

UNIVERSIDAD NACIONAL DE COLOMBIA

SECCIONAL MANIZALES

FACULTAD DE CIENCIAS Y ADMINISTRACIÓN



ALGORITMOS ESTRUCTURADOS

ADMINISTRADOR DE EMPRESAS

ALONSO TAMAYO ALZATE

MANIZALES, COLOMBIA

© **ALGORITMOS ESTRUCTURADOS**

Autor: ADMINISTRADOR DE EMPRESAS ALONSO TAMAYO ALZATE

**Universidad Nacional de Colombia
Seccional Manizales**

**1ª. Edición
Febrero de 1991
300 Ejemplares**

**Reservados todos los derechos.
Prohibida la reproducción total o parcial
de esta obra sin la autorización escrita
del Autor.**

ISBN · 958 - 95323 - 2 - 2

**Darío Valencia Restrepo
Rector**

**Carlos Enrique Ruíz
Vicerrector Seccional**

**Luz Stella Cortés Gálvez
Jefe Centro Publicaciones**

**Impreso en los Talleres del
Centro de Publicaciones de la
Universidad Nacional de Colombia
Seccional Manizales
Apartado Aéreo 127
Manizales Colombia**

CONTENIDO

CAPITULO I. CONCEPTOS BASICOS

1.1	INTRODUCCION	6
1.2	GENERALIDADES	6
1.3	DEFINICION DE ALGORITMO	7
1.4	PROBLEMAS PROPUESTOS	12
1.5	RECOMENDACIONES A SEGUIR PARA LA SOLUCION DE UN PROBLEMA.	14
1.6	CONSIDERACIONES ADICIONALES	15
1.6.1	Constante	15
1.6.2	Variable	15
1.6.3	Operación de Asignación	16
1.6.4	Operadores Aritméticos	17
1.6.5	Operadores Relacionales	18
1.6.6	Operadores Lógicos	18
1.6.7	Expresiones	18
1.6.8	Expresión Aritmética	18
1.6.9	Expresión Lógica	18
1.6.10	Reglas para Expresiones Aritméticas	19
1.6.11	Jerarquía de las Operaciones Aritméticas	20
1.7	EJERCICIOS PROPUESTOS	21

CAPITULO II. ESTRUCTURAS ALGORITMICAS

2.1	INTRODUCCION	24
2.2	CLASIFICACION DE LOS ALGORITMOS	25
2.2.1	Directos	25
2.2.2	Iterativos o Ciclicos	25
2.2.3	Anidados	25
2.3	ESTRUCTURAS ALGORITMICAS	28
2.3.1	Estructura Secuencial	28
2.3.2	Estructuras Condicionales	32
2.3.2.1	Estructura IF_THEN (SI_ENTONCES)	33
2.3.2.1.1	Prueba de Escritorio	37
2.3.2.2	Estructura IF_THEN_ELSE (SI_ENTONCES_SINO)	38
2.3.2.2.1	Estructura IF anidada	42
2.3.2.2.2	Problemas propuestos	46
2.3.2.3	Estructura de Decisión Múltiple CASE	47
	(EN CASO DE - HACER)	
2.3.2.3.1	Problemas Propuestos	52
2.3.3	Estructuras Repetitivas o Cíclicas	55
2.3.3.1	Estructura WHILE_DO (MIENTRAS_HACER)	56
2.3.3.1.1	Acumulador	60
2.3.3.1.2	Ejercicios Propuestos	66
2.3.3.1.3	Contador	66
2.3.3.2	Estructura REPEAT_UNTIL (REPETIR_HASTA_QUE)	73
2.3.3.3	Estructura FOR (PARA)	81
2.3.3.3.1	Problemas Propuestos	88

2.3.4 Estructuras Repetitivas Anidadas	91
2.4 REGISTRO DUMMY O CENTINELA	106
2.5 PROBLEMAS RESUELTOS	112
2.6 PROBLEMAS PROPUESTOS	141

CAPITULO III. ARREGLOS

3.1 INTRODUCCION	144
3.2 DEFINICION	144
3.3 PROBLEMAS PROPUESTOS	164
3.4 PROBLEMAS RESUELTOS	168
3.5 ARREGLOS DE DOS DIMENSIONES	186
3.6 PROBLEMAS RESUELTOS	195
3.7 PROBLEMAS PROPUESTOS	212

CAPITULO IV. INTRODUCCION AL DISEÑO DESCENDIENTE

4.1 INTRODUCCION	214
4.2 PROGRAMACION MODULAR	214
4.3 CLASIFICACION DE LOS MODULOS	215
4.3.1 Funciones	215
4.3.2 Procedimientos	225
4.4 PROBLEMAS RESUELTOS	243
4.5 PROBLEMAS PROPUESTOS	269

PROLOGO

El desarrollo del texto se circunscribe al conocimiento de las herramientas fundamentales para la solución de problemas, ofreciendo especial importancia al análisis y por consiguiente al diseño y solución algorítmica de los ejercicios planteados.

Este material se proyectó para los cursos iniciales de programación de computadores, tales como:

- Introducción a la programación.
- Fundamentos de computación.
- Lenguajes de programación.
- Lógica computacional.
- Informática.

VISION GENERAL

El capítulo 1 da una introducción a la temática de los Algoritmos y a los conocimientos básicos, necesarios para la comprensión de los demás temas a tratar.

El capítulo 2 brinda un desarrollo pormenorizado de los elementos indispensables para el ofrecimiento de soluciones

algorítmicas a los problemas propuestos. El avance del mismo es deliberadamente lento con el propósito de destacar las diferentes herramientas necesarias para resolver los problemas que se aplican en el resto del texto. Concluido este capítulo, el estudiante estará en capacidad de darle respuesta a la gran mayoría de los problemas planteados.

El capítulo 3 es más ágil puesto que se soporta en los dos anteriores, se ofrecen allí elementos de gran ayuda en la programación de computadores, como los temas de Arreglos de una y dos dimensiones que son los de mayor utilización en la vida práctica.

El cuarto y último capítulo incluye el estudio del tema Diseño Descendente, sustentado en la programación modular, soporte básico y fundamental de la programación estructurada.

Desde el punto de vista metodológico, se incluye una completa gama de conceptos soportados con bastantes ejercicios resueltos y explicados, que tienen un cubrimiento desde niveles elementales hasta intermedios de complejidad, encontrándose en algunos de ellos, varias soluciones a un mismo ejercicio, unas más ágiles y prácticas que otras.

En conclusión, lo fundamental de éste trabajo es propiciar la creatividad y el análisis, ántes que la memorización de soluciones a un determinado ejercicio.

El Autor.

CAPITULO I

CONCEPTOS BASICOS

1.1 INTRODUCCION

Más que entrar directamente en el tema, lo que se persigue con éste capítulo es introducir algunos conceptos elementales e importantes para el mejor entendimiento de los próximos temas a tratar, necesarios en la solución de problemas con el computador; pero en ningún momento se entra en especificidades de algún lenguaje de programación, con el propósito de darle libertad al estudiante en su futura elección.

1.2 GENERALIDADES

La palabra Algoritmo tiene su origen en el nombre del matemático árabe del siglo IX llamado MOHAMMED AL-KHOWARIZMI, quien enunció las reglas a seguir para la realización de las operaciones aritméticas básicas como la suma, resta, multiplicación y división, así mismo desarrolló varios métodos para la solución de problemas con ecuaciones. La traducción al latín de su apellido "KHOWARIZMI" en ALGORITMUS, derivó en la palabra ALGORITMO.

1.3 DEFINICION

Algoritmo es un conjunto de acciones precisas y lógicas que se deben realizar en un orden determinado, para así obtener la solución a un problema en un número finito de pasos. Un ejemplo obvio de algoritmo lo constituye la apertura de una caja fuerte; o el seguimiento de los instructivos que traen los juegos como el parqués, damas o ajedrez, entre otros, o aún más simple lo es una receta de cocina.

Un ejemplo sutil consiste en hacer una llamada telefónica, el cual se puede expresar en seis pasos sencillos a saber:

Paso 1	INICIO
Paso 2	Levantar la bocina
Paso 3	Marcar número
Paso 4	Hablar
Paso 5	Descargar bocina
Paso 6	FIN

Esta solución, aunque buena, no es la más apropiada por carecer de exactitud, tal es el caso de no contar con sucesos como: no encontrar tono, estar ocupado el número solicitado ó simplemente no contestan.

Lo anterior conlleva a que un Algoritmo debe reunir las siguientes características:

- No se debe prestar a ambigüedades.
- El número de pasos debe ser cuantificable, es decir,

finito.

- Debe alcanzar el fin propuesto, es decir, efectivo.

Una segunda versión del ejercicio se puede presentar así:

Paso 1	INICIO
Paso 2	Levantar bocina hasta obtener tono
Paso 3	Marcar número
Paso 4	Esperar hasta que contesten la llamada
Paso 5	Hablar
Paso 6	Descargar la bocina
Paso 7	FIN

Pero se continúa con ambigüedades como se aprecia en los pasos 2 y 4. No se debe mantener levantada la bocina esperando tono, porque el tiempo transcurrido puede ser demasiado, por lo tanto no es práctico. Igualmente sucede con el paso 4, en donde no se debe esperar hasta que contesten la llamada porque se puede dar el caso de no recibir contestación.

Lo anterior obliga a analizar el ejercicio con un mayor nivel de profundidad, evitando así que se presenten situaciones indefinidas en un momento dado.

La tercera versión del ejercicio muestra la simplificación de actividades en operaciones más detalladas, buscando

siempre dar mayor exactitud y claridad a cada acción que se deba ejecutar.

- Paso 1** **INICIO**
- Paso 2** Levantar bocina
- Paso 3** Hay tono?
- Si no hay tono, repítase el siguiente proceso hasta lograrlo:
- * Descargar bocina
 - * Esperar un tiempo prudente
 - * Levantar bocina nuevamente
- Paso 4** Marcar número
- Paso 5** Contestan?
- Si no contestan, repítase el siguiente proceso hasta lograrlo:
- * Esperar un tiempo prudente
 - * Interrumpir comunicación
 - * Marcar número nuevamente
- Paso 6** Hablar
- Paso 7** Descargar bocina
- Paso 8** **FIN**

En el anterior algoritmo aparece el concepto de **DECISION** representado en las preguntas formuladas, las cuales tienen dos respuestas, una cierta o positiva y la otra falsa o negativa.

En el paso 3, si la respuesta es positiva continúa con el paso 4, en caso contrario aparece otro concepto y es el de **REPETICION**, el cual agrupa una ó más actividades que se repiten hasta satisfacer una condición determinada.

Entre mayor conocimiento se tenga de una situación o problema, más facilidad se presentará en el momento de darle solución algorítmica. El ejercicio anterior ofrece la solución al problema que se plantea con un mínimo de condiciones o tal vez ninguna, pero no siempre se van a presentar problemas tan elementales y es así como el grado de complejidad aumenta, en la medida en que entran a participar de la problemática un mayor número de restricciones.

El ejercicio que se ha estado analizando, cambia ostensiblemente si tiene en cuenta las siguientes alternativas:

- * Es teléfono público ?
- * Se dispone de cambio suficiente ?
- * Es teléfono directo ?
- * Es llamada local ?

EJERCICIO

Conocidos el nombre y edad de una persona, imprimir el nombre de ella sólo si es mayor de edad.

Solución

a) Análisis

Para determinar si la persona es mayor de edad, se compara la edad contra 18 años.

b) Algoritmo

Paso 1 INICIO

Paso 2 Conocer nombre y edad

Paso 3 Es edad mayor de 18 años ?

 Si no, acabar el ejercicio

 Si si, Imprimir el nombre

Paso 4 FIN

EJERCICIO

Conocido un número, calcular su raíz cuadrada. Si el número es negativo, imprimir el número y un mensaje que indique "tiene raíz imaginaria".

Solución

a) Análisis

Para determinar si una cantidad es negativa, es preciso compararla contra cero.

b) Algoritmo

Paso 1 INICIO

Paso 2 Conocer N

Paso 3 Es N menor que cero ?

 Si sí, imprimir N, "tiene raiz imaginaria"

 Si no, $RAIZ = \sqrt{N}$

 Imprimir N, RAIZ

Paso 4 FIN

1.4 PROBLEMAS PROPUESTOS

1- Ordenar las siguientes actividades en forma lógica.

- * Tiene cigarrillos ?
- * Comprar fósforos.
- * Empezar.
- * Tiene fósforos ?
- * Terminar.
- * Tomar un cigarrillo.
- * Encender fósforo y prender cigarrillo.
- * Comprar cigarrillos.

2- Ordenar las siguientes actividades en la forma más conveniente, con el propósito de abastecer un carro con gasolina.

- * Llenar tanque de gasolina.
- * Desea gasolina corriente ?
- * Empezar.
- * Llevar carro a la bomba de gasolina.
- * Terminar.

- * Pagar consumo de gasolina. ⁶
- * Utilizar gasolina extra. ⁶
- * Consumir lo deseado. ⁷

3- Elaborar un algoritmo para cambiar la llanta de un vehículo.

4- Elaborar un algoritmo para despachar una encomienda por correo.

5- Elaborar un algoritmo que indique como sembrar una planta en el jardín.

6- Conocidos dos números o cantidades, elaborar un algoritmo para calcular la suma, resta y producto de ellos.

7- Elaborar un algoritmo para calcular el neto a pagar a un empleado de una compañía, en donde se conoce el valor pagado por cada día trabajado y el número de días laborados en el mes. Las deducciones corresponden al 3% del salario bruto.

8- Calcular el área del triángulo según las siguientes fórmulas:

$$\text{AREA} = \sqrt{S (S-A) (S-B) (S-C)}$$

$$\text{donde } S = (A + B + C) / 2$$

9- Leer código, nombre y salario de un trabajador e imprimir la información leída si el trabajador tiene un salario menor o igual a \$50.000=, en caso contrario, imprimir el código, nombre del trabajador y un mensaje que diga "gana más de \$50.000="

1.5 RECOMENDACIONES A SEGUIR PARA LA SOLUCION DE UN PROBLEMA

1- Entender el problema claramente.

Es necesario leer el enunciado del problema que se va a resolver, dos, tres o más veces hasta tanto se entienda. Sería inútil tratar de darle solución a un problema, cuando ni siquiera el enunciado se entiende. Entre mayor claridad se tenga del enunciado, más facilidad se presentará en la solución.

2- Buscar la solución del problema.

- Determinar o definir la información requerida.
- Determinar los cálculos o cálculos necesarios.
- Definir la información que va a producir el sistema.

Se resumen estas tres actividades como:



Es necesario establecer prioridades, porque así como hay cálculos que se tendrán que realizar primero que otros, así mismo ocurrirá tanto para la información de entrada, como para la información de salida.

3- Definir variables.

Definir las variables a utilizar en el algoritmo.

4- Presentar la solución algorítmica.

1.6 CONSIDERACIONES ADICIONALES

1.6.1 CONSTANTE: Es una cantidad fija en donde el valor no cambia durante la ejecución del problema.

1.6.2 VARIABLE: Como su nombre indica, representa un valor susceptible de modificación. A las variables se les da nombre para poderlas identificar. Este nombre tiene restricciones en cuanto a su longitud se refiere, por ejemplo, algunos lenguajes de programación (compiladores) aceptan variables de 5 caracteres de longitud máxima, otros de 6, otros de 8, etc, pero siempre el primer caracter debe ser una letra y no puede contener caracteres especiales como * , ; / + -

() @ > ? \$ etc., ni espacios en blanco.
 Lo recomendable en este caso es que debe hacer referencia con lo que representa, es decir, debe ser nemotécnica.

Ejemplos Válidos:

A	NETO	A124X
XYZ	SALARIO	X1Z
KARLA	PESO	K123

Ejemplos Incorrectos:

1XYZ	No comienza por letra,
A\$B	Contiene caracter especial
TOTAL NETO	Contiene espacio en blanco

1.6.3 OPERACION DE ASIGNACION

Se produce cuando a una variable se le señala un valor determinado, bien puede ser por asignación directa, o como resultante de la evaluación de una expresión.

La operación de asignación se denota por el símbolo ←, e indica que el valor o variable que está a la derecha del símbolo de asignación, se almacena en la variable que está a la izquierda del mismo símbolo. Si en lugar de tener un valor o variable tiene una expresión, ella se almacena de

idéntica forma, una vez sea evaluada.

Ejemplos:

$K \leftarrow 3$ Indica que a la variable K se le ha asignado un valor de 3.

$C \leftarrow A+B$ Indica que el resultado de sumar A+B se almacene en la variable C.

$L \leftarrow M$ Indica que el contenido de la variable M se almacene en la variable L.

$Z \leftarrow Z+1$ Indica que el valor de la variable Z se incremente en la unidad y el resultado se almacene nuevamente en la misma variable Z.

Al lado izquierdo del símbolo de asignación irá siempre una variable, no se admite una constante , ni tampoco otra expresión.

Ejemplos incorrectos:

$A + B * C \leftarrow D$

$57 \leftarrow E$

$X * Y \leftarrow Z + W$

1.6.4 OPERADORES ARITMETICOS

- + Suma ó adición
- Resta ó sustracción
- * Multiplicación
- / División

↑ ^ Potenciación ó Exponenciación

1.6.5 OPERADORES RELACIONALES

> Mayor que
 >= Mayor o igual
 < Menor
 <= Menor o igual
 <> No igual o diferente

1.6.6 OPERADORES LOGICOS

.AND. CONJUNCION LOGICA
 .OR. DISYUNCION LOGICA
 .NOT. NEGACION LOGICA

1.6.7 EXPRESIONES

Son constantes y variables unidas entre sí por operadores lógicos y aritméticos. Se usan para expresar cálculos y tienen como función asignar valores a una variable.

1.6.8 EXPRESION ARITMETICA

Es una secuencia de constantes numéricas, variables y operadores aritméticos que indican la cantidad de cálculos a efectuar.

1.6.9 EXPRESION LÓGICA

Está conformada por la combinación de elementos lógicos y

operadores lógicos. Un elemento lógico puede ser una constante lógica, variable lógica, o expresión relacional, ésta última constituida por la combinación de dos expresiones aritméticas con un operador relacional.

1.6.10 REGLAS PARA EXPRESIONES ARITMETICAS

- Las cantidades pueden estar precedidas por los signos más o menos (+,-), ó pueden estar conectados por cualquiera de los símbolos operacionales (*,/,+,-).
- Una expresión puede contener cantidades enteras, cantidades reales, o ambas.
- Dos símbolos de operación no deben aparecer consecutivamente.

Ejemplos:

forma incorrecta

$$A / - B$$

$$A + - B$$

forma correcta

$$A / (-B)$$

$$A + (-B)$$

- Los símbolos operacionales no son asumidos, por lo tanto, no deben aparecer consecutivamente.

Ejemplos:

forma incorrecta

3I

A(B + C)

B² - 4AC

forma correcta

3 * I

A * (B + C)

B² - 4 * A * C

- Los paréntesis sólo indican agrupación y no multiplicación.
- Las expresiones ambiguas se deben aclarar con el uso del paréntesis.

Ejemplo:

$$\frac{BC}{A} \quad \text{Se expresa como} \quad A \uparrow (B * C)$$

1.6.11 JERARQUIA DE LAS OPERACIONES ARITMETICAS

- 1- Evaluación de las operaciones encerradas entre paréntesis y dentro de éstos, los paréntesis más internos.
- 2- Evaluación de funciones
- 3- Exponenciación o potenciación
- 4- Multiplicación y división
- 5- Suma y resta

Cuando aparece en una expresión dos operaciones que son del mismo orden o prioridad, se procede a evaluar de izquierda a derecha.

Es de aclarar que no todos los lenguajes de programación (compiladores) proceden de acuerdo a ésta jerarquía, pero es la gran mayoría.

Ejemplos:

Expresión algebraica

Expresión computacional

$$A = \frac{X + Y}{(C)^5}$$

$$A \leftarrow (X+Y) / C \uparrow 5$$

$$D = (H/5) (X+Y+Z)$$

$$D \leftarrow H * (X+Y+Z) / 5$$

$$L = 80 W^2 + 60 W - 5$$

$$L \leftarrow 80 * W \uparrow 2 + 60 * W - 5$$

$$Q = x^5 / 5!$$

$$Q \leftarrow X \uparrow 5 / 120$$

$$Y = \sqrt{B^2 - 4AC}$$

$$Y \leftarrow (B \uparrow 2 - 4 * A * C) \uparrow 0.5$$

$$Y = \sqrt{A} / (4 + \sqrt{A})$$

$$Y \leftarrow A \uparrow (1/2) / (4 + A \uparrow (1/2))$$

$$Z = (A + B) / (7 - C)$$

$$Z \leftarrow (A + B) / (7 - C)$$

1.7 EJERCICIOS PROPUESTOS

Reemplazar las siguientes expresiones algebraicas con expresiones computacionales:

1- $A = X + (Y/Z) + W$

2- $B = X + 1 + (X^5 / 5!) + (X^7 / 7!)$

3- $C = [(X + Y) / (X - Y)]^3 - 3$

4- $D = (X + Y) / [W + A / (B + C)]$

5- $E = (X / Y)^{W - 3}$

Las siguientes expresiones computacionales corresponden a expresiones matemáticas, pero se encuentran erradas. Escribir las expresiones correctas.

EXPRESION COMPUTACIONAL	EXPRESION MATEMATICA
1- $X \leftarrow A/B + (C+D)\uparrow 2$	$X = A / (B + (C + D)^2)$
2- $Y \leftarrow A+7 / B-3$	$Y = (A+7) / (B-3)$
3- $W \leftarrow (B+3.14159)/(B+7)\uparrow 5$	$W = [(B+\pi) / (B+7)]^5$
4- $Z \leftarrow 9/(1/X) - (4/Y)$	$Z = 9 / [(1/X) - (4/Y)]$
5- $X \leftarrow A\uparrow L+2+B\uparrow L+4$	$X = A^{L+2} + B^{L+4}$

EJERCICIO

Sea $X = 10$

$$Y = 5$$

$$Z = 2$$

$$W = 3$$

Evaluar las siguientes expresiones:

$$1- A \leftarrow X + Y \uparrow Z * W$$

Solución

$$A = 10 + 5^2 * 3$$

$$A = 10 + 25 * 3$$

$$A = 10 + 75$$

$$A = 85$$

$$2- \quad B \leftarrow (X * (Y+Z)) / W$$

Solución

$$B = (10 * (5+2)) / 3$$

$$B = (10 * 7) / 3$$

$$B = 70 / 3$$

$$B = 23.33$$

$$3- \quad C \leftarrow (X * Y - Z) / W$$

Solución

$$C = (10 * 5 - 2) / 3$$

$$C = (50 - 2) / 3$$

$$C = 48 / 3$$

$$C = 16$$

EJERCICIOS PROPUESTOS

Evaluar las siguientes expresiones:

$$4- \quad D \leftarrow X * (Y * Z / W)$$

$$5- \quad E \leftarrow X * (Y - Z) / W$$

$$6- \quad F \leftarrow X * Y - Z / W$$

CAPITULO II

ESTRUCTURAS ALGORITMICAS

2.1 INTRODUCCION

Todo problema de computación tiene dos fases muy demarcadas como son la solución del problema, que se describe por medio de un algoritmo y la puesta en marcha del mismo, que se expresa en un lenguaje cualquiera de programación, convirtiéndose ésta en una labor sencilla y rutinaria, pero no tan importante como la primera etapa, porque el éxito o fracaso de una aplicación radica en el diseño algorítmico, es por eso que éste material se dedica a la enseñanza de esa difícil pero loable tarea.

Dado lo anterior, se ha procurado no vincular el lenguaje algorítmico con un lenguaje de programación en particular o equipo de computación en específico, tratando de conservar siempre la transparencia, característica que debe mantener todo algoritmo, aunque existen influencias obvias, y dejando la decisión final relacionada con la escogencia del lenguaje de programación al usuario final.

2.2 CLASIFICACION DE LOS ALGORITMOS.

Los algoritmos se clasifican en tres grandes grupos a saber:

2.2.1 **DIRECTOS:** Conformado por el seguimiento de una serie de pasos elementales secuenciales, consistente en la mayoría de las veces de una entrada seguida de uno o más procesos, incluyendo decisiones, y terminando con una salida.

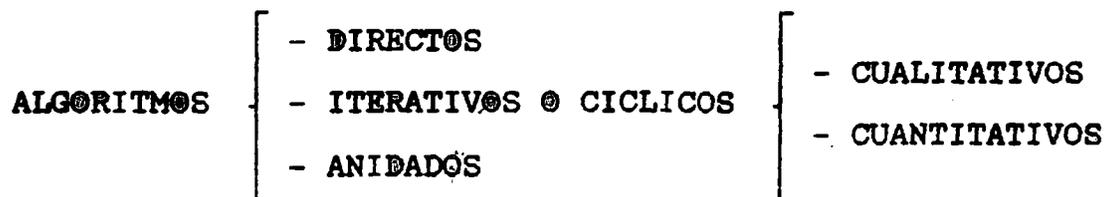
2.2.2 **ITERATIVOS O CICLICOS:** Cuando se repite una serie de pasos o actividades hasta satisfacer una condición pre-determinada.

2.2.3 **ANIDADOS:** Cuando dentro de las instrucciones que comprende un ciclo, están incluidos a su vez otros ciclos o iteraciones.

La solución a un problema se puede presentar en forma cualitativa o cuantitativa, sin importar su clasificación.

Cualitativa: Como su nombre lo indica, se expresa por medio de palabras.

Cuantitativa: Cuando en la solución se involucran cálculos numéricos.

CUADRO SINOPTICO:**Ejemplo de algoritmo cualitativo.**

Enunciado: Elaborar un algoritmo para ingresar a la Universidad a realizar estudios superiores.

Solución**a) Algoritmo**

- | | |
|---------------|--|
| Paso 1 | INICIO |
| Paso 2 | Reune todos los requisitos ?
Si no, completar requisitos. |
| Paso 3 | Comprar formulario |
| Paso 4 | Diligenciar formulario |
| Paso 5 | Formulario bien diligenciado ?
Si no, repetir proceso de diligenciamiento hasta que sea correcto. |
| Paso 6 | Presentar examen de admisión |
| Paso 7 | Ganó examen ?
Si no, repetir proceso el próximo semestre.
Si sí, ingresar a la Universidad. |
| Paso 8 | FIN |

Ejemplo de algoritmo cuantitativo.

Enunciado: Conocidos dos números o cantidades, calcular la suma, resta y producto de ellos.

Solución

a) Algoritmo

Paso 1	INICIO
Paso 2	Conocer los números A y B
Paso 3	Calcular $A+B$ y el resultado almacenarlo en C. ($C \leftarrow A + B$)
Paso 4	Calcular $A-B$ y el resultado almacenarlo en D. ($D \leftarrow A - B$)
Paso 5	Calcular $A*B$ y el resultado almacenarlo en E. ($E \leftarrow A * B$)
Paso 6	Imprimir el contenido de A, B, C, D, E
Paso 7	FIN

En los ejemplos anteriores se presentan dos algoritmos, uno cualitativo y el otro cuantitativo, pero ambos son directos, puesto que consisten de sólo una entrada, un proceso y una salida, así:

EJERCICIO ETAPA	INGRESO A LA UNIVERSIDAD	CONOCIDAS DOS CANTIDADES
ENTRADA	Reunir requisitos	Conocer los números A y B
PROCESO	Comprar y diligenciar formulario. Presentar examen.	Calcular suma, resta y producto de A y B
SALIDA	Ingresar o no a la Universidad.	Imprimir A, B, C, D, E.

No todas las soluciones son tan elementales como las presentadas hasta el momento, por eso se hace necesario avanzar un poco más y conocer las diferentes estructuras algorítmicas.

2.3 ESTRUCTURAS ALGORITMICAS

Las estructuras se pueden clasificar de la siguiente forma:

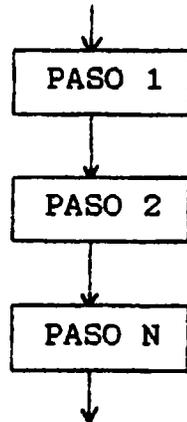
ESTRUCTURAS ALGORITMICAS	- SECUENCIAL
	- CONDICIONAL
	- REPETITIVAS O CICLICAS
	- REPETITIVAS ANIDADAS

2.3.1 ESTRUCTURA SECUENCIAL:

Es aquella en la que un paso o acción es ejecutado antes de continuar con el inmediatamente siguiente. Se caracteriza

por que el flujo del algoritmo recorre todo el proceso. desde el punto de entrada hasta su culminación, sin incluir ciclos o repeticiones.

Gráficamente se aprecia de la siguiente forma:



EJERCICIO

Se desea calcular el volumen de un paquete y para el efecto se dispone de las siguientes dimensiones: longitud, anchura y altura. Imprimir las dimensiones del paquete así como el volumen calculado.

Solución

a) Análisis

Entrada: Longitud, anchura y altura. Estas variables requieren ser leídas.

Proceso: $\text{Volumen} = \text{longitud} \times \text{anchura} \times \text{altura}$

Salida: Longitud, anchura, altura y volumen.

b) Variables a utilizar.

LON = longitud

AN = anchura

ALT = altura

VOL = volúmen

c) Algoritmo.

Paso 1	INICIO
Paso 2	Leer LON, AN, ALT
Paso 3	VOL ← LON * AN * ALT
Paso 4	Imprimir LON, AN, ALT, VOL
Paso 5	FIN

EJERCICIO

Un ahorrador desea acumular \$150.000= dentro de 3½ años. Por lo tanto coloca su dinero en una corporación de ahorro que le paga el 33% anual de interés compuesto. Cuánto debe depositar hoy para acumular tal cantidad al final del tiempo proyectado ?

Solución

a) Análisis

Entrada: Suma futura, Interés y Tiempo. Estas son variables que requieren ser leídas.

Proceso: Aplicar la siguiente fórmula:

$$\text{Suma Presente} = \text{Suma Futura} \cdot \left[\frac{1}{1 + \text{Interés}} \right]^{\text{Tiempo}}$$

Salida: Suma futura, Interés, Tiempo, Suma presente.

b) Variables a utilizar

S = Suma futura

P = Suma de dinero actual ó suma presente

I = Interés

N = Tiempo ó número de periodos.

c) Algoritmo

Paso 1	INICIO
Paso 2	Leer S, I, N
Paso 3	$P \leftarrow S * (1/(1 + I)^N)$
Paso 4	S, I, N, P
Paso 5	FIN

EJERCICIO

Leer los valores A y B. Calcular Z según la siguiente ecuación.

$$Z = \frac{-Y}{\sqrt[5]{A^2 - B^2}} - \frac{1}{Y \sqrt[3]{A^2 - B^2}} \quad \text{donde } Y = A - 2B^2$$

Solución

a) Análisis

Entrada: A, B

Proceso: Calcular $Y = A - 2B^2$, para luego buscar Z.

También se puede resumir el ejercicio calculando

$$D = A^2 - B^2$$

Salida: Z

b) Variables a utilizar

A, B = Valores a ser tratados

Y = Valor intermedio

Z = Valor resultante

c) Algoritmo

Paso 1	INICIO
Paso 2	Leer A, B
Paso 3	$Y \leftarrow A - 2 * B^2$
Paso 4	$D \leftarrow A^2 - B^2$
Paso 5	$Z \leftarrow -Y / D^{(1/5)} - 1/(Y * D^{(1/3)})$
Paso 6	Imprimir Z
Paso 7	FIN

En este ejercicio no se hizo necesario leer la variable Y, porque ella surge de efectuar el cálculo $A - 2B^2$

2.3.2 ESTRUCTURAS CONDICIONALES

Es frecuente que en la mayoría de los algoritmos se tengan que tomar decisiones, lo que origina una bifurcación del flujo dependiendo de la condición que se esté analizando en ese momento. Si la condición es verdadera toma una ruta, en

caso contrario toma la ruta alterna. A este tipo de estructuras también se les conocen como Estructuras de Decisión y se pueden clasificar así:

ESTRUCTURAS CONDICIONALES	[IF_THEN
		IF_THEN_ELSE
		CASE

2.3.2.1 Estructura IF - THEN (SI - ENTONCES).

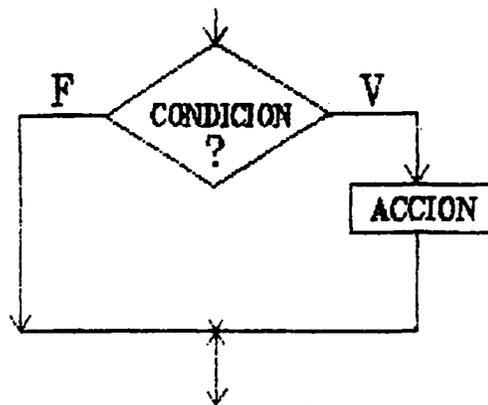
Se emplea cuando se desea ejecutar una ó más instrucciones dependiendo de si cumple una determinada condición del problema, en caso contrario, no se ejecuta instrucción alguna y se continúa con el flujo del algoritmo.

