código del motivo de la transferencia

(1: alquiler, 2: expensas, 3: facturas, 4: préstamo, 5: seguro, 6: honorarios y 7: varios). Esta estructura no posee orden alguno.

Se pide:

- a) Generar una nueva estructura que contenga sólo las transferencias a terceros (son aquellas en las que las cuentas origen y destino no pertenecen al mismo titular). Esta nueva estructura debe estar ordenada por número de cuenta origen.

Una vez generada la estructura del inciso a), utilizar dicha estructura para:

- b) Calcular e informar para cada cuenta de origen el monto total transferido a terceros.
- c) Calcular e informar cuál es el código de motivo que más transferencias a terceros tuvo.
- d) Calcular e informar la cantidad de transferencias a terceros realizadas en el mes de Junio en las cuales el número de cuenta destino posea menos dígitos pares que impares.

9. Un cine posee la lista de películas que proyectará durante el mes de Febrero. De cada película se tiene: código de película, título de la película, código de género (1: acción, 2: aventura, 3: drama, 4: suspenso, 5: comedia, 6: bélica, 7: documental y 8: terror) y puntaje promedio otorgado por las críticas. Dicha estructura no posee orden alguno.

Se pide:

- a) Actualizar (en la lista que se dispone) el puntaje promedio otorgado por las críticas. Para ello se debe leer desde teclado las críticas que han realizado críticos de cine, de cada crítica se lee: DNI del crítico, apellido y nombre del crítico, código de película y el puntaje otorgado. La lectura finaliza cuando se lee el código de película -1 y la información viene ordenada por código de película.
- b) Informar el código de género que más puntaje obtuvo entre todas las críticas.
- c) Informar el apellido y nombre de aquellos críticos que posean la misma cantidad de dígitos pares que impares en su DNI.
- d) Realizar un módulo que elimine de la lista que se dispone una película cuyo código se recibe como parámetro (el mismo puede no existir).

10. Una compañía de venta de insumos agrícolas desea procesar la información de las empresas a las que les provee sus productos. De cada empresa se conoce su código, nombre, si es estatal o privada, nombre de la ciudad donde está radicada y los cultivos que realiza (a lo sumo 20). Para cada cultivo de la empresa se registra: tipo de cultivo (trigo, maíz, soja, girasol, etc.), cantidad de hectáreas dedicadas y la cantidad de meses que lleva el ciclo de cultivo.

- a. Realizar un programa que lea la información de las empresas y sus cultivos. Dicha información se ingresa hasta que llegue una empresa con código -1 (la cual no debe procesarse). Para cada empresa se leen todos sus cultivos, hasta que se ingrese un cultivo con 0 hectáreas (que no debe procesarse).

Una vez leída y almacenada la información, calcular e informar:

- b. Nombres de las empresas radicadas en "San Miguel del Monte" que cultivan trigo y cuyo código de empresa posee al menos dos ceros.
- c. La cantidad de hectáreas dedicadas al cultivo de soja y qué porcentaje representa con respecto al total de hectáreas.
- d. La empresa que dedica más tiempo al cultivo de maíz
- e. Realizar un módulo que incremente en un mes los tiempos de cultivos de girasol de menos de 5 hectáreas de todas las empresas que no son estatales.

11. Realizar un programa para una empresa productora que necesita organizar 100 eventos culturales. De cada evento **se dispone** la siguiente información: nombre del evento, tipo de evento (1: música, 2: cine, 3: obra de teatro, 4: unipersonal y 5: monólogo), lugar del evento, cantidad máxima de personas permitidas para el evento y costo de la entrada. Se pide:

1. Generar una estructura con las ventas de entradas para tales eventos culturales. De cada venta se debe guardar: código de venta, número de evento (1..100), DNI del comprador y cantidad de entradas adquiridas. La lectura de las ventas finaliza con código de venta -1.
2. Una vez leída y almacenada la información de las ventas, calcular e informar:
 - a. El nombre y lugar de los dos eventos que han tenido menos recaudación.
 - b. La cantidad de entradas vendidas cuyo comprador contiene en su DNI más dígitos pares que impares y que sean para el evento de tipo “obra de teatro”.
 - c. Si la cantidad de entradas vendidas para el evento número 50 alcanzó la cantidad máxima de personas permitidas.

12. El centro de deportes Fortaco’s quiere procesar la información de los 4 tipos de suscripciones que ofrece: 1) *Musculación*, 2) *Spinning*, 3) *Cross Fit*, 4) *Todas las clases*. Para ello, el centro **dispone** de una tabla con información sobre el costo mensual de cada tipo de suscripción.

Realizar un programa que lea y almacene la información de los clientes del centro. De cada cliente se conoce el nombre, DNI, edad y tipo de suscripción contratada (valor entre 1 y 4). Cada cliente tiene una sola suscripción. La lectura finaliza cuando se lee el cliente con DNI 0, el cual no debe procesarse.

Una vez almacenados todos los datos, procesar la estructura de datos generada, calcular e informar:

- La ganancia total de Fortaco’s
- Las 2 suscripciones con más clientes.
- Genere una lista con nombre y DNI de los clientes de más de 40 años que están suscritos a *CrossFit* o a *Todas las clases*. Esta lista debe estar ordenada por DNI.

13. La tienda de libros Amazon Books está analizando información de algunas editoriales. Para ello, Amazon cuenta con una tabla con las 35 áreas temáticas utilizadas para clasificar los libros (Arte y Cultura, Historia, Literatura, etc.).

De cada libro se conoce su título, nombre de la editorial, cantidad de páginas, año de edición, cantidad de veces que fue vendido y código del área temática (1..35).

Realizar un programa que:

- A) Invoque a un módulo que lea la información de los libros hasta ingresar el título “Relato de un naufrago” (que debe procesarse) y devuelva en una estructura de datos adecuada para la editorial “Planeta Libros”, con la siguiente información:
 - Nombre de la editorial
 - Año de edición del libro más antiguo
 - Cantidad de libros editados
 - Cantidad total de ventas entre todos los libros
 - Detalle con título, nombre del área temática y cantidad de páginas de todos los libros con más de 250 ventas.
- B) Invoque a un módulo que reciba la estructura generada en A) e imprima el nombre de la editorial y el título de cada libro con más de 250 ventas.

14. La biblioteca de la Universidad Nacional de La Plata necesita un programa para administrar información de préstamos de libros efectuados en marzo de 2020. Para ello, se debe leer la información de los préstamos realizados. De cada préstamo se lee: nro. de préstamo, ISBN del libro prestado, nro. de socio al que se prestó el libro, día del préstamo (1..31). La información de los préstamos se lee de manera ordenada por ISBN y finaliza cuando se ingresa el ISBN -1 (que no debe procesarse).

Se pide:

- A) Generar una estructura que contenga, para cada ISBN de libro, la cantidad de veces que fue prestado. Esta estructura debe quedar ordenada por ISBN de libro.
- B) Calcular e informar el día del mes en que se realizaron menos préstamos.
- C) Calcular e informar el porcentaje de préstamos que poseen *nro. de préstamo* impar y *nro. de socio* par.