La estructura general es:

en Inglés	en Español
IF { Condición }	SI { Condición }
THEN ACCION	ENTONCES ACCION
ENDIF	FINSI

Como se observa, va acompañada de un indicador de final del rango del IF o del SI, para señalar cual es la acción ó acciones que se deben ejecutar cuando la condición sea verdadera.

En los diagramas de flujo comúnmente se usa un símbolo en forma de rombo ó diamante para indicar la decisión. Gráficamente se puede observar la estructura IF_THEN (SI - ENTONCES), de la siguiente manera:



La estructura tiene una sola entrada y el flujo se vuelve a unir a partir de la condición analizada, para continuar con la salida unificada.

EJERCICIO

Leer dos valores en un registro y calcular la suma de los números leídos; sólo si la suma es negativa, imprimir el resultado.

Solución

a) análisis

Entrada: Leer A, B

Proceso: Calcular $SUMA = A+B$

Averiguar si $SUMA$ es menor que cero para imprimir su contenido, en caso contrario no se imprime.

Salida: Imprimir $SUMA$ dependiendo de su valor.

b) Variables a utilizar

A, B = Valores a ser tratados

SUMA = Valor resultante

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer A, B
Paso 3          SUMA ← A + B
Paso 4          SI SUMA < 0
                  ENTONCES Imprimir SUMA
                  FINSI
Paso 5          FIN
  
```

De quererse obtener la misma solución un poco más ajustada a la sintaxis de los lenguajes de programación, se debe reemplazar los verbos INICIAR por START, LEER por READ, IMPRIMIR por WRITE y FINALIZAR por EXIT, con la siguiente presentación:

```

Paso 1          START
Paso 2          READ (A, B)
Paso 3          SUMA ← A + B
Paso 4          IF SUMA < 0
                  THEN WRITE (SUMA)
                  ENDIF
Paso 5          EXIT
  
```

Se ha incluido una sangría en la proposición IF para que

sea más fácil de leer y comprender.

EJERCICIO

Leer en un registro los valores A y B. Calcular la suma, resta, multiplicación y división de los números leídos. Imprimir los valores leídos, así como todo lo calculado.

Nota: Si $B = 0$, no se debe efectuar la división A/B .

Solución

a) Análisis

Entrada: Leer A, B

Proceso: Calcular

$$C = A + B$$

$$D = A - B$$

$$E = A * B$$

$$\text{Si } B \neq 0, \text{ calcular } F = A / B$$

Salida: A, B, C, D, E, F

b) Variables a utilizar

A, B = Variables a ser tratadas

C, D, E, F = Variables resultantes

c) Algoritmo

Paso 1	INICIO
Paso 2	Leer A, B
Paso 3	$C \leftarrow A + B$

```

Paso 4      D ← A - B
Paso 5      E ← A * B
Paso 6      Imprimir A, B, C, D, E
Paso 7      SI B <> 0
              ENTONCES F ← A / B
              IMPRIMIR F
              FINSI
Paso 8      FIN

```

Siempre que se formule un algoritmo, es fundamental probarlo con diversos datos para determinar su comportamiento. De aplicarse ésta técnica en forma permanente, se reducirá la probabilidad de errores y aumentará la confianza en las soluciones producidas.

2.3.2.1.1. Prueba de Escritorio

Consiste en reemplazar las variables que intervienen en el algoritmo con valores de diferente índole, efectuándoles seguimiento y observando su comportamiento. En esta forma se determina si el algoritmo funciona correctamente ó si por el contrario presenta errores de lógica. Esta prueba se debe hacer en repetidas ocasiones y con diversidad de datos, porque se puede presentar el caso de soluciones algorítmicas que se comportan eficientemente con ciertos datos, pero no con otros.

Las variables que intervienen en el proceso se colocan

horizontalmente y debajo de ellas se ubican los correspondientes valores que van tomando de acuerdo al flujo del algoritmo.

Ejemplo de prueba escritorio para el ejercicio anterior:

Si $A = 10$

$B = 5$

Se tiene:

A	B	C	D	E
10	5	15	5	50

Como B es diferente de 0, se efectúa la división y se tiene $F = 2$, para luego culminar el ejercicio.

Probando el mismo ejercicio con los siguientes datos:

Si $A = 10$

$B = 0$

Se tiene:

A	B	C	D	E
10	0	10	10	0

Como B es igual a 0, no se efectúa la división y termina el ejercicio.

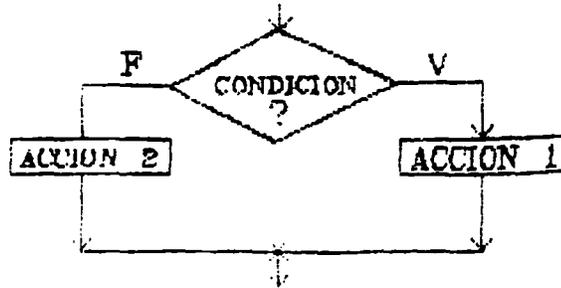
2.3.2.2 Estructura IF-THEN-ELSE (SI-ENTONCES- DE LO CONTRARIO ó SI-ENTONCES-SINO).

En caso de que la condición analizada sea verdadera se efectúa una ó más instrucciones a seguir. en caso contrario se realiza otra acción o conjunto de acciones diferentes,

unificándose en los flujos de tal manera que al final de la estructura se enrutan hacia el mismo punto. La estructura general es:

<u>en Inglés</u>	<u>en Español</u>
IF { Condición }	SI { Condición }
THEN PASO 1	ENTONCES PASO 1
ELSE PASO 2	SINO PASO 2
ENDIF	FINSI

Gráficamente se tiene:



EJERCICIO

Leer un valor diferente de cero y determinar si se trata de un número positivo o negativo.

Solución

a) Análisis

Entrada: Leer N

Proceso: Determinar si N es $>$ ó $<$ que cero

Salida: Producir el correspondiente mensaje.

b) Variables a utilizar

N: Variable a ser leída y analizada

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer N
Paso 3          SI N > 0
                  ENTONCES IMPRIMIR N "ES POSITIVO"
                  SINO IMPRIMIR N "ES NEGATIVO"
                  FINSI
Paso 4          FIN
  
```

d) Prueba de escritorio.

Si $N = 7$ IMPRIMIR 7 ES POSITIVO y termina el ejercicio.

Si $N = -3$ IMPRIMIR -3 ES NEGATIVO y termina el ejercicio.

EJERCICIO

Leer en un registro los valores X, Y, Z. Calcular W, R, T, así:

Si $X \leq Y$	_____	$W = X^2 + Y^2$
$X > Y$	_____	$W = X + Z^2 Y$
$Y = Z$	_____	$R = X^2 + Y^2 + Z^2$
$Y \langle \rangle Z$	_____	$R = X * Y / Z$
$W \geq R$	_____	$T = W * R$
$W < R$	_____	$T = W / R$

Solución

a) Análisis

Entrada: leer X, Y, Z

Proceso: Se debe Comparar X contra Y para calcular W.

Comparar Y contra Z para calcular R.

Comparar W contra R para calcular T.

Salida: Imprimir X, Y, Z, W, R, T

b) Variables a utilizar

X, Y, Z = Variables a ser leídas y analizadas

W, R, T = Variables resultantes.

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer X, Y, Z
Paso 3          SI X <= Y
                . ENTONCES W ← X * 2 + Y * 2
                SINO      W ← X + Z * 2 + Y
                FINSI
Paso 4          SI Y = Z
                ENTONCES R ← X * 2 + Y * 2 + Z * 2
                SINO      R ← X + Y * 2
                FINSI
Paso 5          SI W >= R
                ENTONCES T ← W * 2

```

SINO T ← W / R

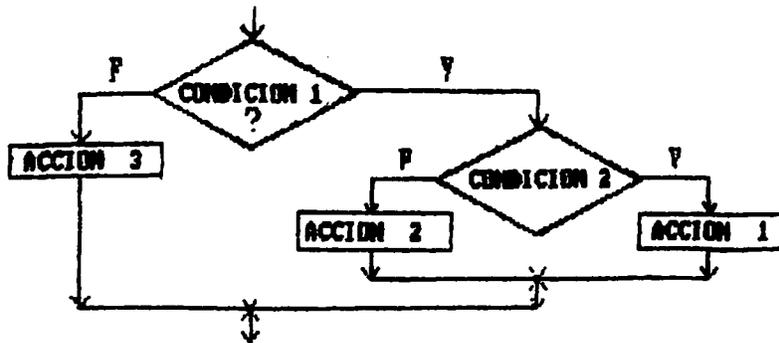
FINSI

Paso 6 IMPRIMIR X, Y, Z, W, R, T

Paso 7 FIN

2.3.2.2.1 En problemas de cualquier tamaño, las decisiones son necesarias y numerosas conformandose con frecuencia estructuras que incluyen a su vez otras condiciones, a éste tipo de estructuras se les conoce como **IF ANIDADAS** ó **CASCADA DE DECISIONES**.

Gráficamente su estructura es así:



Si la condición 1 es verdadera, entonces se analiza la condición 2, en caso de ser verdadera se ejecuta la acción 1, de lo contrario se ejecuta la acción 2; de no cumplirse la condición 1, se ejecuta la acción 3.

EJERCICIO

Leer código, nombre y salario bruto de un trabajador. Calcular el salario neto según la siguiente tabla de reten-

ción en la fuente:

SALARIO BRUTO	RETENCION EN LA FUENTE
0 - 41.000	0
41.001 - 60.000	1% del salario bruto
60.001 - 80.000	2% del salario bruto
80.001 - 100.000	4% del salario bruto
100.001 Y MAS	7% del salario bruto

Solución

a) Análisis

Entrada: Código del trabajador, Nombre del trabajador y salario bruto.

Proceso: Determinar a qué intervalo salarial corresponde el salario del trabajador para poder hallar la retención en la fuente, esto se logra mediante comparaciones sucesivas. Una vez calculada la retención en la fuente se procede al cálculo del salario neto.

Salida: Imprimir la información leída, así como la calculada.

b) Variables a utilizar

CODIGO: Código del trabajador

NOMBRE: Nombre del trabajador

SBRUTO@: Salario bruto

RET: Retención en la fuente

SNETO: Salario neto.

c) Algoritmo

Paso 1 INICIO

Paso 2 Leer CODIGO, NOMBRE, SBRUTO

Paso 3 SI SBRUTO \leq 41.000

ENTONCES RET \leftarrow 0

SINO SI SBRUTO \leq 60.000

ENTONCES RET \leftarrow SBRUTO * 0.01

SINO SI SBRUTO \leq 80.000

ENTONCES RET \leftarrow SBRUTO *0.02

SINO SI SBRUTO \leq 100.000

ENTONCES RET \leftarrow SBRUTO *0.04

SINO RET \leftarrow SBRUTO *0.07

FINSI

FINSI

FINSI

FINSI

Paso 4 SNETO \leftarrow SBRUTO - RET

Paso 5 IMPRIMIR CODIGO, NOMBRE, SBRUTO, RET, SNETO

Paso 6 FIN

d) Prueba de escritorio

CODIGO	NOMBRE	SBRUTO	RET	SNETO
32745	JUAN PEREZ	70000	$70000 \times 0.02 = 1400$	$70000 - 1400 = 68600$

Otro ejemplo:

CODIGO	NOMBRE	SBRUTO	RET	SNETO
47681	MARIA REYES	110000	$110000 \times 0.07 = 7700$	$110000 - 7700 = 102300$

EJERCICIO

La información de un contribuyente consta de su identificación y valor del patrimonio gravable (VPG). Se requiere calcular el impuesto a pagar sobre el patrimonio (IPP) de la siguiente forma:

$IPP = 0$ si $VPG \leq \$1000000$

$IPP = (VPG - \$1000000) \times 0.01$ si $\$1000000 < VPG \leq \2000000

$IPP = \$50000 + (VPG - \$2000000) \times 0.03$ si $\$2000000 < VPG \leq \3000000

$IPP = \$100000 + (VPG - \$3000000) \times 0.05$ si $VPG > \$3000000$

Solución

a) Análisis

Entrada: Identificación del contribuyente, valor del patrimonio gravable.

Proceso: Determinar en que rango se encuentra el valor del patrimonio gravable, para así calcular el impuesto a pagar sobre el patrimonio.

Salida: Identificación del contribuyente, valor del patrimonio gravable, impuesto a pagar sobre el patrimonio.

b) Variables a utilizar

IC = Identificación del contribuyente

VPG = Valor del patrimonio gravable

IPP = Impuesto a pagar sobre el patrimonio

c). Algoritmo

Paso 1 INICIO

Paso 2 Leer IC, VPG

Paso 3 SI VPG \leq 1000000ENTONCES IPP \leftarrow 0SINO SI (VPG > 1000000) y (VPG \leq 2000000)ENTONCES IPP \leftarrow (VPG - 1000000) * 0.01SINO SI (VPG > 2000000) y (VPG \leq 3000000)ENTONCES IPP \leftarrow 50000 + (VPG - 2000000) * 0.03SINO IPP \leftarrow 100000 + (VPG - 3000000) * 0.05

FINSI

FINSI

FINSI

Paso 4 IMPRIMIR IC, VPG, IPP

Paso 5 FIN

2.3.2.2.2 PROBLEMAS PROPUESTOS

1- Leer en un registro los valores A, B, y C. Calcular:

$$Z = (A+B)/C \quad \underline{\hspace{2cm}} \quad \text{si } C > 0$$

$$Z = (A - B)^C \quad \underline{\hspace{2cm}} \quad \text{si } C < 0$$

$$Z = 0 \quad \underline{\hspace{2cm}} \quad \text{si } C = 0$$

Imprimir A, B, C y Z

2- La calificación de una asignatura está compuesta por la suma de 3 exámenes parciales que tienen la siguiente ponderación: 25%, 35% y 40%. Calcular la nota definitiva. Imprimir las notas parciales y la nota definitiva.

3- Leer código, nombre y salario bruto de un trabajador. Calcular su retención en la fuente como el 5% de su salario bruto si es superior a \$80.000=, en caso contrario su retención se calcula como el 2% de su salario bruto. Calcular el salario neto e imprimir tanto la información leída como la calculada.

4- Leer en un registro el valor de A y determinar:

$$Z = (A + 7)^2 \quad \underline{\hspace{2cm}} \quad \text{si } 1 \leq A \leq 3$$

$$Z = \sqrt{3A - 5} \quad \underline{\hspace{2cm}} \quad \text{si } 4 \leq A \leq 7$$

$$Z = 7A / (A - 7)^3 \quad \underline{\hspace{2cm}} \quad \text{si } 8 \leq A \leq 10$$

Imprimir tanto el valor de A como su correspondiente valor Z.

2.3.2.3 Estructura de Decisión múltiple CASE (EN CASO DE-HACER).

Permite simplificar una serie de IF anidados y es eficiente cuando la decisión gira alrededor de datos discretos, cuan-

tificables o enumerables; en cualquier otra situación no tiene efectos positivos.

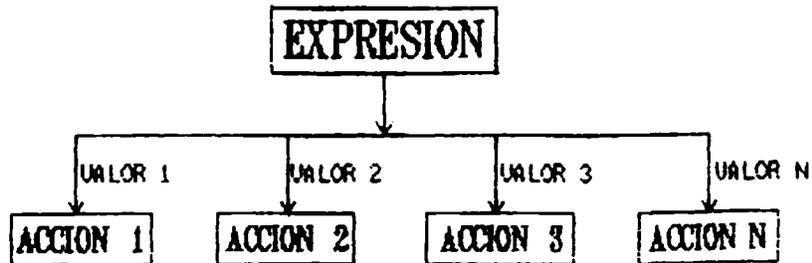
Esta estructura evalúa una expresión, la cual puede tomar diferentes valores y dependiendo del asumido en un momento dado, ejecuta la correspondiente acción de las múltiples posibles que se encuentran agrupadas.

Es de aclarar que la estructura de decisión múltiple, no es indispensable, ya que puede ser sustituida por la estructura IF, sin embargo algunos lenguajes de programación la incluyen y resulta ser de gran ayuda por su simplicidad y legibilidad al elaborar y estudiar las soluciones algorítmicas.

La estructura general es:

<u>en Inglés</u>	<u>en Español</u>
CASE EXPRESION OF	EN CASO DE EXPRESION HACER
V1: ACCION 1	VALOR 1: ACCION 1
V2: ACCION 2	VALOR 2: ACCION 2
. .	. .
. .	. .
. .	. .
VN: ACCION N	VALOR N: ACCION N
END	FIN_CASO

Gráficamente se tiene:



EJERCICIO

Calcular el salario neto devengado por un trabajador de una compañía según la siguiente información:

CODIGO, NOMBRE, CATEGORIA, NUMERO DE HORAS EXTRAS LABORADAS

Si:

CATEGORIA	SALARIO BASE	VALOR HORAS EXTRAS	SEGURO SOCIAL	OTROS DESCUENTOS
1	40.000	400	1% sal. base	0
2	60.000	600	1.5% sal. base	1% sal. base
3	80.000	800	2% sal. base	2% sal. base
4	100.000	1.000	2.5% sal. base	3% sal. base
5	120.000	1.200	3% sal. base	3.5% sal. base
6	140.000	1.400	3.5% sal. base	4% sal. base
7	200.000	2.000	4% sal. base	5% sal. base

Solución

a) Análisis

Entrada: Código del trabajador, nombre del trabajador, categoría, número de horas extras.

Proceso: Con base en la categoría del trabajador se determina el resto de la información necesaria para efectuar el cálculo del salario neto. Las fórmulas a emplear son:

$\text{Valor horas extras} = \text{No. horas extras} \times \text{valor respectivo.}$

$\text{Seguro social} = \text{salario base} \times \text{porcentaje respectivo.}$

$\text{Descuento} = \text{salario base} \times \text{porcentaje respectivo}$

$\text{Salario bruto} = \text{salario base} + \text{valor horas extras}$

$\text{Total descuentos} = \text{seguro social} + \text{descuentos}$

$\text{Salario Neto} = \text{salario bruto} - \text{total descuentos.}$

Salida: Imprimir código del trabajador, Nombre del trabajador, Salario bruto, Total descuentos, Salario Neto.

b) Variables a utilizar

CODIGO: Código del trabajador

NOMBRE: Nombre del trabajador

CATEGORIA: Categoría

NHE: Número de horas extras

SAL: Salario base

VHE: Valor horas extras

SS: Seguro Social

DCTOS: Descuentos

SBRUTO: Salario bruto

TDCITOS: Total descuentos

SNETO: Salario Neto.

c) Algoritmo

Paso 1

INICIO

Paso 2

Leer CODIGO, NOMBRE, CATEGORIA. NHE

Paso 3

EN CASO DE CATEGORIA HACER

1: SAL \leftarrow 40000

VHE \leftarrow NHE * 400

SS \leftarrow SAL * 0.01

DCTOS \leftarrow 0

2: SAL \leftarrow 60000

VHE \leftarrow NHE * 600

SS \leftarrow SAL * 0.015

DCTOS \leftarrow SAL * 0.01

3: SAL \leftarrow 80000

VHE \leftarrow NHE * 800

SS \leftarrow SAL * 0.02

DCTOS \leftarrow SAL * 0.02

4: SAL \leftarrow 100000

VHE \leftarrow NHE * 1000

SS \leftarrow SAL * 0.025

DCTOS \leftarrow SAL * 0.03

5: SAL \leftarrow 120000

VHE \leftarrow NHE * 1200

SS ← SAL * 0.03 .

DCTOS ← SAL * 0.035

6: SAL ← 140000

VHE ← NHE * 1400

SS ← SAL * 0.035

DCTOS ← SAL * 0.04

7: SAL ← 200000

VHE ← NHE * 2000

SS ← SAL * 0.04

DCTOS ← SAL * 0.05

FIN_CASO

Paso 4

SBRUTO ← SAL + VHE

Paso 5

TDCTOS ← SS + DCTOS

Paso 6

SNETO ← SBRUTO - TDCTOS

Paso 7

IMPRIMIR CODIGO, NOMBRE, SBRUTO, TDCTOS,
SNETO.

Paso 8

FIN

2.3.2.3.1 PROBLEMAS PROPUESTOS

1- Considérese el siguiente segmento de algoritmo

A ← 7

B ← 13

SI A > B

ENTONCES A ← A + 2

SINO B ← B + 1

FINSI

Determinar el valor de cada una de las variables después de la ejecución.

2- Considérese el siguiente segmento de algoritmo

A ← 7

B ← 13

SI A > B

 ENTONCES C ← 17

 SINO D ← 17

FINSI

Después de ejecutado lo anterior, cuál de las siguientes afirmaciones es cierta ?

a) C = 17 y D = 17

b) C = 17 y D queda indefinida

c) D = 17 y C queda indefinida

d) C y D quedan indefinidas

3- Considérese el siguiente segmento de algoritmo

X ← 7

Y ← 9

Z ← 17

SI X > Y

 ENTONCES SI Y > Z

 ENTONCES IMPRIMIR X

 SINO IMPRIMIR Y

 FINSI

 SINO IMPRIMIR Z

FINSI

Qué resultados se imprimen cuando se ejecuta el algoritmo ?

4- Considérese el siguiente segmento del programa

X ← -1

Y ← -3

SI X >= Y

ENTONCES IMPRIMIR X

SINO SI Y <= 0

ENTONCES IMPRIMIR Y

SINO IMPRIMIR Z

FINSI

FINSI

Qué resultados se imprimen cuando se ejecuta el algoritmo ?

5- Determinar cuál es el valor final al ser ejecutadas las siguientes instrucciones ?

EN CASO DE NOTA = 3 HACER

 1: IMPRIMIR "nota insuficiente"

 2: IMPRIMIR "nota reprobada"

 3: IMPRIMIR "nota aprobatoria"

 4: IMPRIMIR "nota buena"

 5: IMPRIMIR "nota excelente"

FIN_CASO

6- Escribir una estructura **EN_CASO_DE_HACER** equivalente a la siguiente porción algorítmica.

SI CALIF < 2

ENTONCES IMPRIMIR "el trabajo es deficiente"

SINO SI CALIF > 3

ENTONCES IMPRIMIR "El trabajo es regular"

SINO SI CALIF = 4

ENTONCES IMPRIMIR "El trabajo es bueno"

SINO IMPRIMIR "El trabajo es excelente"

FINSI

FINSI

FINSI

2.3.3 ESTRUCTURAS REPETITIVAS O CICLICAS

Los algoritmos estudiados hasta el momento presentan una estructura simple, compuesta sencillamente de una entrada, seguida de algunos procesos y concluyendo con una salida, pero no todos los problemas tienen soluciones de este tipo, algunos de ellos deben repetir una o más actividades (instrucciones) hasta que ciertas condiciones sean satisfechas, formando lo que se conoce como ciclos o procesos repetitivos.

Las estructuras repetitivas o cíclicas se pueden clasificar así:

Estructuras Repetitivas o Cíclicas

- * WHILE - DO
- * REPEAT - UNTIL
- * FOR

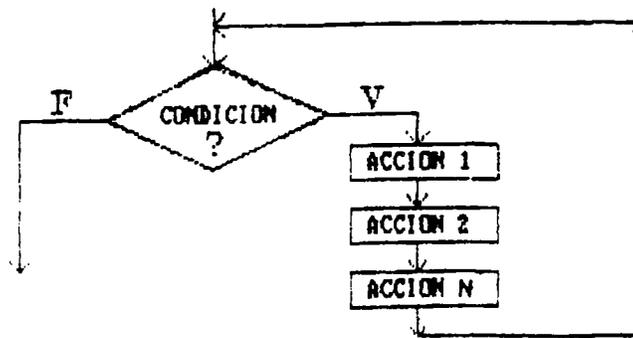
2.3.3.1 Estructura WHILE-DO (Mientras-Hacer).

Simboliza un conjunto de actividades que se desarrollan mientras una condición sea verdadera. Cuando la condición se torna falsa, se interrumpe dicha ejecución y continúa la secuencia normal en el algoritmo o programa.

La estructura general es:

<u>en Inglés</u>	<u>en Español</u>
WHILE {Condición} DO	MIENTRAS {Condición} HACER
ACCION 1	ACCION 1
ACCION 2	ACCION 2
ACCION N	ACCION N
END	FIN_MIENTRAS

Su representación gráfica es:



De acuerdo a la condición analizada es posible que la acción ó conjunto de acciones jamás se ejecuten.

Dentro del rango de acciones debe haber por lo menos un

parámetro que permita que la condición cambie su valor lógico, es decir, que en un momento dado pase de cierto a falso y continúe el flujo del algoritmo, de no ser así se caería en un ciclo infinito.

EJERCICIO

Elaborar un algoritmo para que calcule e imprima una tabla de multiplicar.

Solución

a) Análisis

Entrada: Leer el multiplicando o valor de la tabla que se desea calcular.

Proceso: Se inicializa la variable que actúa como multiplicador en 1 y se va incrementando de unidad en unidad en la medida en que se realizan los ciclos hasta llegar al límite, en este caso 10. Por cada ciclo se efectúa la multiplicación del multiplicando por el multiplicador.

Salida: Imprimir por cada ciclo, el multiplicando, el multiplicador y el resultado.

b) Variables a utilizar

MDO = Multiplicando

MDOR = Multiplicador

RES = Resultado

c) Algoritmo

Paso 1 **INICIO**
Paso 2 **Leer MDO**
Paso 3 **MDOR ← 1**
Paso 4 **MIENTRAS MDOR <= 10 HACER**
 RES ← MDO * MDOR
 IMPRIMIR MDO, MDOR, RES
 MDOR ← MDOR + 1
 FIN_MIENTRAS
Paso 5 **FIN**

d) Prueba de escritorio

Si la variable leída MDO representa un valor de 5, se tiene:

MDO		MDOR		RES
5	x	1	=	5
	x	2	=	10
	x	3	=	15
		.		.
		.		.
		.		.
		10		50

EJERCICIO

Sea la ecuación $Y = X^3 - 3X^2 - X + 5$

Calcular Y para cada valor de X que oscila entre -5 y 5 y

que sufre incrementos de 0.5 cada vez. Imprimir X y su correspondiente valor Y.

Solución

a) Análisis:

Entrada: No requiere lectura de variables, pero se debe inicializar la variable X en -5.

Proceso: Tan pronto se calcula el valor Y correspondiente a X, se debe proceder a incrementar ésta última en 0.5 por cada ciclo, sin que llegue a superar el valor de 5. Para cada valor de X se evalúa la ecuación.

Salida: Imprimir las variables X, Y

b) Variables a utilizar

X = Valor a ser considerado en la ecuación

Y = Valor resultante de la ecuación

c) Algoritmo:

```

Paso 1          INICIO
Paso 2          X ← -5
Paso 3          MIENTRAS X <= 5 HACER
                Y ← X ↑ 3 - 3 * X ↑ 2 - X + 5
                IMPRIMIR X, Y
                X ← X + 0.5
                FIN_MIENTRAS
Paso 4          FIN

```

2.3.3.1.1 Acumulador

Es una variable que tiene como función acumular o totalizar los cálculos intermedios que se realizan en un proceso. Esta variable se debe inicializar en cero para que no altere los cálculos definitivos.

EJERCICIO

Calcular la suma de los 100 primeros números naturales.

Solución

a) Análisis:

Entrada: No requiere lectura de variables, pero se debe inicializar una variable en el primer número natural ó sea en 1, así mismo se debe inicializar un acumulador en cero.

Proceso: Incrementar la variable que haga las veces de número natural en una unidad por cada ciclo, mientras sea inferior a 100, e igualmente aumentar la variable acumuladora en el nuevo número natural.

Salida: Imprimir la variable acumuladora, pero sólo al culminar el proceso de sumar los 100 primeros números naturales.

b) Variables a utilizar

I = Número natural

SUMA = Acumulador

c) Algoritmo

```

Paso 1          INICIO
Paso 2          SUMA ← 0
Paso 3          I ← 1
Paso 4          MIENTRAS I ≤ 100 HACER
                SUMA ← SUMA + I
                I ← I + 1
                FIN_MIENTRAS
Paso 5          IMPRIMIR SUMA
Paso 6          FIN

```

Las instrucciones o actividades que se encuentran dentro del ciclo **MIENTRAS-HACER** se ejecutan tantas veces como la condición lo permita.

Analizar el siguiente algoritmo

```

Paso 1          INICIO
Paso 2          SUMA ← 0
Paso 3          I ← 1
Paso 4          MIENTRAS I < 100 HACER
                SUMA ← SUMA + I
                I ← I + 1

```

IMPRIMIR SUMA

Paso 5 **FIN MIENTRAS**

Paso 6 **FIN**

Pregunta

- Cuántos números naturales son procesados ?
- Qué valores son impresos ?

EJERCICIO

Imprimir la suma de los números pares comprendidos entre 1 y 100.

Solución**a) Análisis**

Entrada: Se debe inicializar el acumulador de números pares en cero. No obstante ser el primer número par 2, se debe inicializar una variable que haga sus veces en 1, por restricciones del enunciado del problema.

Proceso: Se debe aplicar un artificio de tal forma que genere todos los pares comprendidos entre 1 y 100, incluyendo el primero que es 2, e igualmente incrementar la variable que haga las veces de número par en 2 mientras sea inferior a 100.

Salida: Imprimir la variable acumuladora, pero solo al culminar el proceso de sumar todos los números

paros.

b) Variables a utilizar

J = Número par

ACUM = Acumulador

c) Algoritmo

```

Paso 1          INICIO
Paso 2          ACUM ← 0
Paso 3          J ← 1
Paso 4          MIENTRAS J < 100 HACER
                  ACUM ← ACUM + (J - 1) + 2
                  J ← J + 2
                  FIN_MIENTRAS
Paso 5          IMPRIMIR ACUM
Paso 6          FIN
  
```

d) Prueba de escritorio

ACUM	J
0	1
$0+(1-1)+2 = 0+2 = 2$	3
$2+(3-1)+2 = 2+4 = 6$	5
$6+(5-1)+2 = 6+6 = 12$	7
$12+(7-1)+2 = 12+8 = 20$	9
$20+(9-1)+2 = 20+10 = 30$	

Pregunta

Es frecuente oír en los estudiantes de lógica computacional, la siguiente pregunta:

Al abordar un problema, cuándo se debe leer y cuándo no ?

Respuesta

Se debe leer cuando los datos a ser tratados se desconocen, por lo tanto éstos son reemplazados por variables. Cuando los datos son conocidos se inicializan directamente las variables con sus respectivos valores o datos.

EJERCICIO

$$\text{Calcular SUM} = \frac{1}{1} + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{55}$$

Solución**a) Análisis**

Entrada: Cuando se trata de una sumatoria, se debe inicializar una variable acumuladora en cero.

Observando el denominador de cada término, tiene un comportamiento impar, por lo tanto, se debe inicializar una variable en 1 para que maneje el denominador. Con el numerador no hay problema por

ser constante.

Proceso: Generar cada término de la serie mediante incrementos de dos unidades que debe sufrir el denominador, siempre que sea inferior al límite preestablecido que en este caso es 55, e irlo acumulando.

Salida: Imprimir el total acumulado.

b) Variables a utilizar

K = Denominador del término

SUM = Sumatoria de términos

c) Algoritmo

Paso 1	INICIO
Paso 2	SUM ← 0
Paso 3	K ← 1
Paso 4	MIENTRAS K ≤ 55 HACER
	SUM ← SUM + 1 / K
	K ← K + 2
	FIN_MIENTRAS
Paso 5	IMPRIMIR SUM
Paso 6	FIN

2.3.3.1.2 EJERCICIOS PROPUESTOS

- 1- Imprimir los 100 primeros números naturales.
- 2- Producir una tabla de enteros y cuadrados para todos los impares comprendidos entre el 3 y 75 inclusive.
- 3- Producir una tabla de raíces cuadradas para todos los números impares comprendidos entre 0 y 50.
- 4- Calcular $R = \frac{1}{5} + \frac{1}{10} + \frac{1}{15} + \frac{1}{20} + \dots + \frac{1}{100}$
- 5- Considérese la ecuación $Y = X^3 - 16X^2 + 13X - 7$. Diseñar un algoritmo que calcule Y para los valores de X que oscilan entre 4 y -4 y que tiene decrementos de 0.2. Imprimir para cada X su correspondiente valor Y.

2.3.3.1.3 Lo usual en los procesos computacionales es elaborar soluciones que consideren ciertos volúmenes de información, pero se presenta el inconveniente que el computador no sabría determinar cuando debe culminar de leer un lote de registros ó cuando finalizar la repetición de un cálculo o proceso, salvo que se le indique la conclusión del proceso repetitivo y eso se logra mediante el empleo de una variable contadora de ciclos. En la medida en que se incrementen o decrementen los ciclos, así mismo se comporta la variable contadora, hasta llegar a un tope o límite previamente establecido. A este proceso se le conoce con el

previamente establecido. A este proceso se le conoce con el nombre de **CONTADOR**.

Observando los enunciados de los ejercicios desarrollados hasta el momento, presentan la siguiente formulación: "Se tiene un registro con la siguiente información ...", ó "leer un registro con la siguiente información...", ó "leer en un registro..."; lo cierto es que siempre se trataba de un registro, ahora bien, una de las grandes ayudas que brindan los computadores es el manejo eficiente de grandes volúmenes de información en tiempos mínimos, resultando utópico pensar en resolver un problema en forma computacional para unos datos individuales o únicos, por lo tanto, lo visto hasta el momento tiene un carácter netamente académico - docente, más no práctico.

De acuerdo a lo anterior y con el soporte del conocimiento de los conceptos de **ACUMULADOR** y **CONTADOR**, se realizarán en adelante ejercicios con un mayor grado de eficiencia y eficacia.

EJERCICIO

Leer 10 valores enteros positivos, uno por registro y obtener el cuadrado, el cubo, raíz cuadrada y raíz cúbica. Imprimir cada valor leído, así como lo hallado.

Solución

a) Análisis

Entrada: Leer un dato por registro.

Inicializar una variable contadora de ciclos en uno.

Proceso: Mientras el contador de ciclos sea menor que 10 se debe leer un dato y calcular el cuadrado, el cubo, la raíz cuadrada y raíz cúbica del dato leído.

Incrementar el contador de ciclos en 1

Salida: Por cada dato leído imprimir lo calculado

b) Variables a utilizar:

CC = Contador de ciclos

K = dato a ser leído

CUAD = cuadrado

CUBO = cubo

RCUA = raíz cuadrada

RCUB = raíz cúbica

c) Algoritmo

Paso 1

INICIO

Paso 2

CC ← 1

Paso 3

MIENTRAS CC ≤ 10 **HACER**

Leer K

CUAD ← $K \uparrow 2$

CUB ← $K \uparrow 3$

RCUA ← $K \uparrow (1/2)$

RCUB ← $K \uparrow (1/3)$

IMPRIMIR K, CUAD, CUB, RCUA, RCUB

CC ← CC + 1

FIN_MIENTRAS

Paso 4 FIN

La variable contadora de ciclos puede estar inicializada indistintamente en cero o en uno, pero se debe tener especial cuidado al controlar el límite con el fin no incurrir en problemas como realizar menos procesos de los deseados, o por el contrario, sobrepasar los límites establecidos.

Lo anterior se puede observar con un sencillo ejemplo de una nómina de 100 empleados. Por lógica el cálculo del pago para cada uno de los empleados se realiza en forma individual y es así como se calcula la liquidación respectiva y se edita su correspondiente pago o cheque: si por error en el control del límite, éste sólo realiza 99 ciclos, dejaría de calcular el pago para un trabajador, o si por el contrario realiza 101 ciclos, el flujo del algoritmo se quedaría en espera de ser alimentado con la información correspondiente al trabajador 101, el cual no existe.

EJERCICIO

Presentación del ejercicio inmediatamente anterior trabajado con el contador de ciclos inicializado en CERO.

Algoritmo

Paso 1 **INICIO**
Paso 2 $CC \leftarrow 0$
Paso 3 **MIENTRAS** $CC < 10$ **HACER**
 Leer K
 $CUAD \leftarrow K \uparrow 2$
 $CUB \leftarrow K \uparrow 3$
 $RCUA \leftarrow K \uparrow (1/2)$
 $RCUB \leftarrow K \uparrow (1/3)$
 IMPRIMIR K, CUAD, CUB, RCUA, RCUB
 $CC \leftarrow CC + 1$
 FIN_MIENTRAS
Paso 4 **FIN**

EJERCICIO

Se conoce la edad de cada uno de los 50 estudiantes de un grupo. Determinar cuántos estudiantes son mayores de 30 años y cuántos tienen 30 años o menos.

Solución**a) Análisis**

Entrada: Se requiere de un contador de procesos (ciclos) inicializado en cero.

Un contador de edades mayores de 30 años inicializado en cero

Un contador de edades menores o iguales a 30 años

inicializado en cero.

Proceso: Mientras el contador de ciclos sea inferior a 50, debe leer una edad y compararla contra 30 para determinar si se trata de una edad mayor de 30 años. de ser afirmativo, se incrementa en 1 el contador de mayores de 30 años. en caso contrario, se incrementa en 1 el contador de menores o iguales a 30 años. Una vez hecho esto, se debe incrementar el contador de procesos en 1.

Salida: Imprimir el contador de mayores de 30 años y el contador de menores o iguales a 30 años.

b) Variables a utilizar

CP = Contador de procesos

CPME = Contador de personas menores o iguales a 30 años

CPMA = Contador de personas mayores de 30 años

EDAD = Edad de una persona.

c) Algoritmo

Paso 1	INICIO
Paso 2	CP ← 0
Paso 3	CPME ← 0
Paso 4	CPMA ← 0
Paso 5	MIENTRAS CP < 50 HACER

Leer EDAD

SI EDAD \leq 30

ENTONCES CPME \leftarrow CPME + 1

SINO CPMA \leftarrow CPMA + 1

FINSI

CP \leftarrow CP + 1

FIN_MIENTRAS

Paso 6 **IMPRIMIR** CPMA, CPME

Paso 7 **FIN**

EJERCICIO PROPUESTO

Analizar el siguiente algoritmo mediante una prueba de escritorio y determinar lo que calcula, así como el último valor que se imprime en la variable IMP.

Variables a utilizar

IMP = Número impar

Algoritmo

Paso 1 **INICIO**

Paso 2 IMP \leftarrow 1

Paso 3 **MIENTRAS** IMP < 50 **HACER**

IMPRIMIR IMP

IMP \leftarrow IMP + 2

FIN_MIENTRAS

Paso 4 **FIN**

Pregunta

Cuál es la diferencia entre un CONTADOR y un ACUMULADOR ?

Respuesta

La variable que hace las veces de CONTADOR, sufre modificaciones por cada ciclo en valores constantes, en cambio, la variable ACUMULADORA se ve afectada en cantidades variables por cada iteración.

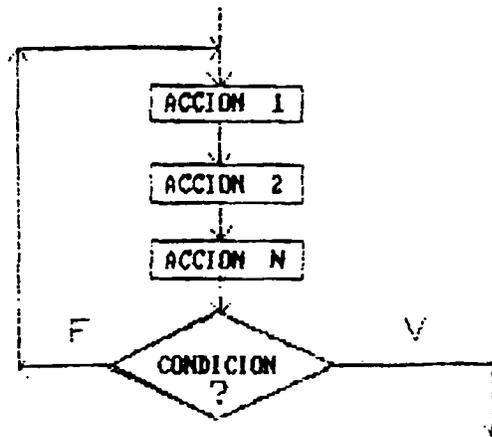
2.3.3.2 Estructura REPEAT-UNTIL (REPETIR_HASTA_QUE).

Esta estructura se ejecuta en forma cíclica en tanto el valor de la condición sea falso, es allí donde cesa la repetición y continúa la marcha normal del algoritmo. Esta estructura es exactamente contraria a la estructura WHILE-DO.

La estructura general es:

<u>en Inglés</u>	<u>en Español</u>
REPEAT	REPETIR
ACCION 1	ACCION 1
ACCION 2	ACCION 2
ACCION N	ACCION N
UNTIL {CONDICION}	HASTA_QUE {CONDICION}

Su representación gráfica es:



ESTA ESTRUCTURA SE EJECUTA AL MENOS UNA VEZ.

Observaciones:

- Debe haber por lo menos un parámetro que permita que la condición pase de falso a verdadero.
- Todo lo que es capaz de hacer el **REPEAT** lo puede hacer el **WHILE** y con mayor eficiencia, ya que por el hecho de examinar antes de ejecutar evita situaciones anómalas. Es en éstos casos cuando se requiere del **WHILE-DO**, siendo esta estructura más amplia o general.
- En la primera pasada o ciclo del **REPEAT** se pueden establecer las condiciones iniciales, en el **WHILE-DO** se deben establecer las condiciones iniciales antes de entrar en él. por ejemplo tener dos bloques de lectura, uno para arrancar y el otro para permanecer en el cuerpo

del WHILE-DO.

- La utilización de la estructura REPEAT es recomendable cuando se conoce previamente el número de repeticiones a efectuar; así mismo es muy empleada para validación de datos.

EJERCICIO

Leer 20 registros, cada uno conteniendo una EDAD.

Determinar:

- * número de personas que tienen 20 años o menos.
- * número de personas que son mayores de 20 años y menores o iguales a 40 años.
- * número de personas que son mayores de 40 años y menores o iguales a 60 años.
- * número de personas mayores de 60 años.

Solución

a) Análisis

Entrada: Se requiere de cinco contadores que se inicializan todos en cero y son:

- * contador de registros
- * contador de menores o iguales a 20 años
- * contador de mayores de 20 años y menores o iguales a 40 años.
- * contador de mayores de 40 años y menores o iguales a 60 años.
- * contador de mayores de 60 años.

Se debe leer una edad por registro.

Proceso: Determinar en qué intervalo se encuentra la Edad leída para incrementar el correspondiente contador, así como el contador de registros sin que sobrepase de 20.

Salida: Imprimir los diferentes contadores de edades.

b) Variables a utilizar

NR = Número de registros

NP20 = Número de personas menores o iguales a 20 años

NP40 = Número de personas mayores de 20 años y menores o iguales a 40 años.

NP60 = Número de personas mayores de 40 años y menores o iguales a 60 años.

NPMA = Número de personas mayores de 60 años

EDAD = Edad de la persona.

c) Algoritmo

Paso 1 INICIO

Paso 2 $NR \leftarrow 0$

Paso 3 $NP20 \leftarrow 0$

Paso 4 $NP40 \leftarrow 0$

Paso 5 $NP60 \leftarrow 0$

Paso 6 $NPMA \leftarrow 0$

Paso 7 REPETIR

Leer EDAD

```

SI EDAD <= 20
    ENTONCES NP20 ← NP20 + 1
    SINO SI EDAD <= 40
        ENTONCES NP40 ← NP40 + 1
        SINO SI EDAD <= 60
            ENTONCES NP60 ← NP60 + 1
            SINO NPMA ← NPMA + 1
        FINSI
    FINSI
FINSI
    FINSI
FINSI
NR ← NR + 1
HASTA QUE NR = 20

```

Paso 8 IMPRIMIR NP20, NP40, NP60, NPMA

Paso 9 FIN

EJERCICIO

Leer un número entero positivo y calcular su factorial.

Recuerde que $0! = 1$

$3! = 1 \times 2 \times 3 = 6$

$N! = 1 \times 2 \times \dots \times N$

Solución

a) Análisis

Entrada: Leer la variable a la cual se le calculará el factorial.

Inicializar una variable en 1, que se comportará como el multiplicador.

Inicializar una variable acumuladora del multiplicador en 1.

Proceso: Es necesario controlar que el valor leído no sea negativo, en caso de presentarse, se debe producir un mensaje de inconsistencia .

Se debe efectuar la multiplicación de la variable acumuladora por el multiplicador y el resultado almacenarlo en ella misma, e incrementar el multiplicador en 1 hasta que sea mayor que la variable leída, que en este caso hace las veces de límite.

Salida: Imprimir el valor leído, así como el factorial calculado.

b) Variables a utilizar

N = Valor al cual se le calculará el factorial

FACN = Factorial de N

J = Multiplicador

c) Algoritmo

Paso 1	INICIO
Paso 2	FACN ← 1
Paso 3	J ← 1
Paso 4	Leer N
Paso 5	SI N < 0
	ENTONCES IMPRIMIR " Error en Datos"
	SINO REPETIR
	FACN ← FACN * J

$$J \leftarrow J + 1$$

HASTA QUE $J > N$

IMPRIMIR N, FACN

FINSI

Paso 6

FIN

d) Prueba de escritorio

Si $N = 4$ se tiene:

N	FACN	J
4	1	1
	$1 \times 1 = 1$	$1 + 1 = 2$
	$1 \times 2 = 2$	$2 + 1 = 3$
	$2 \times 3 = 6$	$3 + 1 = 4$
	$6 \times 4 = 24$	$4 + 1 = 5$

Observación:

Cuando se requiere inicializar variables con propósito de sumatoria, éstas deben arrancar en cero para que no se presenten distorsiones en los resultados. Si los fines son de productoria, las variables deben comenzar en 1 para que el multiplicador no convierta la función en cero. como sucedería si fueran empezadas en cero.

EJERCICIO

Calcular
$$SUM = \frac{1!}{2} + \frac{2!}{4} + \frac{3!}{6} + \frac{4!}{8} + \dots + \frac{N!}{2N}$$

Solución**a) Análisis**

Entrada: Leer N ó límite de la serie

Inicializar el numerador en 1, sumatoria de términos en cero y multiplicador del factorial en 1.

Proceso: Tan pronto se calcula el numerador de cada término de la serie se halla el denominador como 2 veces el numerador. Conformado cada fraccionario, se va acumulando.

La variable N hace las veces de límite.

Salida: Imprimir el acumulado

b) Variables a utilizar

FACT = Multiplicador del factorial

SUM = Sumatoria de términos

N = Límite de términos

I = Numerador

DOBLE = Denominador

c) Algoritmo

Paso 1	INICIO
Paso 2	FACT ← 1
Paso 3	SUM ← 0
Paso 4	I ← 1

Paso 5 Leer N

Paso 6 REPETIR

FACT ← FACT * I

DOBLE ← 2 * I

SUM ← SUM + FACT / DOBLE

I ← I + 1

HASTA_QUE I > N

Paso 7 IMPRIMIR SUM

Paso 8 FIN

2.3.3.3 Estructura FOR (PARA).

Quando se conoce de antemano el número de veces que debe ejecutar un proceso, se puede realizar automáticamente mediante la utilización de una variable a la cual se le asignan condiciones iniciales, condiciones finales y los incrementos que sufre ésta. Cuando el incremento es negativo el ciclo no se ejecuta, si el valor inicial es menor que el límite, o también, cuando siendo el incremento positivo y el valor inicial es mayor que el valor límite.

La estructura general es:

<u>en Inglés</u>	<u>en Español</u>
FOR V: VI TO VF, IN DO	PARA V DE VI HASTA VF. IN HACER
ACCION 1	ACCION 1
ACCION 2	ACCION 2
·	·
·	·
ACCION N	ACCION N
END	FIN_PARA

Siendo:

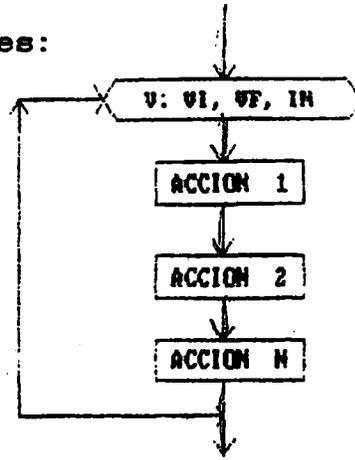
V = Variable controladora del ciclo

VI = Valor inicial

VF = Valor final

IN = Incrementos

Su representación gráfica es:



Cuando no se indica el valor del incremento, se asume la unidad.

Esta estructura es sumamente flexible y práctica, por lo tanto al utilizarse simplifica bastante la solución de los problemas, porque no requiere inicializar variables controladoras de ciclos, ni incrementarlas o compararlas contra límites preestablecidos ya que lo hace en forma automática.

EJERCICIO

Se desea calcular la cantidad a pagar de matrícula para cada uno de los 2000 estudiantes de una universidad, en la

siguiente forma:

Total a pagar es igual a costos fijos más costos variables. Los costos fijos son de \$1000 y los costos variables se calculan sumando el 12% del patrimonio y el 17% de la renta.

Por cada estudiante se tiene un registro con la siguiente información: código del estudiante, nombre del estudiante, renta y patrimonio.

Solución

a) Análisis

Entrada: Leer código del estudiante, nombre, renta y patrimonio. Hacer costos fijos igual a 1000.

Proceso: Aplicar las siguientes fórmulas:

Costo variable = Renta x 12% y Patrimonio x 17%

Costo total = Costo fijo + Costo variable.

Salida: Imprimir código del estudiante, nombre, renta, patrimonio, costos fijos, costos variables y costo total.

b) Variables a utilizar

COD = Código del estudiante

NOM = Nombre

RENTA = Renta

PAT = Patrimonio

CFIJO = Costo fijo

CVAR = Costo variable

TPAG = Total a pagar

I = Variable índice de la estructura FOR (PARA)

c) Algoritmo

Paso 1 INICIO

Paso 2 CFIJO ← 1000

Paso 3 PARA I DE 1 HASTA 2000 HACER

 Leer COD, NOM, RENTA, PAT

 CVAR ← RENTA * 0.12 + PAT * 0.17

 TPAG ← CFIJO + CVAR

 IMPRIMIR COD, NOM, RENTA, PAT, CFIJO,

 CVAR, TPAG.

 FIN_PARA

Paso 4 FIN

EJERCICIO

Leer por registro, el código de un empleado, el salario básico por hora y el número de horas trabajadas en la semana. Calcular el salario neto para una nómina de 100 empleados teniendo en cuenta que si el número de horas trabajadas durante la semana es mayor de 48, las horas demás, se consideran horas extras y tienen un recargo del 35%. Imprimir código y salario neto para cada empleado.

Solución**a) Análisis**

Entrada: Se aplica una estructura **PARA (FOR)**, en donde su valor inicial es 1, su valor final es el total de empleados, ó sea 100 y formando parte de la estructura, se tiene el cálculo del salario bruto igual al salario básico hora por el número de horas trabajadas.

Comparar número de horas trabajadas contra 48, si es mayor, se calcula el recargo correspondiente a horas extras como salario básico hora x (número de horas trabajadas - 48) * 0.35.

Calcular el salario neto = salario bruto + recargo de horas extras.

Salida: Código del empleado, salario neto.

b) Variables a utilizar

COD = Código del empleado

SBH = Salario básico hora

NHT = Número de horas trabajadas en la semana

SB = Salario bruto

RHE = Recargo por horas extras

SN = Salario neto

J = Variable índice de la estructura **FOR (PARA)**

c) Algoritmo

Paso 1 **INICIO**

Paso 2 **PARA J DE 1 HASTA 100 HACER**

Leer COD, SBH, NHT

SB ← SBH * NHT

RHE ← 0

SI NHT > 48

ENTONCES RHE ← SBH * (NHT-48) * 0.35

FINSI

SN ← SB + RHE

IMPRIMIR COD, SN

FIN_PARA

Paso 3 **FIN**

En el presente ejercicio se inicializó la variable RHE (recargo por horas extras) en cero, para que no afecte el salario neto por indefinición cuando al comparar NHT (número de horas trabajadas) contra 48 resulte ser menor o igual.

EJERCICIO

Calcular la combinatoria de N en M, según la siguiente fórmula:

$$\left[\begin{matrix} N \\ M \end{matrix} \right] = \frac{N!}{M! (N - M)!}$$

Restricción: N debe ser mayor que M y ambos enteros positivos.

Solución**a) Análisis**

Entrada: Leer N, M

Inicializar variables multiplicatorias en 1 para ser empleadas en el cálculo de los factoriales.

Proceso: Chequear que N sea mayor que M

Calcular la fórmula por partes, primero el factorial de N, luego el factorial de M y por último el factorial de la diferencia de N con M.

Salida: Imprimir N, M y la combinatoria

b) Variables a utilizar

N, M, K = Valores a ser factorizados

FACN = Factorial de N

FACM = Factorial de M

FACK = Factorial de K

COMBI = Combinatoria

I, J, L ≅ Variables índices de los FOR (PARA)

c) Algoritmo

Paso 1 **INICIO**

Paso 2 **FACN ← 1**

Paso 3 **FACM ← 1**

Paso 4 **LEER N, M**

Paso 5 **SI N < M**

ENTONCES IMPRIMIR "ERROR EN DATOS"

SINO PARA I DE 1 HASTA N HACER

88

FACN ← FACN * I

FIN_PARA

PARA J DE 1 HASTA M HACER

FACM ← FACM * J

FIN_PARA

K ← N - M

FACK ← 1

PARA L DE 1 HASTA K HACER

FACK ← FACK * L

FIN_PARA

COMBI ← FACN / (FACM * FACK)

IMPRIMIR N, M, COMBI

FINSI

Paso 6, FIN

2.3.3.3.1 PROBLEMAS PROPUESTOS

- 1- Qué valores se imprimen al ejecutarse la siguiente fracción de algoritmo.

SUMA ← 0

MIENTRAS J <= 18 HACER

SUMA ← SUMA + J

I ← I + 2

FIN_MIENTRAS

IMPRIMIR J, SUMA

- 2- Dados los siguientes datos de entrada 5, 10, 15, 0.

20, determinar que valores se imprimen según siguiente algoritmo.

SUMA \leftarrow 0

Leer X

MIENTRAS X > 0 **HACER**

SUMA \leftarrow SUMA + X

Leer X

FIN_MIENTRAS

IMPRIMIR SUMA

3- Qué valores se imprimen al ejecutarse el siguiente algoritmo.

I \leftarrow 1

SUMA \leftarrow 0

MIENTRAS I < 5 **HACER**

CUA \leftarrow I \uparrow 2

SUMA \leftarrow SUMA + CUA

I \leftarrow I + 2

FIN_MIENTRAS

IMPRIMIR SUMA

4- Cuántos ciclos realiza la siguiente estructura **REPETIR**.

A \leftarrow 7

B \leftarrow 5

REPETIR

A \leftarrow A + 1

HASTA_QUE A > B

5- Cuál es el valor final de la VARIABLE SUMA ?

```
SUMA ← 0
PARA I DE 1 HASTA 5, 2 HACER
  L ← 2 * I
  PARA J DE 1 HASTA L HACER
    SUMA ← SUMA + J
  FIN_PARA
FIN_PARA
IMPRIMIR SUMA
```

6- Cuál es el valor final que se imprime en la variable Y ?

```
PARA X DE 1 HASTA 3 HACER
  SI X <= 1
    ENTONCES Y ← X - 1
  SINO SI X <= 2
    ENTONCES Y ← A + 1
    SINO Y ← A - 1
  FINSI
FINSI
IMPRIMIR Y
```

7- Cuántos ciclos realiza la estructura REPETIR ? Qué valor se imprime en la variable SUM ?

```
SUM ← 0
A ← 1
B ← 2
```

REPETIR

C ← A * B

SUM ← SUM + C

HASTA_QUE A > 5

IMPRIMIR SUM

2.3.4 ESTRUCTURAS REPETITIVAS ANIDADAS

Dentro del cuerpo de una estructura de repetición se pueden incluir otras estructuras de repetición siempre y cuando no se presente sobreposición ó cruce de ellas, es decir, la estructura interna debe estar totalmente incluida dentro de la estructura externa.

Se debe tener presente al ejecutarse una estructura repetitiva anidada, que por cada valor que toma la variable índice de la estructura externa se ejecuta totalmente la estructura interna. Ejemplo:

PARA J DE 1 HASTA 3 HACER

PARA L DE 1 HASTA 4 HACER

CUAD ← L ↑ 2

IMPRIMIR CUAD

FIN_PARA

FIN_PARA

Prueba de escritorio

J	L	CUAD
1	1	$1^2 = 1$
	2	$2^2 = 4$
	3	$3^2 = 9$
	4	$4^2 = 16$

2	1	$1^2 = 1$
	2	$2^2 = 4$
	3	$3^2 = 9$
	4	$4^2 = 16$

3	1	$1^2 = 1$
	2	$2^2 = 4$
	3	$3^2 = 9$
	4	$4^2 = 16$

Observando la prueba de escritorio se tiene que cuando J vale 1 se ejecuta totalmente la estructura interna L, que está variando de 1 a 4, luego J se incrementa a 2 y se activa de nuevo la estructura interna L y así sucesivamente hasta culminar la estructura externa J.

Ejemplo:

REPETIR

Leer J

PARA L DE 1 HASTA 4 HACER

CUAD ← L ↑ 2

IMPRIMIR CUAD

FIN_PARA

HASTA QUE $J \leq 0$

Prueba de Escritorio

Si $J = 3$	L	CUAD
	1	$1^2 = 1$
	2	$2^2 = 4$
	3	$3^2 = 9$
	4	$4^2 = 16$

Se sale de la estructura interna y lee un nuevo valor de J.

Así:

Si $J = 7$	L	CUAD
	1	$1^2 = 1$
	2	$2^2 = 4$
	3	$3^2 = 9$
	4	$4^2 = 16$

El proceso termina cuando lee un valor $J \leq 0$

EJERCICIO

Para determinar el pago mensual, un almacén de electrodomésticos tiene en cuenta el monto del crédito y el plazo, aplicando la siguiente fórmula:

$$\text{PAGO MENSUAL} = \frac{\text{DEUDA TOTAL}}{\text{PLAZO}} + \text{INTERESES}$$

donde interés = 3% del saldo.

Por cada uno de los 70 clientes, se dispone de la siguiente información: código del cliente, nombre, deuda total y plazo. Determinar para cada cliente, los intereses pagados, el pago mensual y el saldo, para cada mes según el plazo concedido.

Solución

a) Análisis:

Entrada: Leer código, nombre, deuda total y plazo.

Inicializar los acumuladores de total de intereses y total cuota en cero.

Proceso: Se utilizarán dos estructuras PARA anidadas. Una para leer la información de cada uno de los clientes y la otra para calcular e imprimir la correspondiente tabla de pagos mensuales para cada cliente.

Salida: Imprimir código, nombre, deuda total, plazo, pago mensual, saldo, total de intereses.

b) Variables a utilizar

NOM = Nombre

COD = Código

DTOTAL = Deuda total

TINT = Total intereses

PLAZO = Plazo concedido

TCUOTA = Total cuota

CUOTA = Cuota mensual

SALDO = Saldo

INT = Interés

PAGOME = Pago mensual

I, J = Variables Índice de los FOR (PARA)

c) Algoritmo

```

Paso 1          INICIO
Paso 2          PARA I DE 1 HASTA 70 HACER
                  Leer COD, NOM, DTOTAL, PLAZO
                  IMPRIMIR COD, NOM, DTOTAL, PLAZO
                  TINT ← 0
                  TCUOTA ← 0
                  CUOTA ← DTOTAL / PLAZO
                  PARA J DE 1 HASTA PLAZO HACER
                      TCUOTA ← TCUOTA + CUOTA
                      SALDO ← DTOTAL - TCUOTA
                      INT ← SALDO * 0.03
                      PAGOME ← CUOTA + INT
                      TINT ← TINT + INT
                      IMPRIMIR PAGOME, SALDO, TINT
                  FIN_PARA
                  FIN_PARA
Paso 3          FIN

```

EJERCICIO

Calcular el valor de Y según la siguiente serie:

$$Y = \frac{X^0}{0!} + \frac{X^2}{3!} - \frac{X^5}{7!} - \frac{X^9}{12!} + \frac{X^{14}}{18!} + \frac{X^{20}}{25!} \dots \text{HASTA 20 TERMINOS}$$

Para cada valor de X que oscile entre 1 y 10, imprimir el valor de X y su correspondiente valor de Y.

Solución

a) Análisis

Entrada: Inicializar el primer término de la serie en 1.

el exponente en 2,

el denominador en 3,

la diferencia entre exponentes en 3.

la diferencia entre denominadores en 4.

Proceso: Se genera la serie a partir del segundo término puesto que el primero es igual a uno. Se debe observar el comportamiento de los exponentes de la serie así como de los denominadores, porque la diferencia entre ellos es cada vez más creciente. Para encontrar la nueva potencia, se debe proceder a hallar la diferencia entre potencias de los dos términos anteriores e incrementarla en una unidad, para luego aumentarle ese resultado a la nueva potencia. El mismo procedimiento se realiza para hallar el próximo denominador.

Salida: Imprimir X y su correspondiente valor Y.

b) Variables a utilizar

EXP = Exponente

DEN = Denominador

DEXP = Diferencia entre exponentes

DDEN = Diferencia entre denominadores

FACDEN = Factorial del denominador.

c) Algoritmo

Paso 1

INICIO

Paso 2

PARA X DE 1 HASTA 10 HACER

Y ← 1

EXP ← 2

DEN ← 3

DEXP ← 3

DDEN ← 4

PARA N DE 2 HASTA 20 HACER

FACDEN ← 1

PARA I DE 1 HASTA DEN HACER

FACDEN ← FACDEN * I

FIN_PARA

Y ← Y + ((-1)^X * X)^{EXP} / FACDEN

EXP ← EXP + DEXP

DEN ← DEN + DDEN

DEXP ← DEXP + 1

DDEN ← DDEN + 1

FIN_PARA

IMPRIMIR X, Y

FIN_PARA

Paso 3

FIN

Observación

El límite máximo o tope de un conjunto de datos puede ser reemplazado por una variable. En este caso se compara el contador de registros o contador de ciclos contra la variable límite para determinar el final de un proceso ó ciclo repetitivo; de emplearse la estructura FOR (PARA), no es necesario hacer la comparación anterior porque la estructura lo controla automáticamente. La variable límite debe ser definida previamente, siendo lo más lógico, mediante lectura.

EJERCICIO

Una editorial estima los precios de sus libros de acuerdo al siguiente criterio: El precio básico de un libro es de \$100 más 0.70 centavos por página. Si el número de páginas es superior a 250, el precio del libro se incrementará en \$300, pero si el número de páginas excede de 500, el precio del libro se aumentará en otros \$1000,00.

Se dispone de un volumen de M libros y por cada registro se

conoce el código del libro y el número de páginas. Calcular e imprimir el valor de cada libro identificándolo con su código.

Solución

a) Análisis:

Entrada: Leer el límite de datos.

Leer por cada registro, el código del libro y el número de páginas.

Proceso: Calcular el precio básico del libro como \$100,00 + 0.70 por número de páginas.

Dependiendo del número de páginas, se tiene un recargo, así:

Si el número de páginas es mayor de 250 y menor o igual a 500 el recargo será de \$300,00, pero si el número de páginas excede de 500 sufre un incremento de \$1300.

Salida: Imprimir código y precio de cada libro.

b) Variables a utilizar

M = Límite de datos a analizar

J = Índice de la estructura PARA

CODIGO = Código del libro

NPAG = Número de páginas

PBAS = Precio básico del libro

PLIB = Precio del libro

c) Algoritmo

```

Paso 1      INICIO
Paso 2      Leer M
Paso 3      PARA J DE 1 HASTA M HACER
              Leer CODIGO, NPAG
              PBAS ← 100 + 0.7 * NPAG
              SI NPAG ≤ 250
                  ENTONCES PLIB ← PBAS
              SINO SI NPAG ≤ 500
                  ENTONCES PLIB ← PBAS + 300
              SINO      PLIB ← PBAS + 1300
              FINSI
              FINSI
              IMPRIMIR CODIGO, NPAG, PLIB
              FIN_PARA
Paso 4      FIN

```

EJERCICIO

Leer un conjunto N de valores X y encontrar su correspondiente valor Y, según las siguientes condiciones:

$$Y = X^2 - 7 \text{ si } \dots\dots\dots 3 < X < 7$$

$$Y = \frac{(X - 1)!}{(2X)!} \text{ si } \dots\dots\dots 7 \leq X < 15$$

$Y = \sqrt{X}$ si X es diferente a las condiciones anteriores.

Solución

a) Análisis:

Entrada: Leer el límite del conjunto de datos N .

Leer el valor de X , uno por registro.

Cómo el ejercicio debe calcular el factorial en 2 ocasiones, debe inicializar las variables multiplicadoras en 1.

Proceso: Se utiliza una estructura FOR (PARA) externa cuyo límite máximo es N . Por cada ciclo debe leer un valor X y calcular su correspondiente valor Y de acuerdo a las especificaciones del enunciado.

Salida: Imprimir X , Y

b) Variables a utilizar

N = Límite del conjunto de valores

X = Valor a ser analizado, debe ser entero

$FACM$ = Factorial del numerador

$FAC2X$ = Factorial del denominador

L, K = Valores índices de las estructuras PARA

c) Algoritmo

Paso 1 INICIO

Paso 2 Leer N

Paso 3 PARA I DE 1 HASTA N HACER

Leer X

SI $(X > 3)$ Y $(X < 7)$

ENTONCES $Y \leftarrow X \uparrow 2 - 7$

SINO SI $(X \geq 7)$ Y $(X < 15)$

ENTONCES $M \leftarrow X - 1$

FACM $\leftarrow 1$

PARA L DE 1 HASTA M HACER

FACM \leftarrow FACM * L

FIN_PARA

J $\leftarrow 2 * X$

FAC2X $\leftarrow 1$

PARA K DE 1 HASTA J HACER

FAC2X \leftarrow FAC2X * K

FIN_PARA

Y \leftarrow FACM / FAC2X

SINO Y $\leftarrow X \uparrow 0.5$

FINSI

FINSI

IMPRIMIR X, Y

FIN_PARA

Paso 4 FIN

EJERCICIO

En una empresa manufacturera se calcula el jornal para cada uno de sus M obreros de acuerdo al salario básico hora por el número de horas trabajadas, más una bonificación de \$100 por cada unidad extra producida sobre un mínimo de producción diaria.

Cuando la producción es inferior a la producción mínima establecida, el jornal es equivalente al salario básico hora multiplicado por el número de horas trabajadas.

Por cada obrero se lee: código del obrero, nombre, salario básico hora, número de horas trabajadas, producción diaria, y producción mínima por día.

Se debe producir un informe que contenga la misma información leída, además del salario neto por día para cada obrero. Adicionalmente la empresa requiere saber el total de unidades extras producidas, así como el total pagado por nómina cada día.

Solución**a) Análisis**

Entrada: Se requiere de dos acumuladores, uno para el número total de unidades extras producidas y el otro para el total de nómina pagada, ambos inicializados en cero.

Se debe leer el límite de datos a procesar, así como el código del obrero, nombre, salario básico hora, número de horas trabajadas, producción diaria y producción mínima diaria por cada uno de los obreros.

Proceso: Emplear las siguientes fórmulas:

-Salario básico diario = salario básico hora x número de horas trabajadas.

Número de unidades extras producidas = producción diaria - producción mínima diaria.

Salario neto diario = salario básico diario + número de unidades extras producidas x 100

Número total de unidades extras producidas =
Número total de unidades extras producidas +
número de unidades extras producidas.

Total nómina pagada = total nómina pagada +
salario neto diario.

Salida: Imprimir código del obrero, nombre, salario básico hora, producción mínima diaria, producción diaria, salario neto diario, número total de unidades extras producidas y total nómina pagada.

b) Variables a utilizar

COD = Código del obrero

NOM = Nombre

SBH = Salario básico hora

NHT = Número horas trabajadas

PMD = Producción mínima diaria

PD = Producción diaria

NUEP = Número de unidades extras producidas

NTUEP = Número total de unidades extras producidas

SND = Salario neto diario

TNP = Total nómina pagada

SBD = Salario básico diario

M = Límite de datos a ser procesados

K = Variable índice de la estructura PARA (FOR)

c) Algoritmo.

```

Paso 1          INICIO
Paso 2          TNP ← 0
Paso 3          NTUEP ← 0
Paso 4          Leer M
Paso 5          PARA K DE 1 HASTA M HACER
                  Leer COD, NOM, SBH, NHT, PMD, PD
                  SBD ← SBH * NTH
                  SI PD > PMD
                    ENTONCES NUEP ← PD - PMD
                    NTUEP ← NTUEP + NUEP
                    SND ← SBD + NUEP * 100
                  SINO SND ← SBD

```

FINSI

IMPRIMIR COD, NOM, SBH, PMD, PD, SND

TNP ← TNP + SND

FIN_PARA

Paso 6 IMPRIMIR NTUEP, TNP

Paso 7 FIN

2.4 A menudo se encuentra que es imposible determinar el número de registros que se deben procesar y para poder definirsele al computador, se utiliza otro registro no procesable que se coloca al final del volumen de información (ARCHIVO) que se desea procesar. A este registro se le dá el nombre de Registro **DUMMY** o Registro **CENTINELA**.

Generalmente se utiliza como marca de fin de archivo el /*, en otras ocasiones se emplea una comparación contra nueves (99999.) ó contra ceros (00000.), sin que se llegue a pensar que se están procesando 99999 registros o 00000 registros, porque se debe recordar, que ésta es una marca de fin de lote ó fin de archivo y no es procesable.

Por ejemplo, la mejor señal de que un cuaderno de notas se ha acabado, es cuando se han agotado todas sus hojas y se llega a la cubierta o pasta, siendo ésta un indicativo de finalización del cuaderno de notas.

EJERCICIO

Encontrar la calificación promedio de un gran número, aunque desconocido, de exámenes. La escala de calificaciones es de 0 a 5.

Solución**a) Análisis**

Entrada: Se necesita iniciar el contador de calificaciones y la sumatoria de calificaciones en cero.

Proceso: Para poder calcular el promedio, se hace necesario conocer la sumatoria de las calificaciones así como el número de calificaciones sumadas, en este caso desconocido. Para obviar éste inconveniente se utiliza un registro DUMMY ó centinela que controla el total de calificaciones a procesar, consistente en leer una calificación y compararla contra 5, mientras sea inferior o igual a 5 se procesa; cuando encuentra una nota superior a 5, deja de leer más datos y calcula el promedio. Como en este ejercicio la nota oscila entre 0 y 5, no sería normal encontrar una calificación superior a 5, por lo tanto, se emplea esta condición como centinela.

Salida: Imprimir el promedio

b) Variables a utilizar

SUMA = Sumatoria de calificaciones

CONT = Contador de calificaciones

CALIF = Calificación

PROM = Promedio

c) Algoritmo

```

Paso 1          INICIO
Paso 2          SUMA ← 0
Paso 3          CONT ← 0
Paso 4          Leer CALIF
Paso 5          MIENTRAS CALIF <= 5 HACER
                SUMA ← SUMA + CALIF
                CONT ← CONT + 1
                Leer CALIF
                FIN_MIENTRAS
Paso 6          PROM ← SUMA / CONT
Paso 7          IMPRIMIR PROM
Paso 8          FIN
  
```

d) Prueba de escritorio

Se desean procesar las siguientes notas: 3.0, 4.0, y 2.0.

Se utilizará como registro centinela una nota fuera del rango 0-5 que podría ser 6.0, o cualquiera otra fuera del rango permitido.

CALIF	SUMA	CONT	PROM
3.0	0	0	
4.0	0+3.0 = 3.0	0+1 = 1	9.0/3 = 3

2.0	3.0+4.0 = 7.0	1+1 = 2
6.0	7.0+2.0 = 9.0	2+1 = 3

La última nota leída no se procesa, puesto que ella sólo sirve de controladora de final de datos.

EJERCICIO

El radio de una circunferencia se calcula según la siguiente fórmula: $R = \sqrt{(X-CX)^2 + (Y-CY)^2}$, siendo el punto (X,Y) el centro de una circunferencia y (CX,CY) un punto sobre la circunferencia.

Calcular además, el área y la longitud de la circunferencia, así como la suma de las áreas y la suma de las longitudes de todos los registros procesados. El proceso termina cuando el radio sea igual a cero.

Fórmulas a emplear: $AREA = \pi R^2$

$$LONGITUD = 2\pi R$$

Solución

a) Análisis:

Entrada: Inicializar sumatoria de las áreas y sumatoria de las longitudes en cero.

Reemplazar la constante PI (π) por 3.14159

Leer los puntos X, Y, CX, CY.

Proceso: Se leen los puntos X, Y, CX, CY tantas veces mientras el radio sea igual a cero, sirviendo en este caso de registro centinela pues determina la cantidad de datos a procesar.

Salida: Imprimir los puntos X, Y, CX, CY, el área y longitud de la circunferencia, la sumatoria de las áreas y la sumatoria de las longitudes.

b) Variables a utilizar

X, Y = Centro de la circunferencia

CX, CY = Punto sobre la circunferencia

RADIO = Radio

AR = Area de la circunferencia

SUMAR = Sumatoria de las áreas

LON = Longitud de la circunferencia

SUMLON = Sumatoria de longitudes

PI = Número π = 3.14159

c) Algoritmo

Paso 1	INICIO
Paso 2	SUMAR \leftarrow 0
Paso 3	SUMLON \leftarrow 0
Paso 4	PI \leftarrow 3.14159
Paso 5	Leer X, Y, CX, CY
Paso 6	RADIO \leftarrow $((X-CX)^2 + (Y-CY)^2)^{0.5}$
Paso 7	MIENTRAS RADIO > 0 HACER
	AR \leftarrow PI * RADIO \uparrow 2
	LON \leftarrow 2 * PI * RADIO
	SUMAR \leftarrow SUMAR + AR
	SUMLON \leftarrow SUMLON + LON
	IMPRIMIR X, Y, CX, CY, AR, LON

Leer X, Y, CX, CY

RADIO ← $((X-CX)^2 + (Y-CY)^2)^{0.5}$

FIN_MIENTRAS

Paso 8 IMPRIMIR SUMAR, SUMLON

Paso 9 FIN

EJERCICIO

Encontrar el promedio de estaturas de un grupo de estudiantes universitarios. El proceso termina cuando se encuentre una estatura menor o igual a cero.

Solución

a) Análisis

Entrada: Leer una estatura por registro.

Inicializar sumatoria de estaturas y contador de estaturas en cero.

Proceso: Emplear la siguiente fórmula:

$$\text{Promedio} = \frac{\text{sumatoria de estaturas}}{\text{número de estaturas sumadas}}$$

Mientras la estatura sea mayor que cero se debe acumularla e incrementar el contador de estaturas sumadas en 1.

Salida: Imprimir promedio de estaturas.

b) Variables a utilizar

CE = Contador de estaturas

SE = Sumatoria de estaturas

EST = Estatura

PROM = Promedio de estaturas

c) Algoritmo

```

Paso 1          INICIO
Paso 2          CE ← 0
Paso 3          SE ← 0
Paso 4          Leer EST
Paso 5          MIENTRAS EST > 0 HACER
                CE ← CE + 1
                SE ← SE + EST
                Leer EST
                FIN_MIENTRAS
Paso 6          PROM ← SE / CE
Paso 7          IMPRIMIR PROM
Paso 8          FIN
  
```

2.5 PROBLEMAS RESUELTOS

EJERCICIO

La siguiente serie 0, 1, 1, 2, 3, 5, 8, 13,....., K, se conoce como la sucesión de FIBONACCI; elaborar un algoritmo para calcularla.

Solución

a) Análisis

Entrada: Sólo requiere la lectura del límite de la sucesión.

Se inicializa el primer término en cero y el segundo en 1.

Proceso: Analizando la serie se puede observar que el

tercer término lo genera la suma de los dos términos anteriores, así mismo, el cuarto término es generado por la suma de los dos términos que le anteceden y así sucesivamente.

Para no proliferar en la creación de variables por cada término calculado, se hace un intercambio de variables, es decir, inicialmente se suman las variables A y B y el contenido se almacena en la variable C, se imprime C y luego el contenido de B se traslada a la variable A, el contenido de C se traslada a la variable B y se efectúa nuevamente el cálculo de A+B, tantas veces como términos se deseen generar y trabajando siempre con las mismas 3 variables A, B, C.

Salida: Imprimir la serie

b) Variables a utilizar

A, B = Valores iniciales de la serie

C = Término generado

K = Límite de términos de la serie.

c) Algoritmo

Paso 1	INICIO
Paso 2	Leer K
Paso 3	$A \leftarrow 0$
Paso 4	$B \leftarrow 1$
Paso 5	Imprimir A, B

Salida: Imprimir los valores ordenados.

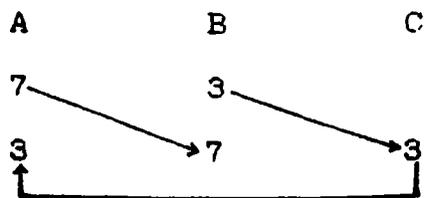
c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer A, B
Paso 3          Si A > B
                  ENTONCES C ← B
                  B ← A
                  A ← C
                  FINSI
Paso 4          IMPRIMIR A, B "ORDENADOS ASCENDENTEMENTE"
Paso 5          FIN
  
```

d) Prueba de escritorio

Si A = 7 y B = 3 entonces:



EJERCICIO

Leer en un registro los valores A, B, C y ordenarlos ascendentemente.

Solución

a) Análisis:

Entrada: Leer A, B, C

Proceso: Se utiliza el mismo procedimiento del ejercicio anterior.

Salida: Imprimir los valores ordenados

b) Variables a utilizar

A, B, C = Variables a ser ordenadas

D, E = Variables temporales

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer A, B, C
Paso 3          REPETIR
                SI A > B
                    ENTONCES D ← B
                        B ← A
                            A ← D
                FINSI
                SI B > C
                    ENTONCES E ← C
                        C ← B
                            B ← E
                FINSI
                HASTA_QUE (A <= B) Y (B <= C)
Paso 4          IMPRIMIR A, B, C "ORDENADOS ASCENDENTEMENTE"
Paso 5          FIN

```

EJERCICIO PROPUESTO

Probar el algoritmo anterior mediante una prueba de escritorio.

EJERCICIO

$$\text{Calcular } W = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \dots\dots - \frac{1}{81}$$

Solución**a) Análisis**

Entrada: No requiere lectura de datos porque cada término se autogenera.

Inicializar la sumatoria de términos en cero y el controlador de términos I en 1.

Proceso: Aplicar la siguiente fórmula:

$$W = W + \frac{(-1)^{(I-1)}}{(2(I-1) + 1)}$$

donde $(-1)^{(I+1)}$ genera el signo del término

$(2(I-1) + 1)$ genera el denominador.

Salida: Imprimir el acumulado W

b) Variables a utilizar

W = Sumatoria de términos

I = Límite de términos

c) Algoritmo

```

Paso 1      INICIO
Paso 2      W ← 0
Paso 3      I ← 1
Paso 4      REPETIR
              W ← W+(-1)I(I-1)/(2*(I-1)+1)
              I ← I + 1
              HASTA QUE I > 31
Paso 5      IMPRIMIR W
Paso 6      FIN

```

Otra forma de resolver el mismo ejercicio. Segunda versión

Algoritmo

```

Paso 1      INICIO
Paso 2      W ← 0
Paso 3      PARA I DE 1 HASTA 31 HACER
              W ← W+(-1)I(I-1)/(2*I-1)
              FIN_PARA
Paso 4      IMPRIMIR W
Paso 5      FIN

```

Otra forma de resolver el mismo ejercicio. Tercera versión

Algoritmo

```

Paso 1      INICIO

```

```

Paso 2      N ← -1
Paso 3      W ← 0
Paso 4      H ← -1 {controla el signo del término}
Paso 5      PARA I DE 1 HASTA 61,2 HACKR {I sufre in-
              crementos de 2 por cada ciclo}

              N ← N * H
              W ← W + N / I

              FIN PARA
Paso 6      IMPRIMIR W
Paso 7      FIN

```

EJERCICIO PROPUESTO

Realizar las correspondientes pruebas de escritorio.

EJERCICIO

$$\text{Calcular } T = \frac{1!}{2} + \frac{3!}{4} + \frac{5!}{6} + \frac{7!}{8} + \dots + \frac{47!}{48!}$$

Solución**a) Análisis**

Entrada: Inicializar el acumulador de términos en cero y la variable manejadora del factorial en 1.

Proceso: Se emplea un anidamiento de 2 estructuras FOR, siendo la interna para calcular el correspondiente factorial y la externa para calcular totalmente cada término y acumularlo.

Salida: Imprimir el acumulado de términos

b) Variables a utilizar

SUMA = Acumulador de términos

FAC = Multiplicador del factorial

T = Término de la serie

L = Numerador

I, J = Variables índice de la estructura PARA

c) Algoritmo

```

Paso 1          INICIO
Paso 2          SUMA ← 0
Paso 3          PARA I DE 1 HASTA 24 HACER
                L = 2 * I - 1
                FAC ← 1
                PARA J DE 1 HASTA L HACER
                    FAC ← FAC * J
                FIN_PARA
                T ← FAC / (2 * I)
                SUMA ← SUMA + T
                FIN_PARA
Paso 4          IMPRIMIR SUMA
Paso 5          FIN

```

Otra versión del ejercicio anterior es:

Solución**a) Análisis**

Entrada: No requiere lectura de datos, puesto que los términos se autogeneran.

Se debe inicializar: una variable para manejar el factorial en 1, una variable acumuladora de términos en 0 y una variable para manejar el denominador en 2.

Proceso: Se genera la serie a partir del segundo término.

Emplear las siguientes fórmulas:

$FAC = FAC * (I-1) * I$, siendo I la variable índice de la estructura PARA.

$DOBLE = DOBLE + 2$

$SUMA = SUMA + FAC / DOBLE$

$T = T + SUMA$

Salida: Imprimir el acumulado T

b) Variables a utilizar

FAC = Factorial

DOBLE = Denominador.

SUMA = Acumulador de términos

T = Primer término

c) Algoritmo

Paso 1 **INICIO**

Paso 2 DOBLE ← 2

Paso 3 $T \leftarrow 1 / DOBLE$
 Paso 4 $FAC \leftarrow 1$
 Paso 5 $SUMA \leftarrow 0$
 Paso 6 **PARA I DE 3 HASTA 47,2 HACER**
 $FAC \leftarrow FAC * (I - 1) * I$
 $DOBLE \leftarrow DOBLE + 2$
 $SUMA \leftarrow SUMA + FAC / DOBLE$
 FIN_PARA
 Paso 7 $T \leftarrow T + SUMA$
 Paso 8 **IMPRIMIR T**
 Paso 9 **FIN**

EJERCICIO

La compañía Colombiana S.A vende 6 productos diferentes y para la venta de éstos tiene 4 vendedores, en donde cada vendedor está encargado de la venta de la misma clase de artículo, pero un artículo puede ser vendido por varios vendedores.

Por cada venta se dispone de la siguiente información:

Código del vendedor, código del artículo, cantidad de unidades vendidas, precio de etiqueta del artículo y precio de venta real.

Calcular e imprimir lo siguiente:

Código del vendedor, código del artículo, precio de etiqueta, precio de venta real y comisión.

La comisión se calcula de acuerdo a la siguiente tabla:

CODIGO DEL ARTICULO**COMISION**

1	2% del precio total de la venta real
2	1% de la diferencia, del precio total de lo etiquetado y el precio total de la venta real.
3	5% del precio total de la venta real, más el 10% de la diferencia del precio total de lo etiquetado y el precio total de la venta real.
4	\$100,00 por unidad vendida.
5	\$50 por unidad, más 7% del precio total de la venta real.
6	\$30,00 por unidad, más 1% del precio de etiqueta del artículo.

Solución**a) Análisis**

Entrada: Leer el límite de vendedores.

Leer el código del vendedor, código del artículo, cantidad de unidades vendidas, precio de etiqueta del artículo y precio de venta real para cada venta.

Proceso: Calcular:

El precio total de la venta real = precio de venta real x cantidad de unidades vendidas.

El precio total de lo etiquetado = precio de

etiqueta por x cantidad de unidades vendidas.

Se utiliza una estructura **EN CASO DE ... HACER** y dependiendo del código del artículo, se calcula la correspondiente comisión.

Salida: Código del vendedor, código del artículo, precio de etiqueta, precio de venta real, precio total de lo etiquetado, precio total de venta real y comisión.

b) Variables a utilizar:

N = Límite de datos a ser procesados
 CODV = Código del vendedor
 CODA = Código del artículo
 CUV = Cantidad de unidades vendidas
 PEA = Precio de etiqueta del artículo
 PVR = Precio de venta real
 PTE = Precio total de lo etiquetado
 PTVR = Precio total de venta real
 COM = Comisión
 K = Índice de la estructura PARA

c) Algoritmo

Paso 1 **INICIO**
 Leer N
 PARA K DE 1 HASTA N HACER
 Leer CODV, CODA, CUV, PEA, PVR

$PTE \leftarrow PEA * CUV$

$PTVR \leftarrow PVR * CUV$

EN CASO DE CODA HACER

1: $COM \leftarrow PTVR * 0.02$

2: $COM \leftarrow (PTE - PTVR) * 0.01$

3: $COM \leftarrow (PTVR * 0.05) + (PTE - PTVR) * 0.1$

4: $COM \leftarrow CUV * 100$

5: $COM \leftarrow CUV * 50 + PTVR * 0.07$

6: $COM \leftarrow CUV * 30 + PEA * 0.01$

FIN_CASO

IMPRIMIR CODA, CODV, PEA, PVR, PTE, PTVR,

COM

FIN_PARA

Paso 4 FIN

EJERCICIO

Una compañía de seguros desea conocer el valor del cheque que debe elaborar cada mes por concepto de daños de vehículos. El taller de reparación envía a la compañía cada una de las N facturas, con la siguiente información:

Placa del vehículo, tipo del daño y costo.

El valor que reconoce la empresa de seguros puede ser de tres tipos:

Tipo 1: reconoce hasta 1'000.000

Tipo 2: reconoce hasta 500.000

Tipo 3: reconoce hasta 300.000

Si el valor del daño reportado por el taller de reparaciones es superior al valor aceptado por la compañía de seguros, ésta solamente reconocerá lo previamente establecido.

Calcular el total del cheque que la empresa debe pagar al final del mes, así como el total de cheque en caso de que no se tenga en cuenta las anteriores restricciones.

Solución

a) Análisis

Entrada: Leer el límite de facturas a procesar.

Leer por cada registro: placa del vehículo, tipo de daño y costo.

Inicializar en cero las variables acumuladoras del costo del cheque sin restricciones y del costo del cheque con restricciones.

Proceso: Dependiendo del tipo de daño y del costo se determina el valor reconocido por la compañía de seguros. para luego efectuar los correspondientes acumulados.

Salida: Imprimir el total del cheque con restricciones y el total del cheque sin restricciones.

b) Variables a utilizar

N = Límite de facturas a procesar

PLACA = Placa del vehículo

TD = Tipo de daño

CD = Costo del daño

VR = Valor reconocido

TCSR = Total cheque sin restricciones

TCCR = Total cheque con restricciones

c) Algoritmo

Paso 1 INICIO

Paso 2 TCSR \leftarrow 0

Paso 3 TCCR \leftarrow 0

Paso 4 LEER N

Paso 5 PARA J DE 1 HASTA N HACER

Leer PLACA, TD, CD

TCSR \leftarrow TCSR + CD

SI TD = 1

ENTONCES SI CD \geq 1'000.000

ENTONCES VR \leftarrow 1'000.000

SINO VR \leftarrow CD

FINSI

SINO SI TD = 2

ENTONCES SI CD \geq 500.000

ENTONCES VR \leftarrow 500.000

SINO VR \leftarrow CD

FINSI

SINO SI CD \geq 300.000

ENTONCES VR \leftarrow 300.000

SINO VR ← CD

FINSI

FINSI

FINSI

TCCR ← TCCR + VR

FIN_PARA

Paso 6 IMPRIMIR TCCR, TCSR

Paso 7 FIN

Otra forma de resolver el mismo ejercicio. Segunda versión

Algoritmo.

Paso 1 INICIO

Paso 2 TCSR ← 0

Paso 3 TCCR ← 0

Paso 4 LEER N

Paso 5 PARA J DE 1 HASTA N HACER

Leer PLACA, TD, CD

TCSR ← TCSR + CD

EN CASO DE TD HACER

1: SI CD ≥ 1'000.000

ENTONCES VR ← 1'000.000

SINO VR ← CD

FINSI

2: SI CD ≥ 500.000

ENTONCES VR ← 500.000

SINO VR ← CD

FINSI

3: **SI** CD \geq 300.000

ENTONCES VR \leftarrow 300.000

SINO VR \leftarrow CD

FINSI

FIN_CASO

TCCR \leftarrow TCCR + VR

FIN_PARA

Paso 6 **IMPRIMIR** TCCR, TCSR

Paso 7 **FIN**

Otra forma de resolver el mismo ejercicio. Tercera versión

Algoritmo.

Paso 1 **INICIO**

Paso 2 TCSR \leftarrow 0

Paso 3 TCCR \leftarrow 0

Paso 4 **LEER** N

Paso 5 **PARA** J DE 1 HASTA N **HACER**

Leer PLACA, TD, CD

TCSR \leftarrow TCSR + CD

SI TD = 1

ENTONCES SI CD \geq 1'000.000

ENTONCES VR \leftarrow 1'000.000

SINO VR \leftarrow CD

FINSI

FINSI

SI TD = 2

ENTONCES SI CD >= 500.000

ENTONCES VR ← 500.000

SINO VR ← CD

FINSI

FINSI

SI TD = 3

ENTONCES SI CD >= 300.000

ENTONCES VR ← 300.000

SINO VR ← CD

FINSI

FINSI

TCCR ← TCCR + VR

FIN_PARA

IMPRIMIR TCCR, TCSR

Paso 6 FIN

EJERCICIO

La compañía Caldas S.A. desea obtener el total de ventas realizadas durante 1989. La compañía ofrece cuatro formas de crédito diferentes al mismo tiempo que puede vender artículos de contado. La información leída por cada venta es la siguiente:

Tipo de pago, cantidad de artículos vendidos y precio unitario.

Si el tipo de pago = 1, la forma de pago es a 10 días
= 2, la forma de pago es a 20 días
= 3, la forma de pago es a 30 días
= 4, la forma de pago es a 60 días
= 5, la forma de pago es de contado
= 6, no debe procesarse por ser el final
de datos.

Calcular:

- 1- Ventas totales realizadas por la compañía
- 2- Total de ventas realizadas por cada una de las cuatro formas de pago a crédito.
- 3- Total de ventas de contado

Solución

a) Análisis

Entrada: Leer tipo de pago, cantidad de artículos vendidos y precio unitario.

Inicializar en cero cada una de las cuatro formas de pago a crédito, así como las ventas totales y las ventas de contado.

Proceso: Como no se conoce el total de datos a procesar se utiliza un registro centinela, en este caso, cuando el tipo de pago sea igual a 6 concluye la lectura de datos.

Se calcula la venta del artículo como precio

unitario por cantidad.

Total ventas = total de ventas + venta de cada artículo.

Se utiliza una estructura **EN CASO DE... HACKER** y dependiendo del tipo de venta se incrementa el correspondiente acumulador.

Salida: Ventas a 10 días, ventas a 20 días, ventas a 30 días, ventas a 60 días, ventas de contado, ventas totales.

b) Variables a utilizar

VEN10 = Ventas a 10 días

VEN20 = Ventas a 20 días

VEN30 = Ventas a 30 días

VEN60 = Ventas a 60 días

VENCON = Ventas de contado

VENTOT = Ventas totales

TIPO = Tipo de pago

CAN = Cantidad de artículos

PU = Precio unitario

VENTA = Venta realizada

c) Algoritmo

Paso 1	INICIO
Paso 2	VEN10 ← 0
Paso 3	VEN20 ← 0
Paso 4	VEN30 ← 0
Paso 5	VEN60 ← 0

Paso 6 VENCON ← 0

Paso 7 VENTOT ← 0

Paso 8 Leer TIPO, CAN, PU

Paso 9 **MIENTRAS TIPO <> 8 HACER**

 VENTA ← PU * CAN

 VENTOT ← VENTOT + VENTA

EN CASO DE TIPO HACKR

 1: VEN10 ← VEN10 + VENTA

 2: VEN20 ← VEN20 + VENTA

 3: VEN30 ← VEN30 + VENTA

 4: VEN60 ← VEN60 + VENTA

 5: VENCON ← VENCON + VENTA

FIN_CASO

 Leer TIPO, CAN, PU

FIN_MIENTRAS

Paso 10 **IMPRIMIR VEN10, VEN20, VEN30, VEN60,**

 VENTOT, VENCON.

Paso 11 **FIN**

EJERCICIO

Se tiene un inventario de 50 artículos y por cada uno de ellos se dispone de la siguiente información: código del artículo, cantidad mínima, cantidad máxima, saldo y costo unitario. Se desea un reporte de aquellos artículos que tienen poca o ninguna existencia en el almacén calculando

el número de unidades a ordenar y el costo de la orden para ese artículo. Al final del informe se desea saber cuál es el costo total de todos los pedidos.

Solución

a) Análisis

Entrada: Inicializar el contador de registros y el costo total de todos los pedidos en cero.

Leer código, cantidad mínima, cantidad máxima, saldo y costo unitario.

Proceso: Si el saldo es menor que la cantidad mínima, se determina el pedido a realizar.

$E_k \text{ pedido} = \text{cantidad máxima} - \text{saldo}$

El costo del pedido = pedido x costo unitario

El costo total de los pedidos = sumatoria de los pedidos.

Salida: Imprimir código, cantidad mínima, cantidad máxima, saldo, costo unitario, pedido, costo del pedido y costo total de todos los pedidos.

b) Variables a utilizar:

CR = Contador de registros

COD = Código

CMIN = Cantidad mínima

CMAX = Cantidad máxima

SALDO = Saldo

CUNIT = Costo unitario

PED = Pedido a efectuar

COSPED = Costo del pedido

CTP = Costo total de los pedidos.

c) Algoritmo

```

Paso 1          INICIO
Paso 2          CR ← 1
Paso 3          CTP ← 0
Paso 4          REPETIR
                  Leer COD, CMIN, CMAX, SALDO, CUNIT
                  SI SALDO >= CMIN
                    ENTONCES PED ← 0
                    SINO PED ← CMAX - SALDO
                  FINSI
                  COSPED ← PED * CUNIT
                  IMPRIMIR COD, CMIN, CMAX, SALDO, CUNIT,
                    PED, COSPED
                  CTP ← CTP + COSPED
                  CR ← CR + 1
                  HASTA_QUE CR > 50
Paso 5          IMPRIMIR CTP
Paso 6          FIN

```

Otra solución al mismo ejercicio empleando la estructura

PARA, segunda versión.

Algoritmo

```

Paso 1          INICIO
Paso 2          CTP ← 0
Paso 3          PARA I DE 1 HASTA 50 HACER
                  Leer COD, CMIN, CMAX, SALDO, CUNIT
                  SI SALDO >= KMIN
                      ENTONCES PED ← 0
                      SINO PED ← CMAX - SALDO
                  FINSI
                  COSPED ← PED * CUNIT
                  IMPRIMIR COD, CMIN, CMAX, SALDO, CUNIT,
                      PED, COSPED
                  CTP ← CTP + COSPED
                  FIN_PARA
Paso 4          IMPRIMIR CTP
Paso 5          FIN

```

EJERCICIO

Se tiene un grupo de N estudiantes y por cada uno de ellos se dispone de la siguiente información: código del estudiante, nombre, primer parcial, segundo parcial, tercer parcial.

La nota definitiva se calcula como el promedio de los tres parciales. Imprimir por cada estudiante, su código, nombre, cada una de las notas parciales y su calificación

definitiva.

Calcular e imprimir además: el total de estudiantes reprobados, total de estudiantes cuya nota definitiva se encuentre entre 3 y 4, y el total de estudiantes cuya nota definitiva sea superior a 4. La escala de calificaciones es de 0 a 5.

Solución

a) Análisis

Entrada: Inicializar: el contador de estudiantes reprobados en cero.

el contador de notas entre 3 y 4 en cero.

el contador de notas mayores a 4 en cero.

Leer el límite de estudiantes N.

Proceso: Calcular la nota definitiva como (primer parcial + segundo parcial + tercer parcial) / 3, para luego compararla contra 3 y así incrementar el contador de reprobados si es menor que 3, igual proceso se hace con las notas que se encuentran entre 3 y 4 y las superiores a 4.

Salida: Imprimir código, nombre, primer parcial, segundo parcial, tercer parcial, calificación definitiva, contador de reprobados, contador de notas entre 3 y 4 y contador de notas mayores de 4.

b) Variables a utilizar:

CREP = Contador de reprobados

CN3Y4 = Contador de notas entre 3 y 4

CNMA4 = Contador de notas mayores de 4

N = Número total de estudiantes

COD = Código

NOM = Nombre

PP = Primer parcial

SP = Segundo parcial

TP = Tercer parcial

CDEF = Calificación definitiva

c) Algoritmo

```

Paso 1          INICIO
Paso 2          CREP ← 0
Paso 3          CN3Y4 ← 0
Paso 4          CNMA4 ← 0
Paso 5          Leer N
Paso 6          PARA I DE 1 HASTA N HACER
                  Leer COD, NOM, PP, SP, TP
                  CDEF ← (PP+SP+TP)/3
                  IMPRIMIR COD, NOM, PP, SP, TP, CDEF
                  SI CDEF < 3
                    ENTONCES CREP ← CREP + 1
                  SINO SI CDEF <= 4

```

ENTONCES CN3Y4 ← CN3Y4 + 1

SINO CNMA4 ← CNMA4 + 1

FINSI

FINSI

FIN_PARA

Paso 7 IMPRIMIR CREP, CN3Y4, CNMA4

Paso 8 FIN

EJERCICIO

Se tiene una serie de registros cada uno con la siguiente terna de valores, X, Y, Z. El proceso termina cuando Z=000.

Calcular y escribir:

$$X \leq Y \quad \underline{\hspace{2cm}} \quad T = ZX + Y$$

$$X > Y \quad \underline{\hspace{2cm}} \quad T = X - Y$$

$$Y < Z \quad \underline{\hspace{2cm}} \quad T = X + Y + Z$$

$$Y \geq Z \quad \underline{\hspace{2cm}} \quad T = 3X + 2Y - Z$$

Hallar ΣT ; $P = \frac{\Sigma X}{\Sigma Y}$; $Q = \Sigma \frac{X}{Z}$

Solución

a) Análisis

Entrada: Inicializar en cero la sumatoria de X, sumatoria de Y, sumatoria de T y sumatoria de X/Z

Leer los valores X, Y, Z

Proceso: Se utiliza como control de parada, la comparación

de Z contra 000. en caso de ser diferente, se hacen los respectivos cálculos que se originan de las comparaciones de X contra Y y de Y contra Z. Cabe anotar que por cada ciclo ejecutado, se calculan dos valores de T, los cuales deben ser almacenados.

Salida: Imprimir la sumatoria de T, P y Q.

b) Variables a utilizar

X, Y, Z = Valores a ser leídos

SX = Sumatoria de X

SY = Sumatoria de Y

ST = Sumatoria de T

SXZ = Sumatoria de X/Z

c) Algoritmo

Paso 1	INICIO
Paso 2	SX ← 0
Paso 3	SY ← 0
Paso 4	ST ← 0
Paso 5	SXZ ← 0
Paso 6	Leer X, Y, Z
Paso 7	MIENTRAS Z <> 000 HACER
	SI X > Y
	ENTONCES T ← X - Y
	SINO T ← Z * X + Y

FINSI

$SX \leftarrow SX + X$

$SY \leftarrow SY + Y$

$SXZ \leftarrow SXZ + X/Z$

$ST \leftarrow ST + T$

SI $Y \geq Z$

ENTONCES $T \leftarrow 3 * X + 2 * Y - Z$

SINO $T \leftarrow X + Y + Z$

FINSI

$ST \leftarrow ST + T$

Leer X, Y, Z

Paso 8 **FIN_MIENTRAS**
 Paso 9 $P \leftarrow SX / SY$
 Paso 10 $Q \leftarrow SXZ$
 Paso 11 **IMPRIMIR** ST, P, Q
 Paso 12 **FIN**

2.6 PROBLEMAS PROPUESTOS

1- Dado un conjunto de N valores, determinar la media aritmética según la siguiente fórmula:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$$

2- Se dispone de un conjunto de registros, cada uno con una terna de valores X, Y, Z, el proceso termina cuando $Z = -99$

$$X < Y \quad \text{-----} \quad A = 3X^2 + 2Y - Z$$

$$X = Y \quad \underline{\hspace{2cm}} \quad A = 2X^3 - 2Y + Z$$

$$X > Y \quad \underline{\hspace{2cm}} \quad A = X + Y + Z$$

$$X > Z \quad \underline{\hspace{2cm}} \quad A = 5X + 2Y - 3Z$$

$$X < Z \quad \underline{\hspace{2cm}} \quad A = X - 5Y + Z^2$$

$$X = Z \quad \underline{\hspace{2cm}} \quad A = 7X - 7Y - 7Z$$

$$\text{Calcular } \Sigma A; \Sigma X; \frac{\Sigma Y}{\Sigma Z}$$

3- El gerente de un almacén desea saber cuántas ventas se realizarán entre 0 y \$5.000, entre 5001 y 10.000, entre 10001 y más.

Además, desea el total de ventas realizadas así como el total recaudado por ventas en un mes determinado.

4- La función exponencial e^x se puede expresar como:

$$e^x = 1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \frac{X^4}{4!} + \dots$$

Calcular e^x para valores de X que se encuentren entre $0 \leq X \leq 1$ y que sufre incrementos de 0.05.

5- Leer un conjunto de valores X dado en radianes y calcular el seno X, según la siguiente expresión:

$$\text{SEN } X = \frac{X}{1!} - \frac{X^3}{3!} + \frac{X^5}{5!} - \frac{X^7}{7!} + \dots$$

6- Calcular $Y = 1 + \frac{X^2}{2!} + \frac{X^4}{4!} + \frac{X^6}{6!} + \frac{X^8}{8!} + \dots + \frac{X^{30}}{30!}$

7- Calcular $Y = 1 - \frac{X^2}{2!} + \frac{X^4}{4!} - \frac{X^6}{6!} + \frac{X^8}{8!} - \dots + \frac{X^{30}}{30!}$

8- Leer en J registros el valor de A. Determinar:

$$Z = (A+7)^2 \quad \text{si } 1 \leq A \leq 3$$

$$Z = \sqrt{3A-5} \quad \text{si } 4 \leq A \leq 7$$

$$Z = 7A/(A-7)^3 \quad \text{si } 8 \leq A \leq 10$$

Imprimir A y el correspondiente valor Z

CAPITULO III

ARREGLOS

3.1 INTRODUCCION

Supóngase que existe un conjunto de 30 notas pertenecientes a un grupo de estudiantes de matemáticas y se desea obtener su promedio. Para solucionar este problema con las herramientas estudiadas hasta el momento, se tendrán que realizar treinta ciclos y por cada ciclo leer una nota, para luego acumularla y por último hallar su promedio; ó para evitar las treinta lecturas, se podría realizar una sola que comprenda las 30 variables, siendo este método el menos indicado porque se tendrían tantas variables como datos se quisieran analizar. Para obviar lo anterior se dispone de otra herramienta de programación como son los **ARREGLOS**.

3.2 DEFINICION

Los arreglos permiten representar un conjunto finito de datos del mismo tipo bajo el nombre de una variable.

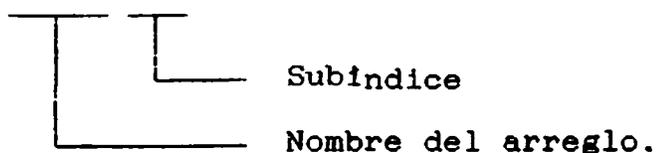
Los arreglos se componen del nombre del arreglo y del subíndice o suscrito, expresándose el subíndice a

continuación del nombre del arreglo mediante el empleo de paréntesis.

Cuando se hace referencia a un valor específico del arreglo se le denomina **ELEMENTO**.

Cada elemento del arreglo ocupa un espacio fijo y contiguo en la memoria del computador, igual al que ocuparía el elemento si fuera tratado como una variable independiente.

NOTAS (M)



El valor dentro del paréntesis o suscrito puede ser:

Una variable. Ejemplo NOTAS(M), X(L)

Una constante. Ejemplo NOTAS(30), Y(8)

Una expresión. Ejemplo NOTAS(I+2), Z(2*N)

Los suscritos pueden ser variables, constantes o expresiones, pero siempre de tipo entero.

Es necesario diferenciar entre el valor o dato y la posición que éste ocupa en el arreglo. Por ejemplo.

Se tienen los siguientes valores 8, 7, 1, -40, 3 y se van a almacenar en un arreglo X.

X				
X(1)	X(2)	X(3)	X(4)	X(5)
8	7	1	-40	3

El valor 8 está en la posición 1 del arreglo X; $X(1) = 8$

El valor -40 está en la posición 4 del arreglo X; $X(4) = -40$

La posición 5ª del arreglo X tiene almacenado el valor 3;

$X(5) = 3$

La posición 2ª del arreglo X tiene almacenado el valor 7;

$X(2) = 7$

EJERCICIO

Asumiendo que se tiene en memoria un arreglo A con los siguientes valores, calcular las diferentes formas de suscrito presentadas a continuación:

A															
A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	K	L	M	J
5	21	7.1	4	3	20	3.5	9	13	-80	1	2	6	4

M E M O R I A

1. Una constante entera.

$Z \leftarrow A(5) * 3$

$Z \leftarrow 3 * 3$

$Z \leftarrow 9$

2. Una variable entera.

$Z \leftarrow A(M)/2$

$Z \leftarrow A(6)/2$

$$Z \leftarrow 20/2$$

$$Z \leftarrow 10$$

3. Una expresión de la forma $(V + C)$, donde V es una variable y C es una constante.

$$Z \leftarrow A(L + 2) \uparrow 2$$

$$Z \leftarrow A(2 + 2) \uparrow 2$$

$$Z \leftarrow A(4) \uparrow 2$$

$$Z \leftarrow 4 \uparrow 2$$

$$Z \leftarrow 16$$

4. Una expresión de la forma $(V - C)$

$$Z \leftarrow A(J - 2) + 5$$

$$Z \leftarrow A(4 - 2) + 5$$

$$Z \leftarrow A(2) + 5$$

$$Z \leftarrow 21 + 5$$

$$Z \leftarrow 26$$

5. Una expresión de la forma $(C * V)$

$$Z \leftarrow A(2 * J)/2$$

$$Z \leftarrow A(2 * 4)/2$$

$$Z \leftarrow A(8)/2$$

$$Z \leftarrow 9/2$$

$$Z \leftarrow 4.5$$

6. Una expresión de la forma $(C * V + C')$

$$Z \leftarrow A(3 * L + 4)/8$$

$$Z \leftarrow A(3 * 2 + 4)/8$$

$$Z \leftarrow A(10)/8$$

$$Z \leftarrow -80/8$$

$$Z \leftarrow -10$$

7. Una expresión de la forma $(C * V - C')$

$$Z \leftarrow A(5 * J - 15) \uparrow 2$$

$$Z \leftarrow A(5 * 4 - 15) \uparrow 2$$

$$Z \leftarrow A(5) \uparrow 2$$

$$Z \leftarrow 3 \uparrow 2$$

$$Z \leftarrow 9$$

EJERCICIO

Leer un arreglo de N elementos y hallar la suma de todos sus elementos. Imprimir el vector original y la suma.

Solución

a) Análisis

Entrada: Leer el límite del arreglo N

Leer el arreglo

Iniciar sumatoria de términos en cero y el subíndice o variable que maneja la posición del vector en 1.

Proceso: Mientras el subíndice sea inferior al límite del vector se debe acumular su contenido.

Salida: Imprimir el vector leído, así como la suma.

b) Variables a utilizar

N = Límite de elementos del vector

SUM = Sumatoria de elementos

X = Nombre del arreglo

J = Subíndice o posición del arreglo

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer N
Paso 3          SUM ← 0
Paso 4          J ← 1
Paso 5          REPETIR
                Leer X(J)
                IMPRIMIR X(J)
                SUM ← SUM + X(J)
                J ← J + 1
                HASTA QUE J > N
Paso 6          IMPRIMIR SUM
Paso 7          FIN
  
```

d) Prueba de escritorio

Sea el vector 1, 7, 3, 8

N	SUM	J	X(J)
4	0	1	X(1) = 1
	0+1 = 1	2	X(2) = 7
	1+7 = 8	3	X(3) = 3
	8+3 = 11	4	X(4) = 8
	11+8 = 19	5	

Otra solución al mismo ejercicio es:

Algoritmo

Paso 1 **INICIO**
Paso 2 **SUM ← 0**
Paso 3 **Leer N**
Paso 4 **PARA J DE 1 HASTA N HACER**
 Leer X(J)
 SUM ← SUM + X(J)
 IMPRIMIR X(J)
 FIN_PARA
Paso 5 **IMPRIMIR SUM**
Paso 6 **FIN**

EJERCICIO

Calcular la suma de los cuadrados de los 20 elementos de un arreglo X.

Solución**a) Análisis**

Entrada: Se inicializa el subíndice del vector en 1 y la sumatoria de los cuadrados en cero.

Leer el arreglo.

Proceso: Se controla la posición del arreglo con el subíndice y se eleva al cuadrado el contenido de cada posición hasta completar sus 20 elementos.

Salida: Imprimir la sumatoria

b) Variables a utilizar

I = Subíndice del arreglo

SUM = Suma de cuadrados

X = Nombre del vector

c) Algoritmo

```

Paso 1          INICIO
Paso 2          I ← 1
Paso 3          SUM ← 0
Paso 4          REPETIR
                  Leer X(I)
                  SUM ← SUM + X(I)²
                  I ← I + 1
                  HASTA QUE I > 20
Paso 5          IMPRIMIR SUM
Paso 6          FIN
  
```

Observación

A los arreglos de una dimensión se les denomina **VECTORES** y su primera posición puede arrancar en cero o en uno, dependiendo del lenguaje de programación que se trabaje.

d) Prueba de escritorio

Sea el arreglo X:

2	5	1	4	3	9
X(1)	X(2)	X(3)	X(4)	X(5)	X(20)

I	X(I)	SUM
1	X(1) = 2	0+2 ² = 4
2	X(2) = 5	4+5 ² = 29
3	X(3) = 1	29+1 ² = 30
4	X(4) = 4	30+4 ² = 46
5	X(5) = 3	46+3 ² = 55
.	.	.
.	.	.
.	.	.
20	X(20) = 9	

Otra solución al ejercicio empleando la estructura PARA es:

Algoritmo

```

Paso 1          INICIO
Paso 2          SUM ← 0
Paso 3          PARA I DE 1 HASTA 20 HACER
                Leer X(I)
                SUM ← SUM + X(I)2
                FIN_PARA
Paso 4          IMPRIMIR SUM
Paso 5          FIN

```

Observación

Quando se trabajan arreglos como parte del rango de una estructura PARA, el subíndice debe ser idéntico al índice de la estructura.

EJERCICIO

Se tienen 2 vectores A y B de 10 elementos cada uno y se desea crear otro vector que contenga la multiplicación de los vectores leídos.

Solución**a) Análisis**

Entrada: Inicializar el subíndice en 1

Leer los dos vectores originales.

Proceso: Se toma la primera posición del vector A y se multiplica por la primera posición del vector B, el resultado se almacena en la primera posición del vector C. Este proceso se repite tantas veces como elementos contengan los vectores.

Salida: Imprimir los 3 vectores

b) Variables a utilizar

L = Subíndice de los arreglos

A, B = Nombre de los vectores originales

C = Nombre del vector resultante

c) Algoritmo

Paso 1	INICIO
Paso 2	L ← 1
Paso 3	REPETIR

Leer A(L), B(L)

C(L) ← A(L) * B(L)

IMPRIMIR A(L), B(L), C(L)

L ← L + 1

HASTA_QUE L > 10

Paso 4

FIN

EJERCICIO

Calcular la media aritmética (\bar{X}) de un conjunto de N valores mediante el empleo de arreglos. Imprimir el vector leído y su media aritmética.

Solución

a) Análisis

Entrada: Inicializar la sumatoria de elementos en cero

Leer el vector X y su límite N.

Proceso: Aplicar la siguiente fórmula:

$$\bar{X} = \frac{\sum_{i=1}^N X_i}{N}$$

Leer cada elemento del vector y acumularlo, para finalmente dividir el acumulado en el número total de elementos y obtener la media aritmética.

Salida: Imprimir el vector leído y la media aritmética.

b) Variables a utilizar

N = Número de valores

SUMA = Sumatoria de valores

MEDIA = Media aritmética

X = Nombre del vector

I = Subíndice

c) Algoritmo

```

Paso 1          INICIO
Paso 2          SUMA ← 0
Paso 3          I ← 1
Paso 4          Leer N
Paso 5          REPETIR
                  Leer X(I)
                  IMPRIMIR X(I)
                  SUMA ← SUMA + X(I)
                  I ← I + 1
                  HASTA QUE I > N
                  MEDIA ← SUMA / N
Paso 6          IMPRIMIR MEDIA
Paso 7          FIN
  
```

Otra versión del ejercicio anterior es:

Algoritmo

```

Paso 1          INICIO
Paso 2          SUMA ← 0
  
```

Paso 3 **Leer N**
Paso 4 **PARA I DE 1 HASTA N HACER**
 Leer X(I)
 IMPRIMIR X(I)
 SUM ← SUM + X(I)
 FIN_PARA
 MEDIA ← SUM / N
Paso 5 **IMPRIMIR MEDIA**
Paso 6 **FIN**

De no quererse leer o imprimir simultaneamente dentro del cuerpo de la estructura PARA se requiere hacer lo siguiente.

Algoritmo

Paso 1 **INICIO**
Paso 2 **SUMA ← 0**
Paso 3 **Leer N**
Paso 4 **PARA I DE 1 HASTA N HACER**
 Leer X(I)
 FIN_PARA
Paso 5 **PARA J DE 1 HASTA N HACER**
 SUMA ← SUMA + X(J)
 FIN_PARA
Paso 6 **MEDIA ← SUMA / N**
Paso 7 **PARA K DE 1 HASTA N HACER**
 IMPRIMIR X(K)

FIN_PARA

Paso 8 **IMPRIMIR MEDIA**

Paso 9 **FIN**

En este ejercicio los índices I, J, K de las estructuras **PARA**, señalan la posición del vector a la cual se quiere referir, por lo tanto, no presentan incompatibilidad en la utilización porque todas ellas tienen sus límites definidos, por ejemplo:

PARA I DE 1 HASTA N HACER

Leer X(I)

FIN_PARA

Cuando I vale 1, lee el contenido de la posición 1 del vector X, es decir lee X(1), así:

PARA K DE 1 HASTA N HACER

IMPRIMIR X(K)

FIN_PARA

Cuando K vale 1 imprime el contenido de la posición 1 del vector X, es decir, imprime X(1) siendo el contenido del arreglo el mismo, lo que cambia es la forma como se llama. Resumiendo, lo que se lee, escribe o procesa, realmente no son los subíndices I, J, K, sino los valores que esas variables representan, siendo su denominación transparente para el computador.

Observación

Cuando se trabaja con variables suscritas, se debe reservar una ó más posiciones de memoria dependiendo de las necesidades del problema, lo que ocasiona que luego de ser leído el arreglo permanezca almacenado en memoria, facilitando las operaciones y escrituras de éstos y otros vectores resultantes cuando se deseen mediante la manipulación de cualquier variable que actúe como subíndice y no necesariamente con el mismo subíndice como fue leído o escrito el arreglo en ocasiones anteriores. Esas reservaciones de memoria son transparentes a las soluciones algorítmicas, por lo tanto no requieren de alguna especificación adicional.

EJERCICIO

Calcular la media aritmética ponderada según la siguiente fórmula:

$$\bar{X}_p = \frac{\sum_{I=1}^N X_i \cdot W_i}{\sum_{I=1}^N W_i}$$

Solución**a) Análisis**

Entrada: Inicializar la sumatoria del numerador, es decir, de los $X_i \cdot W_i$ en cero, la sumatoria del

denominador, o sea, de los W_i en cero y el subíndice en 1.

Leer el límite de los 2 vectores

Proceso: Calcular la sumatoria de los $X(I) * W(I)$ y la sumatoria de los $W(I)$

Salida: Imprimir la media aritmética ponderada.

b) Variables a utilizar

SXW = Sumatoria de los $X_i * W_i$

SW = Sumatoria de los W_i

X, W = Vectores originales

N = Límite de elementos de los vectores

I = Subíndice. común a ambos vectores

MAP = Media aritmética ponderada

c) Algoritmo

Paso 1	INICIO
Paso 2	SXW \leftarrow 0
Paso 3	SW \leftarrow 0
Paso 4	I \leftarrow 1
Paso 5	Leer N
Paso 6	REPETIR
	Leer X(I), W(I)
	SXW \leftarrow SXW + X(I) * W(I)
	SW \leftarrow SW + W(I)
	I \leftarrow I + 1

```

                HASTA_QUE I > N
Paso 7          MAP ← SXW / SW
Paso 8          IMPRIMIR MAP
Paso 9          FIN

```

Otra versión del ejercicio es:

Algoritmo

```

Paso 1          INICIO
Paso 2          SXW ← 0
Paso 3          SW ← 0
Paso 4          Leer N
Paso 5          PARA I DE 1 HASTA N HACER
                Leer X(I), W(I)
                SXW ← SXW + X(I) * W(I)
                SW ← SW + W(I)
                FIN_PARA
Paso 6          MAP ← SXW / SW
Paso 7          IMPRIMIR MAP
Paso 8          FIN

```

EJERCICIO

Leer un vector de K posiciones y ordenarlo ascendentemente.
 Imprimir el vector leído y el vector ordenado.

Solución

a) Análisis

Entrada: Leer el vector a ser ordenado y su límite.

Proceso: Se ordena el primer elemento del vector con respecto al resto de elementos, luego el segundo elemento y así sucesivamente.

Se utilizan 2 estructuras PARA, la estructura externa variando de 1 hasta el límite -1 y la estructura interna variando desde 2 hasta el límite del vector; ésto con el propósito de no comparar el primer elemento consigo mismo, y también para no comparar el último término con el siguiente, que no existe.

Es necesario utilizar una variable temporal, por ejemplo si $B(M)$ tiene un valor de 3 y $B(J)$ tiene un valor de 2, al ordenarlo, se almacena el contenido de $B(J)$ en $B(M)$, perdiéndose el valor original que éste tenía; para evitarlo, se utiliza una variable temporal que almacene el valor de $B(M)$ antes de trasladar el valor de $B(J)$ a $B(M)$, luego el contenido de la variable temporal se almacena en $B(J)$ obteniéndose su ordenamiento.

Salida: Imprimir el vector original y el vector ordenado.

b) Variables a Utilizar

B = Nombre del vector

K = Número de elementos del vector B

C = Variable temporal

N = Penúltimo término del vector

I = Siguiete término del vector.

J, M = Variables indices de la estructura PARA

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer K
Paso 3          PARA J DE 1 HASTA K HACER
                  Leer B(J)
                  IMPRIMIR B(J)
                  FIN_PARA
Paso 4          N ← K - 1
Paso 5          PARA M DE 1 HASTA N HACER
                  I ← M + 1
                  PARA J DE 1 HASTA K HACER
                      SI B(M) > B(J)
                          ENTONCES C ← B(M)
                                      B(M) ← B(J)
                                      B(J) ← C
                  FINSI
                  FIN_PARA
                  FIN_PARA
Paso 6          PARA J DE 1 HASTA K HACER
                  IMPRIMIR B(J)
                  FIN_PARA
Paso 7          FIN

```


ENTONCES $C \leftarrow B(J)$

$B(J) \leftarrow B(J+1)$

$B(J+1) \leftarrow C$

$J \leftarrow 1$

SINO $J \leftarrow J + 1$

FINSI

HASTA_QUE $J = K$

Paso 6

PARA J DE 1 HASTA K HACER

IMPRIMIR $B(J)$

FIN_PARA

Paso 7

FIN

3.3 PROBLEMAS PROPUESTOS

1- Analizar el siguiente algoritmo mediante una prueba de escritorio.

INICIO

IMPRIMIR "llenado de un vector"

Leer N

PARA I DE 1 HASTA N HACER

Leer X

$A(I) \leftarrow X$

IMPRIMIR $A(I)$

FIN_PARA

FIN

2- Probar el siguiente algoritmo con los datos: 1, 3, 5, 7,

INICIO

PARA K DE 1 HASTA 5 HACER

Leer X(K)

FIN_PARA

K ← 5

PARA J DE 1 HASTA 5 HACER

Y(K) ← X(J)

K ← K - 1

FIN_PARA

PARA J DE 1 HASTA 5 HACER

X(J) ← Y(J)

IMPRIMIR X(J)

FIN_PARA

FIN

3- Analizar el algoritmo presentado a continuación mediante una prueba de escritorio tomando como base los siguientes datos:

7, 1, 3, 0, 15, 12, 2, 13, 7, 14, 6, 3, 1, 18, 5

INICIO

SUMA ← 0

J ← 1

REPETIR

Leer Y (J)

SUMA ← SUMA + Y (J)

```

    J ← J + 2
HASTA_QUE J > 13
  IMPRIMIR SUMA
FIN

```

4- INICIO

```

    I ← 0
  REPETIR
    Leer A (I)
    I ← I + 1
  HASTA_QUE I > 9
FIN

```

Cuántos datos se leen en el anterior algoritmo ?

5- INICIO

```

  PARA I DE 1 HASTA 5 HACER
    PARA J DE 1 HASTA 3 HACER
      Leer X (J)
    FIN_PARA
  FIN_PARA
FIN

```

Cuántas lecturas se realizan ?

6- Calcular la varianza de un vector de N posiciones, según la siguiente fórmula:

$$\text{VARIANZA} = \sigma^2 = \sum_{i=1}^N (X_i - \bar{X})^2$$

7- La ecuación de la recta es: $Y = a + bX$

Donde:

$$b = \frac{\sum XY - n \bar{X} \bar{Y}}{\sum X^2 - n \bar{X}^2}$$

$$a = \bar{Y} - b \bar{X}$$

Según las anteriores fórmulas del método de mínimos cuadrados, determinar la ecuación de la recta para los siguientes valores.

X	Y
1	0
2	1
3	2
4	3
5	4
6	5
7	6

8- Leer un vector A de N elementos y calcular la varianza, según la siguiente fórmula.

$$\text{VARIANZA} = \sigma_X^2 = \frac{1}{N-1} \left[\sum_{i=1}^N (X_i - \bar{X})^2 \right]$$

9- Se tiene un par de vectores R y S de N elementos cada

uno, Calcular C como $C = \sum_{K=1}^N \frac{B}{K}$

$$B = \begin{matrix} R & * & S \\ K & K & K \end{matrix} \quad \text{si } R = S \\ \begin{matrix} K & & K \end{matrix}$$

$$B = \begin{matrix} & SK \\ R & \\ K & K \end{matrix} \quad \text{si } R < S \\ \begin{matrix} K & & K \end{matrix}$$

$$B = \begin{matrix} R & - & S \\ K & K & K \end{matrix} \quad \text{si } R > S \\ \begin{matrix} K & & K \end{matrix}$$

10- Leer 2 arreglos de K posiciones y encontrar los elementos que sean iguales en los dos arreglos.

11- Leer un vector de 20 elementos y determinar cual es el elemento mayor del vector e imprimirlo, así como el vector leído.

12- Leer un vector de N elementos, determinar e imprimir cuántos números repetidos hay y cuáles faltan entre el menor y el mayor.

13- Leer un vector de N elementos y calcular su mediana. Imprimir el vector leído y la mediana.

La mediana es el elemento que se encuentra en la mitad del vector luego que éste ha sido ordenado ascendente-mente.

3.4 PROBLEMAS RESUELTOS

EJERCICIO

La compañía Manizales requiere actualizar su inventario con base al movimiento de ventas. El inventario está compuesto de 50 artículos y dispone de la siguiente información por artículo: código, nombre, cantidad y valor unitario.

La compañía realizó 30 ventas de artículos y la información que se tiene por cada venta es: código del artículo vendido y cantidad vendida.

Imprimir: inventario inicial, movimiento de ventas, inventario final actualizado y valor total del inventario final.

Solución

a) Análisis

Entrada: Leer código, nombre, cantidad y valor unitario de cada artículo del inventario inicial, así como código del artículo vendido y cantidad vendida.

Proceso: Cuando el código del artículo vendido sea igual al código del artículo en existencias, se resta la cantidad vendida de la cantidad en existencias. La venta de un artículo se calcula multiplicando la cantidad vendida por el valor unitario.

El total del inventario final se calcula sumando las ventas parciales de artículos.

Salida: Imprimir el inventario inicial, el movimiento de ventas, el inventario final actualizado y el valor

total del inventario final.

b) Variables a utilizar

I = Subíndice

COD = Código del artículo

NOM = Nombre del artículo

CAN = Cantidad del artículo

VUNIT = Valor unitario del artículo

CODV = Código del artículo vendido

CANV = Cantidad vendida

VAR = Valor del artículo en existencias

VIF = Valor del inventario final

c) Algoritmo

Paso 1 **INICIO**

Paso 2 **PARA I DE 1 HASTA 50 HACER**

 Leer COD(I), NOM(I), CAN(I), VUNIT(I)

IMPRIMIR COD(I), NOM(I), CAN(I), VUNIT(I)

FIN_PARA

Paso 3 **PARA K DE 1 HASTA 30 HACER**

 Leer CODV, CANV

IMPRIMIR CODV, CANV

PARA I DE 1 HASTA 50 HACER

SI CODV = COD(I)

ENTONCES CAN(I) ← CAN(I) - CANV

FINSI

FIN_PARA

FIN_PARA

Paso 4

VIF ← 0

Paso 5

PARA I DE 1 HASTA 50 HACER

VAR(I) ← CAN(I) * VUNIT(I)

IMPRIMIR COD(I),NOM(I),CAN(I),VUNIT(I),VAR(I)

VIF ← VIF + VAR(I)

FIN_PARA

Paso 6

IMPRIMIR VIF

Paso 7

FIN

EJERCICIO

Se tienen 3 arreglos con la siguiente información por estudiante: código del estudiante, valor por materia y número de materias. Generar un cuarto arreglo que contenga el valor a pagar por matrícula de cada uno de los N estudiantes sumándole \$1.500= por concepto de derechos médicos. Calcular además el total recaudado por pago de matrículas y la matrícula promedio por estudiante. Imprimir toda la información leída, así como la calculada.

Solución

a) Análisis

Entrada: Leer el número de elementos del vector, así como el código del estudiante, valor por materia y número de materias por estudiante.

Inicializar en cero el total pagado por matrícula.

Proceso: Calcular:

Total matrícula del estudiante = valor ^{materia} x
número de materias + 1500

Total pagado por matrículas = total pagado por
matrículas + total matrícula del estudiante.

Promedio = total pagado por matrículas / número de
estudiantes.

Salida: Imprimir código del estudiante, valor por materia,
número de materias del estudiante, total matrícula
del estudiante, total pagado por matrículas y
matrícula promedio.

b) Variables a utilizar

N = Número de estudiantes

COD = Código de estudiante

VM = Valor materia

NM = Número de materias

PROM = Promedio

TPM = Total pagado por matrículas

TM = Total matrícula del estudiante.

c) Algoritmo

Paso 1 INICIO

Paso 2 TPM ← 0

Paso 3 Leer N

Paso 4 **PARA I DE 1 HASTA N HACER**
 Leer COD(I), VM(I), NM(I)
 $TM(I) \leftarrow VM(I) * NM(I) + 1500$
 $TPM \leftarrow TPM + TM(I)$
 IMPRIMIR COD(I), VM(I), NM(I), TM(I)
 FIN_PARA

Paso 5 **PROM** \leftarrow TPM / N

Paso 6 **IMPRIMIR** TPM, PROM

Paso 7 **FIN**

EJERCICIO

Leer en N registros los valores de A y B, calcular e imprimir Y, siendo:

$$Y = \sum_{i=1}^N X_i$$

$$\text{Donde } X_i = A_i + B_i^2 \quad \text{si } A_i^2 - B_i^2 < 0$$

$$X_i = A_i - 3B_i \quad \text{si } A_i^2 - B_i^2 = 0$$

$$X_i = A_i + B_i - 7 \quad \text{si } A_i^2 - B_i^2 > 0$$

Solución**a) Análisis**

Entrada: Inicializar la sumatoria.

Leer el límite de los datos a ser procesados y por cada registro leer los valores A y B.

Proceso: Leída una pareja de datos A y B, se calcula $A^2 - B^2$ y el resultado se compara contra cero para determinar su correspondiente valor X(I) y luego acumularlo.

Salida: Imprimir la sumatoria

b) Variables a utilizar

SUM = Sumatoria

N = Límite de datos a ser procesados

A, B = Valores a ser leídos

X(I) = Vector resultante

c) Algoritmo

```

Paso 1          INICIO
Paso 2          SUM ← 0
Paso 3          Leer N
Paso 4          PARA I DE 1 HASTA N HACER
                  SI (A2 - B2) < 0
                    ENTONCES X(I) ← A+B2
                    SINO SI (A2 - B2) = 0
                      ENTONCES X(I) ← A-3*B
                      SINO X(I) ← A+B-7
                  FINSI
                FINSI
                SUM ← SUM + X(I)
FIN_PARA

```

Paso 5 **IMPRIMIR SUM**

Paso 6 **FIN**

EJERCICIO

Calcular la desviación estándar para un conjunto de K elementos, según las siguientes fórmulas:

$$\text{DESV} = \sqrt{\frac{\sum_{i=1}^K (X_i - \bar{X})^2}{K}}$$

$$\bar{X} = \frac{\sum_{i=1}^K X_i}{K}$$

Solución

a) Análisis

Entrada: Inicializar la sumatoria de las desviaciones, ó sea, la sumatoria del numerador y la sumatoria de los X_i en cero.

Leer el vector X y su límite

Proceso: Calcular primero la media aritmética \bar{X} , para luego calcular la sumatoria de las desviaciones o el numerador y por último se aplica la fórmula.

Salida: Imprimir el vector y la desviación estándar.

b) Variables a utilizar

K = Número de elementos del vector

SUMX = Sumatoria de los elementos del vector

NUM = Sumatoria de las desviaciones o numerador

MEDIA = Media aritmética

DESV = Desviación estándar.

c) Algoritmo

```

Paso 1          INICIO
Paso 2          NUM ← 0
Paso 3          SUMX ← 0
Paso 4          Leer K
Paso 5          PARA I DE 1 HASTA K HACER
Paso 6          Leer X(I)
                IMPRIMIR X(I)
                FIN_PARA
Paso 7          PARA I DE 1 HASTA K HACER
                SUMX ← SUMX + X(I)
                FIN_PARA
Paso 8          MEDIA ← SUMX / K
Paso 9          PARA I DE 1 HASTA K HACER
                NUM ← NUM + (X(I) - MEDIA) ↑ 2
                FIN_PARA
Paso 10         DESV ← (NUM / K) ↑ 0.5
Paso 11         IMPRIMIR DESV
Paso 12         FIN

```

EJERCICIO

Leer un arreglo de K posiciones. Determinar cual es el elemento mayor y la posición que éste ocupa dentro del

arreglo. Imprimir el vector leído, el elemento mayor y su posición.

Solución

a) Análisis

Entrada: Leer el número de elementos del vector, así como el vector.

Proceso: Se asigna la primera posición del vector a una variable y luego se compara ésta contra la segunda posición del vector para determinar cual de las dos es mayor. Esta comparación se debe hacer a través de todo el vector.

Tan pronto se determina cual es el elemento mayor, se hace otro recorrido a lo largo del vector y se compara el elemento mayor contra cada posición del vector, en donde coincidan se ordena imprimir su posición.

Salida: Imprimir el vector leído, el elemento mayor y su posición.

b) Variables a utilizar

K = Total de elementos del vector

Z = Nombre del vector

MAYOR = Elemento mayor (Inicialmente se asume igual a Z(1))

J = Posición del vector

c) Algoritmo

```

Paso 1      INICIO
Paso 2      Leer K
Paso 3      PARA J DE 1 HASTA K HACER
              Leer Z(J)
              IMPRIMIR Z(J)
              FIN_PARA
Paso 4      MAYOR ← Z(1)
Paso 5      PARA J DE 2 HASTA K HACER
              SI MAYOR < Z(J)
                ENTONCES MAYOR ← Z(J)
              FINSI
              FIN_PARA
Paso 6      IMPRIMIR MAYOR
Paso 7      PARA J DE 1 HASTA K HACER
              SI MAYOR = Z(J)
                ENTONCES IMPRIMIR J
              FINSI
              FIN_PARA
Paso 8      FIN

```

OTRA VERSION DEL EJERCICIO

Algoritmo

```

Paso 1      INICIO
Paso 2      Leer K

```

```

Paso 3      PARA J DE 1 HASTA K HACER
             Leer Z(J)
             IMPRIMIR Z(J)
             FIN_PARA
Paso 4      PARA J DE 1 HASTA (K-1) HACER
             SI Z(J) < Z(J+1)
             ENTONCES MAYOR ← Z(J+1)
             FINSI
             FIN_PARA
Paso 5      IMPRIMIR MAYOR
Paso 6      PARA J DE 1 HASTA K HACER
             SI MAYOR = Z(J)
             ENTONCES IMPRIMIR J
             FINSI
             FIN_PARA
Paso 7      FIN

```

EJERCICIO

Leer un vector de N elementos. Determinar la suma de los elementos positivos, la suma de los elementos negativos, la suma de las diferencias entre números adyacentes, además encontrar cuántos elementos iguales a cero hay en el vector. Imprimir el vector leído y todo lo calculado.

Solución

a) Análisis

Entrada: Inicializar en cero los contadores de números

ceros, la sumatoria de números positivos, la sumatoria de números negativos y la sumatoria de las diferencias entre números adyacentes.

Leer el vector así como el límite de sus elementos.

Proceso: Se utiliza una estructura **PARA** con el propósito de calcular la diferencia entre números adyacentes y la sumatoria de estos. En otra estructura **PARA** se determina la sumatoria de elementos positivos, de elementos negativos y el contador de números ceros, mediante la comparación de cada posición del vector contra cero.

Salida: Imprimir el vector original así como la suma de los elementos positivos, elementos negativos, la suma de las diferencias entre números adyacentes y el número de elementos cero que tiene el vector.

b) Variables a utilizar

N = Número de elementos del vector

Z = Nombre del vector

SUAD = Sumatoria de elementos adyacentes

SUPO = Sumatoria de elementos positivos

SUNE = Sumatoria de elementos negativos

CONCE = Contador de números ceros

DIF = Diferencia de números adyacentes

c) Algoritmo

```

Paso 1      INICIO
Paso 2      CONCE ← 0
Paso 3      SUAD ← 0
Paso 4      SUPO ← 0
Paso 5      SUNE ← 0
Paso 6      Leer N
Paso 7      PARA I DE 1 HASTA N HACER
              Leer Z(I)
              IMPRIMIR Z(I)
              FIN_PARA
Paso 8      M ← N - 1
Paso 9      PARA I DE 1 HASTA M HACER
              J ← I + 1
              DIF ← Z(I) - Z(J)
              SUAD ← SUAD + DIF
              FIN_PARA
Paso 10     PARA I DE 1 HASTA N HACER
              SI Z(I) < 0
                  ENTONCES SUNE ← SUNE + Z(I)
              SINO SI Z(I) = 0
                  ENTONCES CONCE ← CONCE + 1
                  SINO SUPO ← SUPO + Z(I)
              FINSI
              FINSI
              FIN_PARA

```

Paso 11 **IMPRIMIR SUAD, SUNE, SUPO. CONCE**

Paso 12 **FIN**

EJERCICIO

Se tienen 2 vectores X, Y, de N elementos cada uno y se desea calcular el promedio de acuerdo a la siguiente fórmula:

$$\text{PROM} = \frac{\sum_{i=1}^N X_i Y_i + \sum_{i=1}^N (X_i Y_i)^2}{\frac{1}{N} \sum_{i=1}^N X_i^2 - \left[\sum_{i=1}^N Y_i^2 \right]^2}$$

Solución

a) Análisis

Entrada: Leer los dos vectores y su límite, e inicializar las sumatorias.

Proceso: Calcular: la sumatoria de los $X_i \cdot Y_i$

la sumatoria de los $(X_i Y_i)^2$

la sumatoria de los X_i^2

la sumatoria de los Y_i^2

Salida: Imprimir el promedio

b) Variables a utilizar

N = Número de elementos de los vectores

X, Y = vectores originales

SXY = Sumatoria de $X_i * Y_i$

SX2 = Sumatoria de X_i^2

SY2 = Sumatoria de Y_i^2

SXY2 = Sumatoria de los $(X_i * Y_i)^2$

PROM = Promedio

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer N
Paso 3          PARA I DE 1 HASTA N HACER
                Leer X(I), Y(I)
                FIN_PARA
Paso 4          SXY ← 0
Paso 5          SX2 ← 0
Paso 6          SY2 ← 0
Paso 7          SXY2 ← 0
Paso 8          PARA I DE 1 HASTA N HACER
                SXY ← SXY + X(I) * Y(I)
                SXY2 ← SXY2 + (X(I) * Y(I))↑2
                SX2 ← SX2 + X(I) ↑ 2

```

$$SY2 \leftarrow SY2 + Y(I) \uparrow 2$$

FIN_PARA

Paso 9 $PROM \leftarrow (SXY + SXY2)/(1/N * SX2 - SY2 \uparrow 2)$

Paso 10 **IMPRIMIR PROM**

Paso 11 **FIN**

EJERCICIO

Leer un vector de N elementos y conformar un segundo vector que contenga todos los elementos positivos del vector original y un tercer vector que contenga todos elementos negativos del vector original. Imprimir los 3 vectores.

Solución

a) Análisis:

Entrada: Se inicializa la posición del vector de elementos negativos y del vector de elementos positivos en cero.

Leer el límite N y el vector de datos.

Proceso: Cuando se encuentra un elemento bien puede ser positivo o negativo, se debe llevar un controlador de su posición en donde va a ser almacenado para que quede consecutivo con el elemento anteriormente encontrado. De no hacerse esto, se tendría un vector de datos positivos con elementos en blanco intercalados y así mismo sucedería con

el vector de elementos negativos.

Salida: Imprimir los 3 vectores.

b) Variables a utilizar

PN = Posición de elementos negativos

PP = Posición de elementos positivos

N = Limite del vector

X = Nombre del vector original

VN = Vector resultante de elementos negativos

VP = Vector resultante de elementos positivos

c) Algoritmo

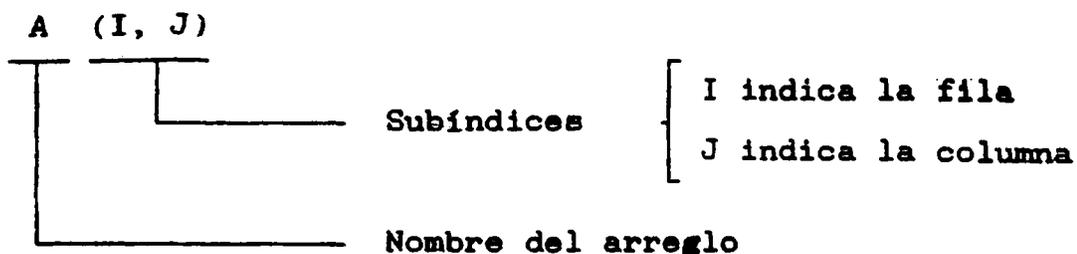
```

Paso 1          INICIO
Paso 2          PN ← 0
Paso 3          PP ← 0
Paso 4          Leer N
Paso 5          PARA J DE 1 HASTA N HACER
                  Leer X(J)
                  IMPRIMIR X(J)
                  FIN_PARA
Paso 6          PARA J DE 1 HASTA N HACER
                  SI X(J) > 0
                    ENTONCES PP ← PP + 1
                    VP(PP) ← X(J)
                  SINO SI X(J) < 0
                    ENTONCES PN ← PN + 1
                    VN(PN) ← X(J)

```

FINSI**FINSI****FIN_PARA**Paso 7 **PARA J DE 1 HASTA PP HACER****IMPRIMIR VP(J)****FIN_PARA**Paso 8 **PARA J DE 1 HASTA PN HACER****IMPRIMIR VN(J)****FIN_PARA**Paso 9 **FIN****3.5 ARREGLOS DE 2 DIMENSIONES**

También llamados arreglos bidimensionales. Está conformado por el nombre del arreglo y dos subíndices separados por coma, el primero de ellos especifica la fila y el segundo la columna. Debido a que éste tipo de formaciones es muy común, también se le conoce con el nombre de **MATRICES**.



Al igual que en los arreglos de una dimensión, es necesario diferenciar entre el valor o dato y la posición que éste ocupa dentro del arreglo bidimensional. Por ejemplo. Se

tiene la matriz A de 3 filas y 3 columnas con los siguientes datos:

	COL 1	COL 2	COL 3
FILA 1	6	9	4
FILA 2	7	2	-3
FILA 3	12	1	40

El valor 6 está en la fila 1 columna 1, del arreglo A, es decir $A(1,1) = 6$.

El valor -3 está en la fila 2 columna 3, del arreglo A, es decir $A(2,3) = -3$

En la posición $A(3,2)$, se tiene almacenado el valor 1

En la posición $A(1,3)$, se tiene almacenado el valor 4.

EJERCICIO

Leer una matriz A de N filas y M columnas. Calcular la suma de los elementos de la última fila.

Solución

a) Análisis

Entrada: Leer el número de filas N y el número de columnas

M. Inicializar la sumatoria de la última fila en

cero.

Proceso: Haciendo referencia a la matriz anteriormente graficada, se pide sumar el contenido de las posiciones $A(3,1) + A(3,2) + A(3,3) = 12+1+40 = 53$. Para lograrlo es necesario utilizar un anidamiento de estructuras PARA, una de ellas para que maneje con su índice las filas y la otra para que controle las columnas. Dentro del cuerpo del anidamiento se efectúa la comparación del índice que maneja las filas contra el límite de filas, es decir, de I contra N, cuando esta comparación sea igual, indica que el control está ubicado en la última fila y se procede a sumar sus correspondientes valores.

Salida: Imprimir la sumatoria de la última fila.

b) Variables a utilizar

SFI = Sumatoria de la fila

A = Nombre de la matriz

N = Número de filas de la matriz A

M = Número de columnas de la matriz A

c) Algoritmo

Paso 1 INICIO

Paso 2 Leer N, M

```

Paso 3          SFI ← 0
Paso 4          PARA I DE 1 HASTA N HACER
                  PARA J DE 1 HASTA M HACER
                    Leer A(I, J)
                  FIN_PARA
                FIN_PARA
Paso 5          PARA I DE 1 HASTA N HACER
                  PARA J DE 1 HASTA M HACER
                    SI I = N
                      ENTONCES SFI ← SFI + A(I, J)
                    FINSI
                  FIN_PARA
                FIN_PARA
Paso 6          IMPRIMIR SFI
Paso 7          FIN

```

d) Prueba de escritorio

Se debe recordar que cuando hay anidamiento, hasta tanto el ciclo más interno no se ejecuta totalmente, no se sale al ciclo más externo.

N	M	I	J	A(I, J)	SFI	EXPLICACION
3	3	1	1	A(1,1)=6	0	En este punto se leyó
			2	A(1,2)=9		la matriz por filas, es
			3	A(1,3)=4		decir, completamente la
		2	1	A(2,1)=7		primera fila, luego la

2	A(2,2)=2	segunda fila y así suce-	
3	A(2,3)=3	sivamente.	
3	1	A(3,1)=12	
2	A(3,2)=1		
3	A(3,3)=40		
1	1	Se compara I contra	
	2	N, es decir, I contra	
	3	3, como no son igua-	
2	1	les no se tiene en	
	2	en cuenta para la suma	
	3	de la última fila.	
3	1	$0+A(3,1)=0+12=12$	En este momento está
	2	$12+A(3,2)=12+1=13$	ubicado en la última
	3	$13+A(3,3)=13+40=53$	fila y procede a
			sumar sus elementos.

EJERCICIO PROPUESTO

Repetir la prueba de escritorio hasta tanto se entienda.

Observación

Cuando se desea procesar la matriz por filas, se deben hacer variar sus columnas, ó sea, que el índice del ciclo interno debe ser el que maneja las columnas y el índice del ciclo externo debe ser el que controla las filas.

Cuando se desee procesar la matriz por columnas, se deben hacer variar sus filas, ó sea, que el índice del ciclo

interno debe ser el que maneja las filas y el índice del ciclo externo debe ser el que controla las columnas.

EJERCICIO

Los elementos de la matriz Z se encuentran precedidos por otro registro que indica el número de filas y el número de columnas que tiene la matriz. Leer la matriz por columnas e imprimirla por filas, además calcular la suma de los elementos de la última columna.

Solución

a) Análisis

Entrada: Inicializar la sumatoria de la columna en cero.

Leer el límite de filas y el límite de columnas.

Proceso: Para leer la matriz por columnas se deben hacer variar sus filas, es decir, el ciclo interno serán las filas.

Para calcular los elementos de la última columna se deben hacer variar sus filas.

Para imprimir la matriz por filas se deben hacer variar sus columnas, ó sea, que el ciclo interno serán las columnas.

Salida: Imprimir la matriz así como la suma de los elementos de la última columna.

b) Variables a utilizar

Z = Nombre de la matriz
I = Índice de filas
J = Índice de columnas
K = Límite de filas de la matriz
L = Límite de columnas de la matriz
SCO = Sumatoria de la columna.

c) Algoritmo

```

Paso 1           INICIO
Paso 2           SCO ← 0
Paso 3           Leer K, L
Paso 4           PARA J DE 1 HASTA L HACER
                   PARA I DE 1 HASTA K HACER
                   Leer Z(I, J)
                   FIN_PARA
                   FIN_PARA
Paso 5           PARA J DE 1 HASTA L HACER
                   PARA I DE 1 HASTA K HACER
                   SI J = L
                   ←
                   ENTONCES SCO ← SCO + Z(I, J)
                   FINSI
                   FIN_PARA
                   FIN_PARA
Paso 6           PARA I DE 1 HASTA K HACER
                   PARA J DE 1 HASTA L HACER
                   IMPRIMIR Z(I, J)
  
```

```

                FIN_PARA
                FIN_PARA
Paso 7          IMPRIMIR SCO
Paso 8          FIN

```

Otra solución al ejercicio anterior es:

Algoritmo

```

Paso 1          INICIO
Paso 2          SCO ← 0
Paso 3          Leer K, L
Paso 4          PARA J DE 1 HASTA L HACER
                PARA I DE 1 HASTA K HACER
                    Leer Z(I, J)
                    SI J = L
                        ENTONCES SCO ← SCO + Z(I, J)
                FINSI
                FIN_PARA
                FIN_PARA
Paso 5          PARA I DE 1 HASTA K HACER
                PARA J DE 1 HASTA L HACER
                    IMPRIMIR Z(I, J)
                FIN_PARA
                FIN_PARA
Paso 6          IMPRIMIR SCO
Paso 7          FIN

```

EJERCICIO

Leer una matriz de N filas, M columnas y un valor Z. Averiguar en qué posiciones de la matriz se repite ese número Z y cuántas veces.

Solución**a) Análisis**

Entrada: Iniciar el contador de valores Z en cero. Leer el límite de filas, límite de columnas y la matriz.

Proceso: Como se trata de averiguar en qué posiciones de la matriz se encuentra un determinado valor, entonces hay que recorrer totalmente la matriz y no importa si se hace por filas o por columnas. Se debe comparar Z contra $A(I, J)$, si son iguales, implica que en la posición $A(I, J)$ se encuentra repetido el valor Z y se procede a incrementar el contador de repeticiones en uno.

Salida: Imprimir la matriz, el valor Z y las posiciones en que se repite.

b) Variables a utilizar

Z = Número a ser buscado

A = Nombre de la matriz

I = Índice de las filas

J = Índice de las columnas

N = Límite de filas

M = Limite de columnas

CZ = Contador de veces que se repite Z

c) Algoritmo

```

Paso 1          INICIO
Paso 2          CZ ← 0
Paso 3          Leer N, M, Z
Paso 4          PARA J DE 1 HASTA M HACER
                  PARA I DE 1 HASTA N HACER
                      Leer A(I, J)
                      IMPRIMIR A(I, J)
                  FIN_PARA
                  FIN_PARA
Paso 5          PARA I DE 1 HASTA N HACER
                  PARA J DE 1 HASTA M HACER
                      SI Z = A(I, J)
                          ENTONCES IMPRIMIR I, J
                                      CZ ← CZ + 1
                      FINSI
                  FIN_PARA
                  FIN_PARA
Paso 6          IMPRIMIR Z, CZ
Paso 7          FIN

```

3.6 PROBLEMAS RESUELTOS

EJERCICIO

Se tiene un grupo de 30 estudiantes, cada uno tomando 6 asignaturas. Calcular el promedio del grupo por cada asignatura y el promedio total. En cada registro se leen las calificaciones pertenecientes a un estudiante.

Solución**a) Análisis**

Entrada: Leer la matriz de calificaciones. Inicializar la sumatoria de las calificaciones de cada materia y la sumatoria total de calificaciones en cero.

Proceso: Se debe leer la matriz de calificaciones por filas, por lo tanto, es necesario hacer variar sus columnas.

Se generan dos vectores, uno para almacenar la sumatoria de cada asignatura y el otro para almacenar el promedio por asignatura.

El promedio de cada asignatura se calcula como la sumatoria de la asignatura dividida en 30.

El promedio total se puede calcular como la sumatoria de todas las notas dividida en 180 (30 estudiantes x 6 asignaturas), ó también sumar el vector de promedios de notas y dividirlo en 6.

Salida: Imprimir la matriz de calificaciones, el vector de promedios y el promedio total.

b) Variables a utilizar

CALIF = Nombre de la matriz de calificaciones

SUM = Sumatoria de notas de cada asignatura

PROM = Promedio de notas

STOT = Sumatoria total de notas del grupo

PTOT = Promedio total

c) Algoritmo

```

Paso 1          INICIO

Paso 2          PARA I DE 1 HASTA 30 HACER
                  PARA J DE 1 HASTA 6 HACER
                      Leer CALIF (I, J)
                      IMPRIMIR CALIF (I, J)
                  FIN_PARA
                  FIN_PARA

Paso 3          PARA J DE 1 HASTA 6 HACER
                      SUM (J) ← 0
                      PARA I DE 1 HASTA 30 HACER
                          SUM (J) ← SUM (J) + CALIF (I, J)
                      FIN_PARA
                      PROM (J) ← SUM (J) / 30
                  FIN_PARA

Paso 4          STOT ← 0

Paso 5          PARA J DE 1 HASTA 6 HACER
                      STOT ← STOT + SUM (J)
  
```

```

                FIN_PARA
Paso 6          PTOT ← STOT / 180
Paso 7          PARA J DE 1 HASTA 6 HACER
                IMPRIMIR PROM (J)
                FIN_PARA
Paso 8          IMPRIMIR PTOT
Paso 9          FIN

```

EJERCICIO

Leer una matriz de N filas y M columnas. Calcular la suma de los elementos de la triangular superior, la suma de los elementos de la triangular inferior y la suma de los elementos de la diagonal principal. Imprimir la matriz leída, así como las diagonales calculadas.

Solución

a) Análisis

Entrada: Inicializar sumatoria de la triangular inferior, sumatoria de la triangular superior, y sumatoria de la diagonal principal, en cero.

Leer el límite de filas y columnas, así como la matriz original.

Proceso: Se debe controlar si el número de filas es igual al número de columnas, para proceder a calcular las diferentes diagonales mediante comparación del índice de las filas contra el índice de las

columnas.

En una matriz cuadrática de 3 x 3, las posiciones de la triangular superior son: (1,2), (1,3), (2,3), de la triangular inferior son: (2,1), (3,1), (3,2), de la diagonal principal son: (1,1), (2,2), (3,3).

Salida: Imprimir la matriz, la triangular superior, la triangular inferior y la diagonal principal.

b) Variables a utilizar

B = Nombre de la matriz

I = Índice de las filas

J = Índice de las columnas

N = Limite de filas

M = Limite de columnas

STI = Sumatoria triangular inferior

STS = Sumatoria triangular superior

SDP = Sumatoria diagonal principal

c) Algoritmo

Paso 1	INICIO
Paso 2	STI ← 0
Paso 3	STS ← 0
Paso 4	SDP ← 0
Paso 5	Leer N, M
Paso 6	PARA I DE 1 HASTA N HACER
	PARA J DE 1 HASTA M HACER

```

    Leer B(I, J)
    IMPRIMIR B(I, J)
    FIN_PARA
FIN_PARA
Paso 7  MIENTRAS N = M HACER
        PARA I DE 1 HASTA N HACER
            PARA J DE 1 HASTA M HACER
                SI I < J
                    ENTONCES STS ← STS + B(I, J)
                SINO SI I = J
                    ENTONCES SDP ← SDP+B(I, J)
                SINO STI ← STI + B(I, J)
            FINSI
        FINSI
    FIN_PARA
    FIN_PARA
    FIN_MIENTRAS
Paso 8  IMPRIMIR SDP, STI, STS
Paso 9  FIN

```

EJERCICIO

Leer 2 matrices A y B. Generar una tercera matriz que contenga la suma de éstas. Imprimir las 3 matrices.

Solución

a) Análisis

Entrada: Leer el límite de filas y columnas de la matriz A y de la matriz B. Luego leer las matrices A y B.

Proceso: Se debe controlar que el número de filas de la matriz A sea igual al número de filas de la matriz B y el número de columnas de la matriz A sea igual al número de columnas de la matriz B; en caso de que ésta condición no se cumpla, se imprime un mensaje indicando la inconsistencia presentada.

Salida: Imprimir las matrices originales y la matriz suma.

b) Variables a utilizar

NFA = Número de filas de la matriz A

NCA = Número de columnas de la matriz A

NFB = Número de filas de la matriz B

NCB = Número de columnas de la matriz B

A = Nombre de la primera matriz original

B = Nombre de la segunda matriz original

S = Nombre de la matriz suma

I = Índice de filas

J = Índice de columnas

c) Algoritmo

Paso 1	INICIO
Paso 2	Leer NFA, NCA, NFB, NCB
Paso 3	PARA I DE 1 HASTA NFA HACER

PARA J DE 1 HASTA NCA HACER

Leer A(I, J)

IMPRIMIR A(I, J)

FIN_PARA

FIN_PARA

Paso 4

PARA I DE 1 HASTA NFB HACER

PARA J DE 1 HASTA NCB HACER

Leer B(I, J)

IMPRIMIR B(I, J)

FIN_PARA

FIN_PARA

Paso 5

SI (NFA = NFB) Y (NCA = NCB)

ENTONCES PARA I DE 1 HASTA NFA HACER

PARA J DE 1 HASTA NCA HACER

$S(I, J) \leftarrow A(I, J) + B(I, J)$

FIN_PARA

FIN_PARA

SINO IMPRIMIR "las matrices no se
pueden sumar"

FINSI

Paso 6

PARA I DE 1 HASTA NFA HACER

PARA J DE 1 HASTA NCA HACER

IMPRIMIR S(I, J)

FIN_PARA

FIN_PARA

Paso 7

FIN

EJERCICIO

Leer 2 matrices A y B. Generar una tercera matriz que contenga el producto de éstas. Imprimir las 3 matrices.

Solución**a) Análisis**

Entrada: Leer el límite de filas y columnas de la matriz A y de la matriz B. Leer las matrices A y B.

Proceso: Para poder efectuar el producto, se debe controlar que el número de columnas de la matriz A sea igual al número de filas de la matriz B; en caso de no cumplirse ésta condición se imprime un mensaje indicando la inconsistencia presentada.

Salida: Imprimir las matrices originales y la matriz producto.

b) Variables a utilizar

NFA = Número de filas de la matriz A

NCA = Número de columnas de la matriz A

NFB = Número de filas de la matriz B

NCB = Número de columnas de la matriz B

A, B = Nombre de las matrices originales

P = Nombre de la matriz producto

I = Índice de las filas

J = Índice de las columnas

c) Algoritmo

```

Paso 1      INICIO
Paso 2      Leer NFA, NCA, NFB, NCB
Paso 3      PARA I DE 1 HASTA NFA HACER
              PARA J DE 1 HASTA NCA HACER
                Leer A(I, J)
                IMPRIMIR A(I, J)
              FIN_PARA
            FIN_PARA
Paso 4      PARA I DE 1 HASTA NFB HACER
              PARA J DE 1 HASTA NCB HACER
                Leer B(I, J)
                IMPRIMIR B(I, J)
              FIN_PARA
            FIN_PARA
Paso 5      SI (NCA = NFB)
              ENTONCES PARA I DE 1 HASTA NFA HACER
                PARA J DE 1 HASTA NCB HACER
                  P(I, J) ← 0
                  PARA K DE 1 HASTA NCA HACER
                    P(I, J) ← P(I, J) + A(I, K) * B(K, J)
                  FIN_PARA
                FIN_PARA
              FIN_PARA
            SINO IMPRIMIR "las matrices no se pue-
              den multiplicar"

```

```

FINSI
Paso 6   PARA I DE 1 HASTA NFB HACER
          PARA J DE 1 HASTA NCA HACER
          IMPRIMIR P(I,J)
          FIN_PARA
          FIN_PARA
Paso 7   FIN

```

EJERCICIO

Se tiene una matriz cuadrática de N filas y N columnas. Encontrar el elemento mayor de cada fila y de cada columna, e imprimirlos así como la matriz leída.

Solución**a) Análisis**

Entrada: Leer el límite de filas y columnas de la matriz y luego leer la matriz original.

Proceso: Se asume que la primera posición de la matriz es la mayor de la primera fila y se compara contra cada uno de los demás elementos de esa misma fila, en caso de encontrar un elemento mayor al asumido inicialmente entonces es reemplazado. Al finalizar el análisis de la fila, el elemento mayor es almacenado en un vector de elementos mayores por fila. Igual procedimiento se hace con cada una de

las demás filas y columnas de la matriz.

Salida: Imprimir la matriz original y los vectores de elementos mayores de las filas y de elementos mayores de las columnas.

b) Variables a utilizar

N = Número de filas y columnas de la matriz

B = Nombre de la matriz original

FIMA = Nombre del vector de filas mayores

COMA = Nombre del vector de columnas mayores

I = Índice de filas

J = Índice de columnas

c) Algoritmo

```

Paso 1          INICIO
Paso 2          Leer N
Paso 3          PARA I DE 1 HASTA N HACER
                  PARA J DE 1 HASTA N HACER
                      Leer B(I, J)
                      IMPRIMIR B(I, J)
                  FIN_PARA
                FIN_PARA
Paso 4          PARA I DE 1 HASTA N HACER
                  FIMA(I) ← B(I, 1)
                  PARA J DE 1 HASTA N HACER
                      SI FIMA(I) < B(I, J)

```

ENTONCES FIMA(I) ← B(I, J)

FINSI

FIN_PARA

FIN_PARA

Paso 5

PARA J DE 1 HASTA N HACER

COMA(J) ← B(1, J)

PARA I DE 1 HASTA N HACER

SI COMA(J) < B(I, J)

ENTONCES COMA(J) ← B(I, J)

FINSI

FIN_PARA

FIN_PARA

Paso 6

PARA I DE 1 HASTA N HACER

IMPRIMIR FIMA(I), COMA(I)

FIN_PARA

Paso 7

FIN

EJERCICIO

Se tiene una matriz de N filas y M columnas y se desea calcular el promedio de cada fila así como el promedio de cada columna. Imprimir la matriz leída y los promedios calculados.

Solución

a) Análisis

Entrada: Leer el límite de filas, el límite de columnas y la matriz.

Proceso: Se calculan los promedios y almacenan en dos vectores, uno para los promedios de las filas y el otro para los promedios de las columnas.

Se inicializan la sumatoria de las filas y sumatoria de las columnas en cero por cada ciclo ejecutado.

Salida: Imprimir la matriz, el vector de promedios de filas y el vector de promedios de columnas.

b) Variables a utilizar

\wedge B = Nombre de la matriz

I = Índice de las filas

J = Índice de las columnas

N = Límite de las filas

M = Límite de las columnas

PROFI = Vector promedio de las filas

PROCO = Vector promedio de las columnas.

c) Algoritmo

Paso 1 **INICIO**

Paso 2 **Leer N, M**

Paso 3 **PARA I DE 1 HASTA N HACER**

PARA J DE 1 HASTA M HACER

Leer A(I, J)

IMPRIMIR A(I, J)

FIN_PARA

FIN_PARA

Paso 4

PARA I DE 1 HASTA N HACER

PROFI(I) ← 0

PARA J DE 1 HASTA M HACER

PROFI(I) ← PROFI(I)+A(I, J)/M

FIN_PARA

IMPRIMIR PROFI(I)

FIN_PARA

Paso 5

PARA J DE 1 HASTA M HACER

PROCO(J) ← 0

PARA I DE 1 HASTA N HACER

PROCO(J) ← PROCO(J)+A(I, J)/N

FIN_PARA

IMPRIMIR PROCO(J)

FIN_PARA

Paso 6

FIN

EJERCICIO

Leer una matriz de N filas y M columnas. Calcular la suma de los elementos de la 1ª fila, la suma de los elementos de la 1ª columna, la suma de los elementos de la diagonal principal y la suma de los elementos de la diagonal secundaria. Imprimir la matriz leída y todo lo calculado.

Solución**a) Análisis**

Entrada: Leer el límite de filas, límite de columnas y la matriz.

Inicializar en cero las sumatorias de la primera fila, la primera columna, la diagonal principal y la diagonal secundaria.

Proceso: Controlar si el número de filas es igual al número de columnas para poder hallar la diagonal principal y diagonal secundaria, si no lo son, se debe calcular la sumatoria de los elementos de la primera fila y la suma de los elementos de la primera columna.

Cuando la matriz es de 3 filas por 3 columnas, la diagonal secundaria se calcula sumando los elementos de las posiciones (1,3), (2,2), (3,1).

La diagonal principal se calcula sumando los elementos de las posiciones (1,1), (2,2), (3,3).

Salida: SF = Sumatoria de la primera fila

SC = Sumatoria de la primera columna

SDP = Sumatoria de la diagonal principal

SDS = Sumatoria de la diagonal secundaria

N = Límite de filas de la matriz

M = Límite de columnas de la matriz

MAT = Nombre de la matriz

I = Índice de las filas

J = Índice de las columnas

c) Algoritmo

```

Paso 1          INICIO
Paso 2          SF ← 0
Paso 3          SC ← 0
Paso 4          SDP ← 0
Paso 5          SDS ← 0
Paso 6          Leer N, M
Paso 7          PARA I DE 1 HASTA N HACER
                  PARA J DE 1 HASTA M HACER
                    Leer MAT(I, J)
                    IMPRIMIR MAT(I, J)
                  FIN_PARA
                FIN_PARA
Paso 8          PARA I DE 1 HASTA N HACER
                  PARA J DE 1 HASTA M HACER
                    SI J = 1
                      ENTONCES SC ← SC + MAT(I, J)
                    FINSI
                  SI I = 1
                      ENTONCES SF ← SF + MAT(I, J)
                  FINSI
                FIN_PARA
                FIN_PARA
Paso 9          IMPRIMIR SC, SF

```

Paso 10

SI $N = M$

ENTONCES PARA I DE 1 HASTA N HACER

PARA J DE 1 HASTA M HACER

SI $I = J$ ENTONCES $SDP \leftarrow SDP + MAT(I, J)$

FINSI

SI $J = M$ ENTONCES $SDS \leftarrow SDS + MAT(I, J)$ $M \leftarrow M - 1$

FINSI

FIN_PARA

FIN_PARA

IMPRIMIR SDP, SDS

SINO IMPRIMIR "la matriz no es cuadrática".

FINSI

Paso 11

FIN

3.7 PROBLEMAS PROPUESTOS

1-

INICIO

PARA I DE 1 HASTA 3 HACER

PARA J DE 1 HASTA 6 HACER

 $K(I, J) \leftarrow I$

FIN_PARA

FIN_PARA

FIN

Qué valores tendrá la matriz K al culminar el ciclo ?

2- Encontrar el elemento mayor y el menor de una matriz. Imprimirlos.

3- Calcular la raíz cuadrada de la suma de los cuadrados de los primeros cincuenta elementos pares de una lista numérica. Dicho de otra forma. Calcular:

$$Z = (A(2)^2 + A(4)^2 + A(6)^2 + \dots + A(100)^2)^{1/2}$$

4- Leer una matriz cuadrada e imprimir la matriz leída y la matriz identidad.

La matriz identidad es una matriz cuadrática en donde los elementos de la diagonal principal son iguales a uno y los demás elementos son iguales a cero.

5- Leer una matriz de $M \times M$. Hallar e imprimir la transpuesta de la matriz.

6- Leer una matriz R de N filas y M columnas. Calcular las sumas de cada fila y almacenarlas en un vector, calcular las sumas de cada columna y almacenarlas en otro vector. Imprimir la matriz leída y los vectores hallados. Calcular además la suma total de cada uno de los vectores e imprimirlas.

CAPITULO IV

INTRODUCCION AL DISEÑO DESCENDENTE

4.1 INTRODUCCION

Una de las ideas esenciales en la programación de computadores se refiere a la división de problemas grandes y complejos, en subproblemas más simples, comprensibles y fáciles de implementar, obviando de ésta forma inconvenientes como los siguientes:

- Repetición de un conjunto de instrucciones o pasos a través del algoritmo.
- La inclusión de pequeños cambios a un algoritmo, se refleja en profundas reformas al mismo.
- La puesta en marcha del algoritmo se vuelve compleja.

Los anteriores problemas se evitan con la utilización de la Programación Modular.

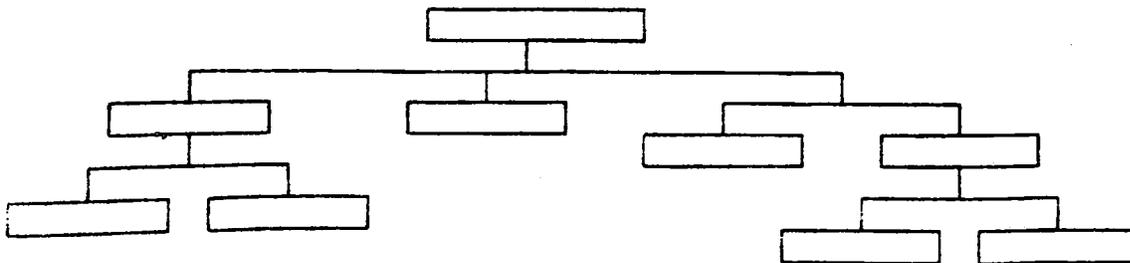
4.2 PROGRAMACION MODULAR

Conocida en ocasiones como Divide y Vencerás, consiste en fraccionar un problema complejo en varios subproblemas, para que la solución se vuelva más simple. A su vez, un subproblema se puede dividir en otros subproblemas y así sucesivamente. A ésta metodología también se le llama

DISEÑO DESCENDENTE ó TOP_DOWN, porque parte del problema general al diseño de soluciones específicas para cada una de las divisiones, hasta obtener una solución efectiva del problema principal. En esta forma será posible resolver y probar cada subproblema por separado, permitiendo inclusive una distribución de tareas entre programadores acortando notablemente el tiempo demandado para la solución.

Cuando se plantea adecuadamente el análisis descendente, obliga a examinar todos los aspectos del problema antes de continuar con el nivel inferior siguiente, siendo el resultado final un conjunto de algoritmos que se pueden implementar usando las estructuras estudiadas en los capítulos anteriores.

Las relaciones de cada módulo del Diseño Descendente pueden ser representadas gráficamente en un diagrama de estructura similar a un organigrama, mediante bloques o rectángulos que señalan una actividad asociada con un nivel particular.



Las normas de esta metodología son:

- Un módulo se ejecuta sólo cuando el control le llega del bloque precedente. Cuando el bloque termina su ejecución, el control regresa al módulo que le antecede jerárquicamente.
- El control de la información se enruta hacia abajo, los resultados se enrutan hacia arriba.
- Cada módulo debe resolver un problema específico.

4.3 CLASIFICACION DE LOS MODULOS

Los módulos se pueden clasificar en Funciones y Procedimientos.

4.3.1 Funciones

Cuando se llama a una función, el control pasa a las instrucciones que la definen y una vez que la función ha sido evaluada, el control regresa con el correspondiente resultado al punto donde fué llamado en el algoritmo principal.

Su forma general es:

FUNCION < NOMBRE_FUNCION > (ARGUMENTOS)

ACCION 1

ACCION 2

.

.

ACCION N

FIN NOMBRE_FUNCION

Donde: El nombre de la función se debe caracterizar por lo que realiza.

Los argumentos son nombres de variables, de funciones o de procedimientos y sólo son utilizadas dentro del cuerpo de la función.

El conjunto de argumentos se llaman **Ficticios** o **Temporales** cuando son declarados en la función y se llaman **Actuales** cuando son declarados en el algoritmo principal.

Una función es evocada de la siguiente forma:

NOMBRE_FUNCION (Lista de argumentos actuales)

Los usuarios de los lenguajes de programación tienen a su disposición cierta cantidad de funciones definidas por el sistema, también llamadas funciones **Internas**, como son las funciones trigonométricas y matemáticas, por ejemplo SIN (Seno), COS (Coseno), TAN (Tangente), SQRT (Raíz cuadrada), etc. Las funciones **Internas** se pueden usar directamente en las expresiones, mediante el llamado de su nombre con los argumentos encerrados entre paréntesis.

EJERCICIO

Elaborar una tabla de Senos y Cosenos para valores de X que oscilan entre 0 y 360 grados y que sufre incrementos de 5.

Solución**a) Análisis**

Entrada: No requiere de lectura de datos ya que éstos se generan automáticamente.

Proceso: No se hace necesario definir las funciones de Seno y Coseno puesto que la mayoría de los sistemas las tienen definidas, por lo tanto se hará uso de ellas en forma directa

Salida: Imprimir el valor de X, el Seno y Coseno respectivamente.

b) Variables a utilizar

X = Valor a ser tratado

SIN = Función Seno

COS = Función Coseno

SENO = Seno de X

COSENO = Coseno de X

c) Algoritmo

```

Paso 1          INICIO
Paso 2          PARA X DE 0 HASTA 360.5 HACER
                  SENO ← SIN (X)
                  COSENO ← COS (X)
                  IMPRIMIR X. SENO. COSENO
                  FIN_PARA
Paso 3          FIN

```

El argumento en éste caso es la variable X, la cual es objeto del cálculo del Seno y Coseno en forma directa.

Cuando las funciones **Internas** no realizan una operación determinada o cálculo en particular, se hace necesario que sean definidas por el usuario y son llamadas funciones **Externas**. Tanto las funciones **Internas** como las **Externas**, retornan un sólo valor bajo el mismo nombre como fué definida y el cual puede ser cualquier tipo elemental de dato.

Las funciones **Externas** son definidas en algoritmos independientes, las cuales realizan una labor específica y retornan el resultado al mismo lugar del algoritmo principal donde se hizo la llamada, una vez hayan sido ejecutadas.

EJERCICIO

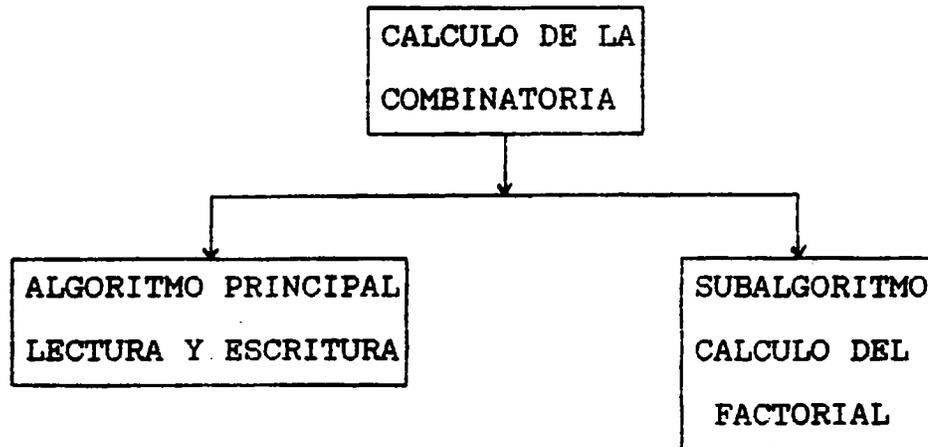
Calcular la combinatoria de N en M según la siguiente fórmula:

$$\left[\begin{array}{c} N \\ M \end{array} \right] = \frac{N!}{M! (N - M)!}$$

Solución

a) Análisis

Estructura del algoritmo



Entrada: Leer N y M

Proceso: Por restricción del problema se controla que $N > M$.

Se hace un subalgoritmo general que calcula el factorial de un valor y se elabora el algoritmo principal del que se hacen tantos llamados al subalgoritmo como se requieran, por ejemplo para calcular el factorial del numerador (N), factorial del denominador (M) y el factorial de la diferencia (N - M).

El argumento del subalgoritmo es un argumento temporal, el cual es reemplazado por el argumento actual del algoritmo principal para posteriormente calcular la combinatoria.

Salida: Imprimir N, M y la combinatoria

b) Variables a utilizar

N, M, K = Valores a ser factorizados (Argumentos Actuales)

FACN = Factorial de N

FACM = Factorial de M

FACK = Factorial de la diferencia (N-M)

FACT = Factorial General

COMBI = Combinatoria

I = Variable controladora de ciclos

J = Argumento temporal

c) Algoritmo principal

```

Paso 1          INICIO
Paso 2          Leer N, M
Paso 3          MIENTRAS (N > M) Y (M >= 0) HACER
                FACN ← FACTORIAL(N)
                FACM ← FACTORIAL(M)
                K ← N - M
                FACK ← FACTORIAL(K)
                COMBI ← FACN / (FACM * FACK)
                IMPRIMIR N, M, COMBI
                FIN_MIENTRAS
Paso 4          FIN
  
```

Subalgoritmo

```

                    FUNCION FACTORIAL(J)

Paso 1             INICIO
Paso 2             FACT ← 1
Paso 3             I ← 1
Paso 4             REPETIR
                    FACT ← FACT * I
                    I ← I + 1
                    HASTA_QUE I > J
Paso 5             FACTORIAL ← FACT
Paso 6             FIN

```

En la primera llamada se evoca desde el algoritmo principal al subalgoritmo y el argumento actual (N) es reemplazado por el argumento temporal (J), se evalúa el factorial y el resultado, almacenado bajo el nombre de la función FACTORIAL, es retornado al algoritmo principal al mismo lugar donde se hizo la llamada para continuar con la ejecución de éste.

EJERCICIO

Mediante la utilización de una función externa calcular la media armónica H, de la siguiente manera:

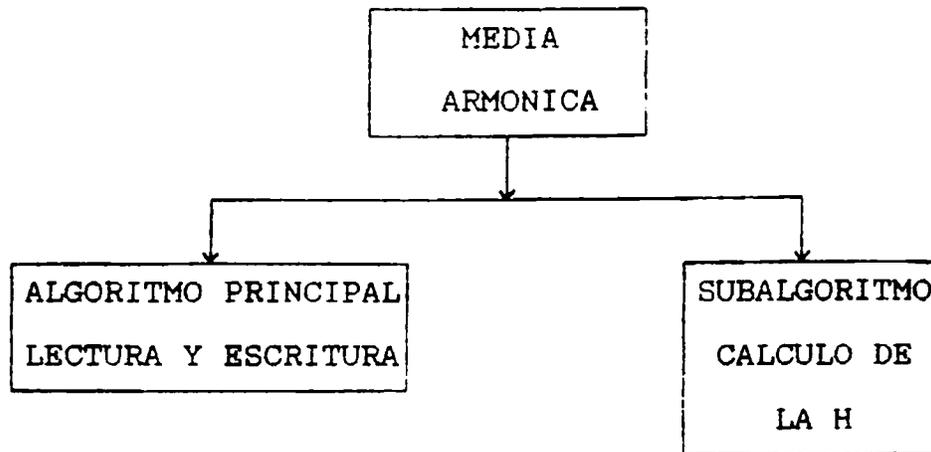
$$H = \frac{N}{\sum_{i=1}^N 1/X_i} \quad \text{Siendo} \quad \sum_{i=1}^N 1/X_i = 1/X_1 + 1/X_2 + \dots + 1/X_N$$

Para un vector Z de M elementos.

Solución

a) Análisis

Estructura del algoritmo



Entrada: Leer el vector a ser analizado y su límite.

Inicializar la sumatoria de inversos y el contador de elementos sumados en cero.

Proceso: Chequear que el elemento a ser sumado sea diferente de cero, en este caso, se incrementa el contador de términos sumados en la unidad. Si el elemento es igual a cero no lo tiene en cuenta porque su inverso daría como resultado un indeterminado y toma el siguiente elemento para hacerle el mismo análisis. Por último se divide la sumatoria de inversos entre el contador de términos sumados. Todo este proceso se hace en un subalgoritmo.

El algoritmo principal se encarga de leer el vector Z y hacer el llamado al subalgoritmo, mediante la entrega del nombre del vector como argumento, para que proceda a realizar el cálculo respectivo.

Salida: Imprimir el vector leído y la media armónica.

b) Variables a utilizar

CE = Contador de elementos

Z = Nombre del vector

M = Límite de elementos del vector

Z,M = Argumentos Actuales

SUMA = Acumulador de inversos

X, N = Argumentos temporales

H = Media armónica

c) Algoritmo Principal

```

Paso 1          INICIO
Paso 2          Leer M
Paso 3          PARA I DE 1 HASTA M HACER
                  Leer Z(I)
                  IMPRIMIR Z(I)
                  FIN_PARA
Paso 4          H ← ARMONICA(Z, M)
Paso 5          IMPRIMIR "La media armónica del vector Z
                  es". H
Paso 6          FIN

```

Subalgoritmo

```

                FUNCION ARMONICA(X, N)

Paso 1          INICIO
Paso 2          SUMA ← 0
Paso 3          CE ← 0
Paso 4          PARA I DE 1 HASTA N HACER
                SI X(I) <> 0
                ENTONCES SUMA ← SUMA + 1/X(I)
                CE ← CE + 1
                FINSI
                FIN_PARA
Paso 5          ARMONICA ← CE / SUMA
Paso 6          FIN

```

Observación

Los argumentos del algoritmo principal y del subalgoritmo no necesariamente deben coincidir en el nombre, puesto que los argumentos **Actuales** son reemplazados en los argumentos **Temporales** y una vez hayan sido evaluados, son reemplazados de nuevo en los argumentos **Actuales** para posterior utilización en cálculos e impresión. Es indispensable que los argumentos **Temporales** correspondan en número, orden y tipo con los argumentos **Actuales**.

4.3.2 Procedimientos

No en todos los algoritmos se hace necesario la utilización de Funciones ó Procedimientos, de hecho se pueden presentar algoritmos compuestos por cientos de pasos o actividades que jamás utilicen un Procedimiento. Sin embargo, esta metodología presenta entre otros, los siguientes problemas: la solución se vuelve más extensa de lo necesario; requiere mayor cantidad de memoria del computador; a mayor número de instrucciones es mayor la probabilidad de cometer errores en la digitación; es más dificultoso su entendimiento; al elaborar una prueba de escritorio es más complejo su seguimiento y se hace más difícil la detección y corrección de errores. Por lo anterior se puede deducir que el Procedimiento es más generalista y recomendable.

La estructura del Procedimiento básicamente es la misma en la mayoría de los lenguajes de programación, aunque en unos se llama Subrutina ó Subprograma, en otros se llama Procedimiento o Módulo.

Los Procedimientos se declaran en forma similar a las Funciones, con pequeñas variaciones en la forma como se le referencia. A diferencia de las Funciones: a) los Procedimientos devuelven 1 ó más valores al algoritmo que lo llama. b) El nombre del Procedimiento no está asociado a

los resultados que produce.

Su forma general es:

PROCEDIMIENTO < NOMBRE_PROCEDIMIENTO > (ARGUMENTOS)

ACCION 1

ACCION 2

.

.

ACCION N

FIN NOMBRE_PROCEDIMIENTO

El procedimiento es evocado de la siguiente forma:

LLAMAR NOMBRE_PROCEDIMIENTO (lista de argumentos actuales).

OBSERVACION: Algunos compiladores exigen que los procedimientos o subprogramas sean definidos ántes que el programa principal, para otros es indiferente. En este estudio se define primero el algoritmo principal y a partir de él se hace el llamado a los diferentes procedimientos en la medida en que se requieran.

EJERCICIO

Se tiene un conjunto de 100 registros con la siguiente información por registro: Altura y Base del triángulo.

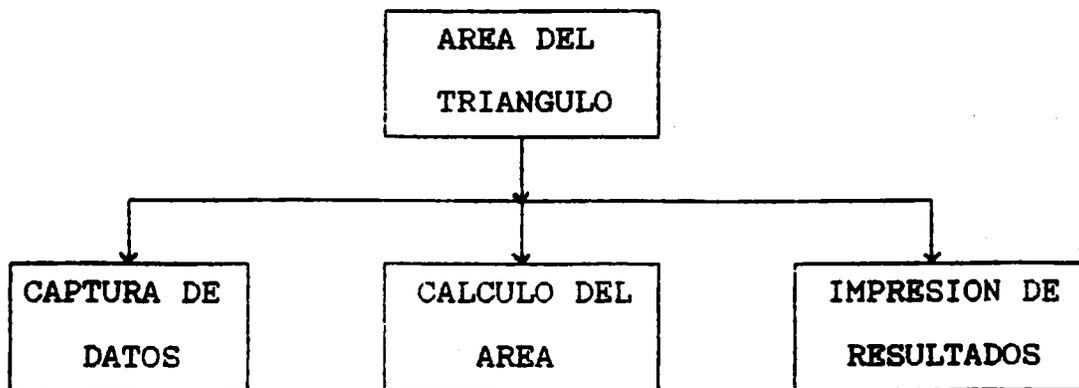
Calcular el área del triángulo según la siguiente fórmula:

$$\text{AREA} = \frac{\text{BASE} \times \text{ALTURA}}{2}$$

Solución

a) Análisis

Estructura del algoritmo



Entrada: Leer la base y la altura. Estos datos se solicitan por pantalla.

Proceso: Se elaboran 3 procedimientos, uno para la captura de datos, otro para el cálculo del área del triángulo y el último para producir los resultados. A partir del algoritmo principal se hace el llamado a cada procedimiento.

Salida: Imprimir base, altura y área del triángulo

PROCEDIMIENTO IMPRIMIR_RESULTADOS (BASE, ALTURA, AREA)

- Paso 1 **INICIO**
- Paso 2 **IMPRIMIR "La base del triángulo es", BASE**
- Paso 3 **IMPRIMIR "La altura del triángulo es", ALTURA**
- Paso 4 **IMPRIMIR "El área del triángulo es", AREA**
- Paso 5 **FIN IMPRIMIR_RESULTADOS**

En el paso 2 del procedimiento DATOS se escribe un mensaje por pantalla en donde se solicita teclear la altura del triángulo, diferente del paso 2 del procedimiento IMPRIMIR_RESULTADOS, porque allí se refiere a una salida por impresora.

Las instrucciones empleadas en el algoritmo principal consisten únicamente de los nombres de los procedimientos y sus argumentos. Después de ejecutado un procedimiento, el programa principal continúa con la siguiente instrucción y así sucesivamente hasta su terminación.

Observación

Los argumentos están conformados por las variables que requiere el procedimiento para poder efectuar los diferentes procesos y por aquellas variables que se generan dentro de él y que serán de posterior utilización, bien puede ser en otros cálculos, o para su correspondiente impresión.

EJERCICIO

Se tiene un lote de N registros con la siguiente información: código, nombre, salario bruto y total de días trabajados. Se desea calcular el interés sobre la cesantía, como el total pagado por intereses de la siguiente forma :

$$\text{Interés a la cesantía} = \frac{\text{Cesantía} \times 12\% \text{ anual} \times \text{total días trabajados}}{360 \text{ días}}$$

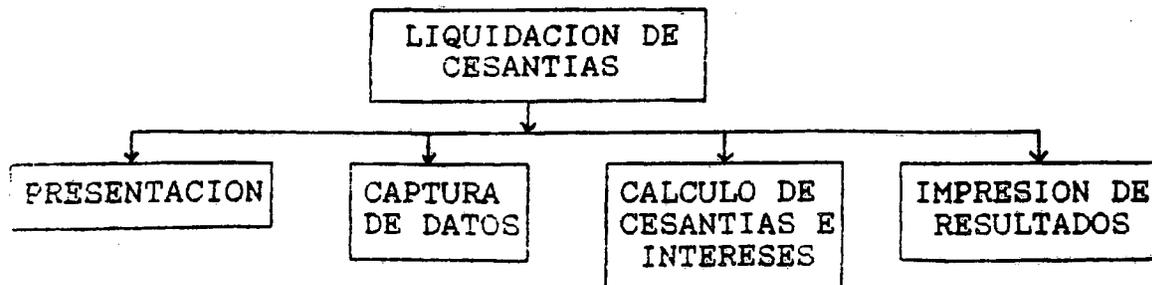
$$\text{Cesantía} = \frac{\text{Salario bruto} \times \text{total días trabajados}}{360 \text{ días}}$$

Nota.

Si el total de días trabajados es inferior a 30, no se tiene derecho a cesantías.

Solución

Estructura del algoritmo



Entrada: Leer el límite de datos a procesar

Inicializar el total de intereses en cero

Leer código, nombre, salario bruto y total de días trabajados.

Proceso: Se elaboran 4 procedimientos para:

- Presentación del trabajo, el cual no contiene argumentos por que no requiere de variables de entrada como de salida y tiene solo fines de documentación.
- Captura de datos. Se solicitan los datos por pantalla.
- Cálculo de cesantías e intereses a las cesantías.
- Impresión de resultados. La salida va acompañada de los títulos correspondientes.

Salida: Imprimir código, nombre, salario bruto, total de días trabajados, cesantías e intereses a las cesantías y el total de intereses.

b) Variables a utilizar

N = Límite de datos a procesar

COD = Código

NOM = Nombre

SB = Salario bruto

TDT = Total de días trabajados

CES = Cesantías

INT = Intereses a las cesantías

TINT = Total de intereses.

c) Algoritmo Principal

```

Paso 1          INICIO
Paso 2          LLAMAR PRESENTACION
Paso 3          Leer N
Paso 4          TINT ← 0
Paso 5          PARA I DE 1 HASTA N HACER
                  LLAMAR CAPTURA_DATOS(COD, NOM, SB, TDT)
                  LLAMAR CESANTIAS(SB, TDT, CES, INT)
                  TINT ← TINT + INT
                  LLAMAR RESULTADOS(COD,  NOM.  SB, TDT,
                                      CES, INT)
                  FIN_PARA
Paso 6          IMPRIMIR "El total de intereses es", TINT
Paso 7          FIN

```

PROCEDIMIENTO PRESENTACION

```

Paso 1          INICIO
Paso 2          IMPRIMIR "UNIVERSIDAD NACIONAL DE COLOMBIA"
Paso 3          IMPRIMIR "SECCIONAL MANIZALES"
Paso 4          IMPRIMIR "CALCULO DE CESANTIAS"
Paso 5          IMPRIMIR "MANIZALES, ENERO DE 1990"
Paso 6          FIN_PRESENTACION

```

PROCEDIMIENTO CAPTURA_DATOS (COD, NOM, SB, TDT)

Paso 1 **INICIO**
Paso 2 **ESCRIBIR** "Teclee el código del trabajador"
Paso 3 **Leer** COD
Paso 4 **ESCRIBIR**" Teclee el nombre del trabajador"
Paso 5 **Leer** NOM
Paso 6 **ESCRIBIR** "Teclee salario bruto"
Paso 7 **Leer** SB
Paso 8 **ESCRIBIR** "Teclee total de días trabajados"
Paso 9 **Leer** TDT
Paso 10 **FIN CAPTURA_DATOS**

PROCEDIMIENTO CESANTIAS (SB, TDT, CES, INT)

Paso 1 **INICIO**
Paso 2 **SI** TDT >= 30
 ENTONCES CES ← SB * TDT / 360
 INT ← CES * 0.12 * TDT / 360
 SINO CES ← 0
 INT ← 0

FINSI

Paso 3 **FIN_CESANTIAS**

PROCEDIMIENTO RESULTADOS (COD, NOM, SB, TDT, CES, INT)

Paso 1 **INICIO**
Paso 2 **IMPRIMIR** "Código del trabajador", COD
Paso 3 **IMPRIMIR** "Nombre del trabajador", NOM
Paso 4 **IMPRIMIR** "Salario bruto", SB

- Paso 5 **IMPRIMIR "Total días trabajados", TDT**
- Paso 6 **IMPRIMIR "Cesantías", CES**
- Paso 7 **IMPRIMIR "Intereses a las cesantías", INT**
- Paso 8 **FIN RESULTADOS**

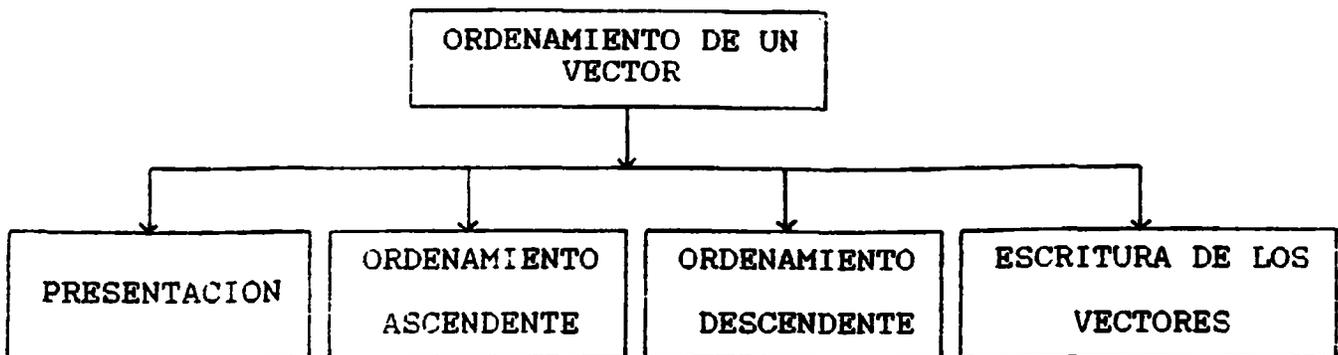
EJERCICIO

Leer un vector K de N elementos y ordenarlo en forma ascendente y descendente. Imprimir el vector original y los vectores ordenados.

Solución

a) Análisis

Estructura del Algoritmo



Entrada : Leer el límite N y el vector K a ser ordenado.

Proceso : Se utilizan cuatro subalgoritmos para realizar lo siguiente :

- Presentación del ejercicio
- Ordenamiento ascendente
- Ordenamiento descendente
- Impresión de los vectores

Salida : Imprimir el vector original y los vectores ordenados.

b) Variables a Utilizar

K = Nombre del vector

N = Número de elementos del vector

T = Variable temporal

J = Posición del vector. a su vez, índice de la estructura PARA.

c) Algoritmo Principal

Paso 1	INICIO
Paso 2	LLAMAR PORTADA
Paso 3	LEER N
Paso 4	PARA J DE 1 HASTA N HACER LEER K(J) FIN PARA
Paso 5	IMPRIMIR EL VECTOR ORIGINAL ES".
Paso 6	LLAMAR SALIDA(N,K)
Paso 7	LLAMAR ASCENDENTE(N,K)
Paso 8	IMPRIMIR EL VECTOR ORDENADO ASCENDENTEMENTE ES'
Paso 9	LLAMAR SALIDA(N,K)
Paso 10	LLAMAR DESCENDENTE(N,K)
Paso 11	IMPRIMIR "EL VECTOR ORDENADO DESCENDENTEMENTE ES
Paso 12	LLAMAR SALIDA(N,K)
Paso 13	FIN

PROCEDIMIENTO PORTADA

Paso 1 **INICIO**

Paso 2 **IMPRIMIR "ORDENAMIENTO DE UN VECTOR"**

Paso 3 **IMPRIMIR "MEDIANTE LA UTILIZACION DE"**

Paso 4 **IMPRIMIR "PROCEDIMIENTOS"**

Paso 5 **FIN PORTADA**

PROCEDIMIENTO SALIDA(N,K)

Paso 1 **INICIO**

Paso 2 **PARA J DE 1 HASTA N HACER**
 IMPRIMIR K(J)
 FIN_PARA

Paso 3 **FIN SALIDA**

PROCEDIMIENTO ASCENDENTE(N,K)

Paso 1 **INICIO**

Paso 2 **J ← 1**

Paso 3 **REPETIR**
 SI K(J) > K(J+1)
 ENTONCES T ← K(J)
 K(J) ← K(J+1)
 K(J+1) ← T
 J ← 1
 SINO J ← J+1
 FINSI
 HASTA_QUE J=N

Paso 4 **FIN ASCENDENTE**

PROCEDIMIENTO DESCENDENTE(N, K)

Paso 1 **INICIO**

Paso 2 $J \leftarrow 1$

Paso 3 **REPETIR**

SI $K(J) < K(J+1)$

ENTONCES $T \leftarrow K(J)$

$K(J) \leftarrow K(J+1)$

$K(J+1) \leftarrow T$

$J \leftarrow 1$

SINO $J \leftarrow J+1$

FINSI

HASTA QUE $J=N$

Paso 4 **FIN DESCENDENTE**

EJERCICIO

Una compañía de sistemas efectuó un estudio relacionando los puntos de una prueba de aptitud con la productividad en computación del nuevo personal.

EMPLEADO	APTITUD (X)	PRODUCTIVIDAD (Y)
1	7	35
2	13	53
3	17	75
4	19	81
5	21	85
6	23	93
7	27	97

La compañía desea obtener la ecuación de la recta que

pueda usarse para predecir la productividad de futuros trabajadores, así como la media aritmética y desviación típica de las variables X, Y, de acuerdo con las siguientes fórmulas:

$$\bar{X} = \frac{\sum X_i}{N} \quad D_x = \sqrt{\frac{\sum X^2}{N} - \bar{X}^2} \quad D_y = \sqrt{\frac{\sum Y^2}{N} - \bar{Y}^2}$$

$$A = \frac{\sum XY + N * \bar{X} * \bar{Y}}{\sum X^2 - N * \bar{X}^2} \quad b = \bar{Y} - A * \bar{X}$$

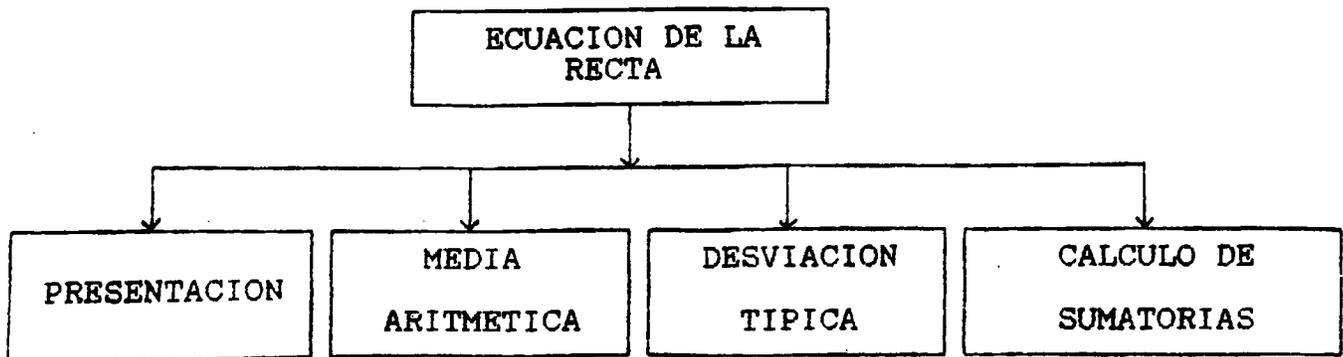
Ecuación de la recta: $Y = AX + b$

Imprimir la información leída como la calculada, mediante el empleo de procedimientos.

Solucion

a) Análisis

Estructura del algoritmo



Entrada: Leer el límite de datos

Leer la identificación de cada empleado

Leer la matriz de datos compuesta de N filas por 2 columnas, correspondientes a la aptitud y a la productividad.

Proceso: Se emplean cuatro subalgoritmos para realizar lo siguiente:

- Presentación del ejercicio
- Cálculo de la media aritmética
- Cálculo de la desviación típica
- Cálculo de las sumatorias X, Y, y Sumatoria de X^2 .

Salida: Imprimir la media aritmética de X, Y. La desviación típica de X, Y, y la ecuación de la recta Y.

b) Variables a utilizar

N = Límite de empleados

I = Variable índice de la estructura PARA

CODE = código del Empleado

DATOS = Matriz de datos

MED = Media Aritmética

DES = Desviación Típica

SUMXC = Sumatoria de los valores de X elevados al cuadrado

SUMXY = Sumatoria de los valores X*Y

SUM = Sumatoria

SUMD = Sumatoria de la Desviación

SDC = Sumatoria de la Desviación al cuadrado

c) Algoritmo principal

```

Paso 1      INICIO

Paso 2      PARA I DE 1 HASTA N HACER
              LEER CODE(I)
              IMPRIMIR CODE(I)
              FIN_PARA

Paso 3      PARA I DE 1 HASTA N HACER
              PARA J DE 1 HASTA 2 HACER
                LEER DATOS(I,J)
                IMPRIMIR DATOS(I,J)
              FIN_PARA
              FIN_PARA

Paso 5      LLAMAR MEDIA(N, DATOS, MED)

Paso 6      LLAMAR DESVIACION(N, DATOS, MED, DES)

Paso 7      LLAMAR SUMATORIAS(N, DATOS, SUMXY, SUMXC)

Paso 8      IMPRIMIR "La media aritmética de X es", MED(1)

Paso 9      IMPRIMIR "La desviación típica de X es", DES(1)

Paso 10     IMPRIMIR "La media aritmética de Y es", MED(2)

Paso 11     IMPRIMIR "La desviación típica de Y es", DES(2)

Paso 12     A ← (SUMXY + N * MED(1) * MED(2))/(SUMXC - N *
              MED(1)↑2)

Paso 13     b ← MED(2) - A * MED(1)

Paso 14     IMPRIMIR "La ecuación de la recta para predecir
              la productividad es "Y= ",A," X +",B

Paso 15     FIN

```

PROCEDIMIENTO PRESENTACION

Paso 1 **INICIO**

Paso 2 **IMPRIMIR "Cálculo estadístico. Ecuación de 1a"**

Paso 3 **IMPRIMIR "Recta para predecir la productividad"**

Paso 4 **IMPRIMIR "De los Trabajadores de la"**

Paso 5 **IMPRIMIR "Compañía XXX"**

Paso 6 **FIN PRESENTACION**

PROCEDIMIENTO MEDIA(N, DATOS, MED)

Paso 1 **INICIO**

Paso 2 **PARA J DE 1 HASTA 2 HACER**

SUM ← 0

PARA I DE 1 HASTA N HACER

SUM ← SUM + DATOS(I,J)

FIN_PARA

MED(J) ← SUM / N

Paso 3 **FIN MEDIA**

PROCEDIMIENTO DESVIACION(N, DATOS, MED, DES)

Paso 1 **INICIO**

Paso 2 **PARA J DE 1 HASTA 2 HACER**

SUMD ← 0

PARA I DE 1 HASTA N HACER

SUMD ← SUMD + DATOS(I,J)²

FIN_PARA

SDC(J) ← SUMD / N

DES(J) ← (SDC(J) - MED(J)²)^{1/2}

FIN_PARAPaso 3 **FIN DESVIACION**

PROCEDIMIENTO SUMATORIAS(N, DATOS, SUMXY, SUMXC)

Paso 1 **INICIO**Paso 2 **SUMXY ← 0**Paso 3 **PARA I DE 1 HASTA N HACER****SUMXY ← SUMXY + DATOS(I,1) * DATOS(I,2)****FIN_PARA**Paso 4 **SUMXC ← 0**Paso 5 **PARA I DE 1 HASTA N HACER****SUMXC ← SUMXC + DATOS(I,1)12****FIN_DATOS**Paso 6 **FIN SUMATORIAS****4 PROBLEMAS RESUELTOS****EXERCICIO**

Calcular la depreciación de activos por el método de la línea recta. La información por cada registro consta de: año de compra del activo, costo del activo y vida útil. La salida debe contener la misma información leída, además de la depreciación anual, depreciación acumulada y saldo.

Ejemplo de depreciación en línea recta: Depreciar un artículo (a 5 años) cuyo precio es de \$1'000.000.

$$\text{Depreciación 1er año} = \frac{1'000.000}{5} = 200.000$$

$$\text{Depreciación } 2^{\text{o}} \text{ año} = \frac{1'000.000}{5} = 200.000$$

. . .
 . . .
 . . .

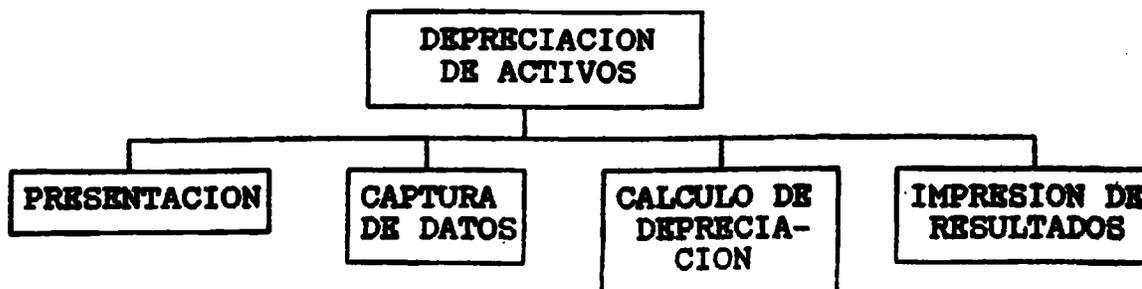
$$\text{Depreciación } 5^{\text{o}} \text{ año} = \frac{1'000.000}{5} = 200.000$$

La depreciación por el método de la línea recta es constante durante la vida útil del activo.

Solución

a) Análisis

Estructura del algoritmo



Entrada: Leer el límite de datos a procesar.

Inicializar la depreciación acumulada en cero.

Leer año de compra del activo, costo del activo y vida útil.

Proceso: Se elaboran los siguientes procedimientos

IMPRIMIR "Costo del Activo", COS
 IMPRIMIR " Vida Util del Activo", VU
 PARA L DE 1 HASTA VU HACER
 LLAMAR DEPRECIACION(COS, VU, DEP,
 DEPAC, SDO)
 LLAMAR IMPRIMIR_DATOS(ANO, DEP, DEPAC,
 SDO)
 ANO ← ANO + 1

FIN_PARA

FIN_PARA

Paso 6 FIN

PROCEDIMIENTO PRESENTACION

Paso 1 INICIO
 Paso 2 IMPRIMIR "Cálculo de Depreciación de Activos"
 Paso 3 IMPRIMIR "Por el método de la Línea Recta"
 Paso 4 IMPRIMIR "Mediante la aplicación de la metodo-
 logia"
 Paso 5 IMPRIMIR " DISEÑO DESCENDENTE "
 Paso 6 FIN PRESENTACION

PROCEDIMIENTO LEER_DATOS (ANO, COS, VU)

Paso 1 INICIO
 Paso 2 ESCRIBIR "Teclee el año de compra del activo"
 Paso 3 Leer ANO
 Paso 4 ESCRIBIR "Teclee el costo del activo"

- Paso 5 Leer COS
- Paso 6 ESCRIBIR "Teclee la vida útil del activo"
- Paso 7 Leer VU
- Paso 8 FIN LEER_DATOS

PROCEDIMIENTO DEPRECIACION (COS, VU, DEP, DEPAC, SDO)

- Paso 1 INICIO
- Paso 2 $DEP \leftarrow COS / VU$
- Paso 3 $DEPAC \leftarrow DEPAC + DEP$
- Paso 4 $SDO \leftarrow COS - DEPAC$
- Paso 5 FIN DEPRECIACION

PROCEDIMIENTO IMPRIMIR_DATOS (ANO, DEP, DEPAC, SDO)

- Paso 1 INICIO
- Paso 2 IMPRIMIR "Año de depreciación del Activo", ANO
- Paso 3 IMPRIMIR "Depreciación", DEP
- Paso 4 IMPRIMIR "Depreciación acumulada", DEPAC
- Paso 5 IMPRIMIR "Saldo", SDO
- Paso 6 FIN IMPRIMIR_DATOS

EJERCICIO

Calcular la Depreciación de Activos por el método de la suma de los dígitos de los años.

Ejemplo. Depreciar un vehículo (a 5 años) cuyo precio es de \$1'000.000=

La suma de los dígitos de los años = $1+2+3+4+5 = 15$

Se suman los dígitos de los años de acuerdo a la vida útil.

$$\text{Depreciación 1er año} = \frac{5}{15} \times 1'000.000$$

$$\text{Depreciación 2º año} = \frac{4}{15} \times 1'000.000$$

.

.

.

$$\text{Depreciación 5º año} = \frac{1}{15} \times 1'000.000$$

Solución

a) Análisis

Tanto la entrada como la salida de datos son idénticas al ejercicio anterior y por estar ya definidas en los procedimientos no se especificarán nuevamente.

Proceso: Se acoplan dos procedimientos, uno de presentación para indicar cuál es el método de depreciación a utilizar, el otro para el cálculo de la depreciación correspondiente a este método.

Se aplican las siguientes fórmulas adicionales:

Sumatoria de dígitos = $(1 + \text{vida útil}) * \text{vida útil} / 2$.

Disminución de la vida útil = $\text{vida útil} + 1 - \text{el valor del índice de la estructura interna PARA}$.

b) Variables a utilizar

N = Límite de datos a procesar

AÑO = Año de compra del Activo

COS = Costo del Activo

VU = Vida útil

DEP = Depreciación

DEPAC = Depreciación acumulada

SDO = Saldo

SUMD = Sumatoria de dígitos

DVU = Disminución de la vida útil (NUMERADOR)

J, L = Indices de las estructuras PARA

c) Algoritmo Principal

Paso 1 **INICIO**

Paso 2 **LLAMAR OTRA_PRESENTACION**

Paso 3 **ESCRIBIR "Teclee el número de datos a procesar"**

Paso 4 **Leer N**

Paso 5 **PARA J DE 1 HASTA N HACER**

 DEPAC ← 0

LLAMAR LEER_DATOS(ANO,COS,VU)

IMPRIMIR "Costo del Activo", COS

IMPRIMIR "Vida Util del Activo", VU

 SUMD ← (1 + VU) * VU / 2

PARA L DE 1 HASTA VU HACER

LLAMAR SUMA_DIGITOS(L,COS,VU,DEP,DEPAC,SDO)

LLAMAR IMPRIMIR_DATOS(ANO, DEP, DEPAC, SDO)

 ANO ← ANO + 1

FIN_PARA

FIN_PARA

Paso 6 FIN

PROCEDIMIENTO OTRA_PRESENTACION

Paso 1 INICIO

Paso 2 IMPRIMIR "Cálculo de la Depreciación de Activos"

Paso 3 IMPRIMIR "Por el método de la Suma de los"

Paso 4 IMPRIMIR "Digitos de los Años mediante la"

Paso 5 IMPRIMIR "Aplicación del Diseño Descendente"

Paso 6 FIN OTRA_PRESENTACION

PROCEDIMIENTO SUMA_DIGITOS (L, COS, VU, DEP, DEPAC, SDO)

Paso 1 INICIO

Paso 2 $DVU \leftarrow VU + 1 - L$

Paso 3 $DEP \leftarrow DVU / SUMD * COS$

Paso 4 $DEPAC \leftarrow DEPAC + DEP$

Paso 5 $SDO \leftarrow COS - DEP$

Paso 6 FIN SUMA_DIGITOS

Otra solución al ejercicio de Depreciación de Activos por el método de la Suma de los Dígitos de los Años es la siguiente:

a) Análisis

Proceso: El cálculo de la suma de los dígitos de los años se realiza mediante el empleo de una estructura **PARA**.

Se crea otro procedimiento para el cálculo de la

suma de los dígitos de los años; los demás procedimientos ya están definidos en soluciones anteriores, por lo tanto se hará uso de ellos.

Algoritmo Principal

```
Paso 1      INICIO
Paso 2      LLAMAR OTRA_PRESENTACION
Paso 3      ESCRIBIR "Teclee el número de datos a procesar"
Paso 4      Leer N
Paso 5      PARA J DE 1 HASTA N HACER
              DEPAC ← 0
              SUMD ← 0
              LLAMAR LEER_DATOS(ANO, COS, VU)
              IMPRIMIR "Costo del Activo", COS
              IMPRIMIR "Vida Util del Activo", VU
              PARA L DE 1 HASTA VU HACER
                  SUMD ← SUMD + L
              FIN_PARA
              REPETIR
                  LLAMAR DEPRECIACION_DIGITOS(COS, VU, SUMD,
                  DEP, DEPAC, SDO)
                  LLAMAR IMPRIMIR_DATOS(ANO, DEP, DEPAC, SDO)
                  VU ← VU - 1
                  ANO ← ANO + 1
              HASTA_QUE VU = 0
FIN_PARA
```

Paso 6 **FIN**

PROCEDIMIENTO DEPRECIACION_DIGITOS (COS, VU, SUMD, DEP,
DEPAC, SDO)

Paso 1 **INICIO**

Paso 2 $DEP \leftarrow VU / SUMD * COS$

Paso 3 $DEPAC \leftarrow DEPAC + DEP$

Paso 4 $SDO \leftarrow COS - DEPAC$

Paso 5 **FIN DEPRECIACION_DIGITOS**

EJERCICIO

Se tiene un lote de registros cada uno con la siguiente información: código, nombre, salario bruto y número de hijos. El proceso termina cuando el código ≤ 0 .

Calcular el salario neto, el valor devengado por bonificación anual, auxilio de transporte y subsidio familiar. Calcular además, el total de ingresos, total de egresos y total pagado por toda la nómina.

La bonificación anual es del 50% para salarios brutos inferiores a \$60.000, en caso contrario es del 30%.

El auxilio de transporte es de \$3.000 para salarios brutos inferiores a \$80.000.

El subsidio familiar es de \$2000 por hijo, para salarios brutos inferiores a \$100.000.

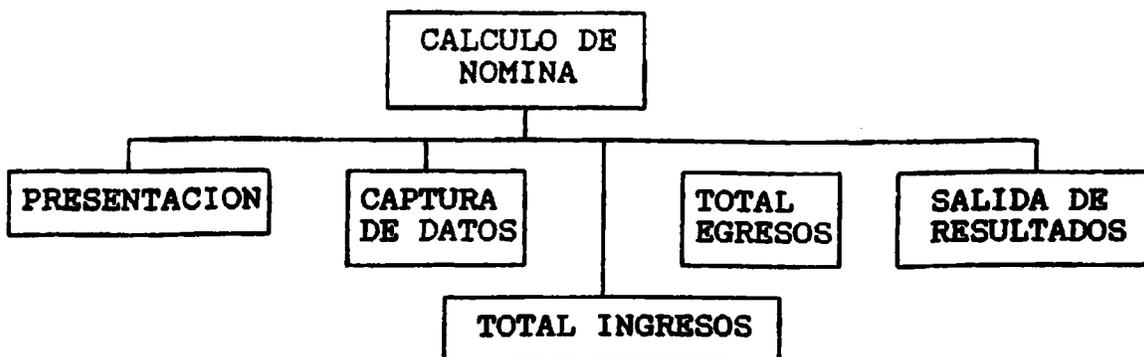
Los seguros sociales y la Retención en la fuente se calculan según la siguiente tabla:

SALARIO BRUTO (S.B.)	SEGURO SOCIAL	RETENCION EN LA FUENTE
Hasta 40.000	1% del S.B.	0% del S.B.
40.001 - 60.000	3% del S.B.	2% del S.B.
60.001 - 80.000	5% del S.B.	4% del S.B.
80.001 - 100.000	7% del S.B.	6% del S.B.
100.001 - y mas	10% del S.B.	8% del S.B.

Solución

a) Análisis

Estructura del algoritmo



Entrada: Leer código, nombre, salario bruto y número de hijos.

Inicializar el acumulador de salarios netos.

Se deben procesar tantos registros mientras el código sea menor o igual a cero.

Proceso: Se emplean cinco subalgoritmos para realizar lo siguiente:

- Presentación del ejercicio
- Capturar datos
- Totalizar ingresos
- Totalizar egresos
- Presentación de los cálculos hallados.

Salida: Imprimir todo lo leído y calculado

b) Variables a utilizar

COD = Código

NOM = Nombre

SB = Salario bruto

NH = Número de hijos

BON = Bonificación anual

ST = Subsidio de transporte

SF = Subsidio familiar

SS = Seguro social

RET = Retención en la fuente

TOTIN = Total de Ingresos

TOTEG = Total de egresos

SN = Salario neto

TOTNOM = Total nómina pagada

c) Algoritmo principal

Paso 1 INICIO

Paso 2 LLAMAR PORTADA

Paso 5 Leer NOM
 Paso 6 ESCRIBIR "Teclee el salario bruto"
 Paso 7 Leer SB
 Paso 8 ESCRIBIR "Teclee el número de hijos"
 Paso 9 Leer NH
 Paso 10 FIN CAPTURA DATOS

PROCEDIMIENTO INGRESOS(SB, NH, BON, ST, SF, TOTIN)

Paso 1 INICIO
 Paso 2 SI SB < 60.000
 ENTONCES BON ← SB * 0.5
 SINO BON ← SB * 0.3
 FINSI
 Paso 3 SI SB < 80.000
 ENTONCES ST ← 3.000
 SINO ST ← 0
 FINSI
 Paso 4 SI SB < 100.000
 ENTONCES SF ← NH * 2000
 SINO SF ← 0
 FINSI
 Paso 5 TOTIN ← SB + BON + ST + SF
 Paso 6 FIN INGRESOS

PROCEDIMIENTO EGRESOS (SB, SS, RET, TOTEG)

Paso 1 INICIO
 Paso 2 SI SB <= 40.000

ENTONCES SS ← SB * 0.01

RET ← 0

SINO SI SB ≤ 60.000

ENTONCES SS ← SB * 0.03

RET ← SB * 0.02

SINO SI SB ≤ 80.000

ENTONCES SS ← SB * 0.05

RET ← SB * 0.04

SINO SI SB ≤ 100.000

ENTONCES SS ← SB * 0.07

RET ← SB * 0.06

SINO SS ← SB * 0.1

RET ← SB * 0.08

FINSI

FINSI

FINSI

FINSI

Paso 3 TOTEG ← SS + RET

Paso 4 **FIN EGRESOS**

PROCEDIMIENTO SALIDA (COD, NOM, SB, NH, BON, ST, SF, SS,
RET, TOTIN, TOTEG, SN)

Paso 1 **INICIO**

Paso 2 **IMPRIMIR** "Código del trabajador", COD

Paso 3 **IMPRIMIR** "Nombre del trabajador", NOM

Paso 4 **IMPRIMIR** "Salario bruto", SB

- Paso 5 **IMPRIMIR** "Número de hijos", NH
 Paso 6 **IMPRIMIR** "Bonificación", BON
 Paso 7 **IMPRIMIR** "Subsidio de transporte", ST
 Paso 8 **IMPRIMIR** "Subsidio familiar", SF
 Paso 9 **IMPRIMIR** "Seguro social", SS
 Paso 10 **IMPRIMIR** "Retención en la fuente", RET
 Paso 11 **IMPRIMIR** "Total de Ingresos", TOTIN
 Paso 12 **IMPRIMIR** "Total de Egresos", TOTEG
 Paso 13 **IMPRIMIR** "Salario Neto", SN
 Paso 14 **FIN SALIDA**

EJERCICIO

Para un lote de N datos no superior a 1.000, calcular el rango, media aritmética, desviación media, desviación estandar y varianza, según las siguientes fórmulas:

RANGO = DATO MAYOR - DATO MENOR

$$\text{MEDIA ARITMETICA } \bar{X} = \frac{\sum X_i}{N}$$

$$\text{DESVIACION MEDIA} = \frac{\sum |X_i - \bar{X}|}{N}$$

$$\text{DESVIACION ESTANDAR} = \sqrt{\frac{\sum (X_i - \bar{X})^2}{N}}$$

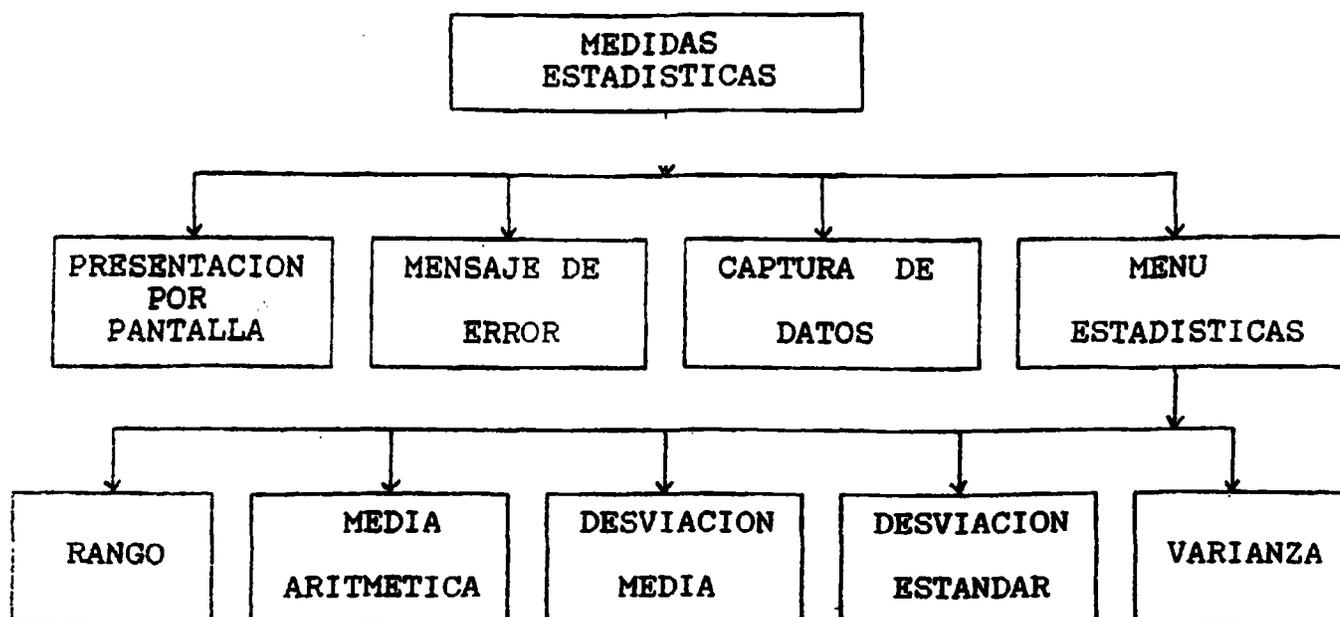
$$\text{VARIANZA} = \text{DESVIACION ESTANDAR}^2$$

Calcular e imprimir lo anterior mediante la utilización de un menú.

Solución

Análisis

Estructura del Algoritmo



Entrada: Leer el límite y los datos a procesar, los cuales son almacenados en un vector.

Inicializar el acumulador de datos en cero.

Proceso: Se utilizan los siguientes subalgoritmos

- Presentación por pantalla
- Mensaje de error

- Menú estadísticas
- Captura de datos
- Cálculo del rango
- Cálculo de la media aritmética
- Cálculo de la desviación media
- Cálculo de la desviación estandar
- Cálculo de la varianza

Se emplean validaciones de datos, tanto en el procedimiento Menú para controlar que la operación seleccionada se encuentre dentro del rango de alternativas permitidas, como en el procedimiento de captura de datos para validar que el total de datos no exceda de 1.000 tal como reza el enunciado del ejercicio. En ambas validaciones cuando se detecta una inconsistencia, se activa el procedimiento Error el cual envía un mensaje al usuario.

Todo el ejercicio se desarrolla con base en procedimientos, buscando simplicidad y mayor entendimiento en cuanto al diseño del mismo se refiere.

Salida: Cualquiera de las alternativas del MENU seleccionada genera su propia salida impresa, algunas por pantalla (escribir), otras por impresora (imprimir).

b) Variables a utilizar

N = Límite de datos a procesar

X = Dato a ser procesado

Respuesta = Valor asumido cuando se produce el mensaje de error.

Opción = Valor seleccionado del menú

Mayor = Elemento mayor de los datos analizados

Menor = Elemento menor de los datos analizados

R = Rango

SUM = Sumatoria de elementos

XMED = Media aritmética

DM = Desviación estandar

S = Varianza

c) Algoritmo principal

{ CALCULOS ESTADISTICOS MEDIANTE LA UTILIZACION DE MENU }

Paso 1 INICIO

Paso 2 LLAMAR PANTALLA

Paso 3 REPETIR

LLAMAR MENU(ERROR, OPCION)

EN CASO DE OPCION HACER

1: LLAMAR RANGO(N,X,MAYOR,MENOR,R,RESPUESTA)

2: LLAMAR MEDIA(CAPTURA, MED, RESPUESTA)

3: LLAMAR DESMED(CAPTURA, MED, DM, RESPUESTA)

4: LLAMAR ESTANDAR(SUMDES, DESV, RESPUESTA)

5: LLAMAR VARIANZA(SUMDES, S, RESPUESTA)

FIN_CASO**HASTA_QUE OPCION = 6****Paso 4 FIN****PROCEDIMIENTO PANTALLA****{ PRESENTACION INICIAL DEL PROGRAMA }****Paso 1 INICIO****Paso 2 ESCRIBIR "UNIVERSIDAD NACIONAL"****Paso 3 ESCRIBIR "PROGRAMA PARA CALCULAR"****Paso 4 ESCRIBIR "ALGUNAS MEDIDAS ESTADISTICAS"****Paso 5 ESCRIBIR "A TRAVES DEL MANEJO DE"****Paso 6 ESCRIBIR " MENU"****Paso 7 ESCRIBIR "AUTOR: ALONSO TAMAYO ALZATE"****Paso 8 ESCRIBIR "MANIZALES, ENERO DE 1990"****Paso 9 FIN PANTALLA****PROCEDIMIENTO ERROR(RESPUESTA)****{ PRODUCE MENSAJE DE ERROR POR PANTALLA }****Paso 1 INICIO****Paso 2 ESCRIBIR "ERROR EN DATOS. PRESIONE LA TECLA
ENTER PARA CONTINUAR"****Paso 3 LEER RESPUESTA****Paso 4 ESCRIBIR " "****Paso 5 FIN ERROR**

PROCEDIMIENTO MENU(ERROR, OPCION)

{ CREA EL MENU DE ESTADISTICA }

Paso 1 INICIO

Paso 2 ESCRIBIR "MENU DE ESTADISTICA"

Paso 3 ESCRIBIR " "

Paso 4 ESCRIBIR " 1 - RANGO"

Paso 5 ESCRIBIR " 2 - MEDIA ARITMETICA"

Paso 6 ESCRIBIR " 3 - DESVIACION MEDIA"

Paso 7 ESCRIBIR " 4 - DESVIACION ESTANDAR"

Paso 8 ESCRIBIR " 5 - VARIANZA"

Paso 9 ESCRIBIR " 6 - FIN DEL PROCESO"

Paso 10 REPETIR

ESCRIBIR "TECLEE LA OPCION SELECCIONADA"

LEER OPCION

SI (OPCION < 1) o (OPCION > 6)

ENTONCES LLAMAR ERROR(RESPUESTA)

FINSI

HASTA_QUE (OPCION >= 1) y (OPCION <= 6)

Paso 11 FIN MENU

PROCEDIMIENTO CAPTURA(N,X)

{ PRODUCE CAPTURA DE DATOS }

Paso 1 INICIO

Paso 2 ESCRIBIR "SON DATOS NUEVOS PARA PROCESAR? (S/N)"

Paso 3 LEER RESPUESTA

Paso 4 SI (RESPUESTA="S") O (RESPUESTA="s")

Paso 5

ENTONCES ESCRIBIR "TECLEE EL TOTAL DE
 .DATOS A PROCESAR"

REPETIR

LEER N

SI (N<=000) O (N>999)

ENTONCES LLAMAR ERROR(Respuesta)

FINSI

HASTA_QUE (N>000) Y (N<=999)

PARA J DE 1 HASTA N HACER

X(J) ← 0

FIN_PARA

ESCRIBIR "DIGITE CADA DATO Y PRESIONE LA
 TECLA ENTER"

PARA J DE 1 HASTA N HACER

LEER X(J)

FIN_PARA

ESCRIBIR "LOS DATOS ORIGINALES SON: "

PARA J DE 1 HASTA N HACER

IMPRIMIR X(J)

FIN_PARA

FINSI

Paso 6 FIN CAPTURA

PROCEDIMIENTO RANGO(N, X, MAYOR, MENOR, R, Respuesta)

{ CALCULA EL RANGO DE UN CONJUNTO DE DATOS }

Paso 1 INICIO

```

Paso 2      LLAMAR CAPTURA(N, X)
Paso 3      MAYOR ← X(1)
Paso 4      MENOR ← X(1)
Paso 5      PARA J DE 2 HASTA N HACER
              SI X(J) > MAYOR
                ENTONCES MAYOR ← X(J)
              SINO SI X(J) < MENOR
                ENTONCES MENOR ← X(J)
              FINSI
            FINSI
          FIN_PARA
Paso 6      R ← MAYOR - MENOR
Paso 7      IMPRIMIR "LOS DATOS ANALIZADOS SON: "
Paso 8      PARA J DE 1 HASTA N HACER
              IMPRIMIR X(J)
            FIN_PARA
Paso 9      IMPRIMIR "EL ELEMENTO MAYOR ES ", MAYOR
Paso 10     IMPRIMIR "EL ELEMENTO MENOR ES ", MENOR
Paso 11     IMPRIMIR "EL RANGO ES ", R
Paso 12     ESCRIBIR "PRESIONE LA TECLA ENTER PARA CONTINUAR"
Paso 13     LEER RESPUESTA
Paso 14     FIN RANGO

```

PROCEDIMIENTO MED(N, X, XMED)

{ CALCULO DE LA MEDIA ARITMETICA }.

Paso 1 INICIO
Paso 2 SUM ← 0
Paso 3 PARA J DE 1 HASTA N HACER
 SUM ← SUM + X(J)
 FIN_PARA
Paso 4 XMED ← SUM / N
Paso 5 FIN MKD

PROCEDIMIENTO MEDIA(CAPTURA, MED, RESPUESTA)

{ IMPRESION DE LA MEDIA ARITMETICA }

Paso 1 INICIO
Paso 2 LLAMAR CAPTURA(N, X)
Paso 3 LLAMAR MED(N, X, XMED)
Paso 4 IMPRIMIR LOS DATOS ANALIZADOS SON:
Paso 5 PARA J DE 1 HASTA N HACER
 IMPRIMIR X(J)
 FIN_PARA
Paso 6 IMPRIMIR LA MEDIA ARITMETICA ES . XMED
Paso 7 ESCRIBIR "PRESIONE LA TECLA ENTER PARA CONTINUAR"
Paso 8 LEER RESPUESTA
Paso 9 FIN MEDIA

PROCEDIMIENTO DESMED(CAPTURA, MED, DM, RESPUESTA)

CALCULO DE LA DESVIACION MEDIA }

Paso 1 INICIO

Paso 2 **LLAMAR CAPTURA(N, X)**
Paso 3 **LLAMAR MED(N, X, XMED)**
Paso 4 **SUM ← 0**
Paso 5 **PARA J DE 1 HASTA N HACER**
 SUM ← SUM + | X(J)-XMED |
 FIN_PARA
Paso 6 **DM ← SUM/N**
Paso 7 **IMPRIMIR "LOS DATOS ANALIZADOS SON:**
Paso 8 **PARA J DE 1 HASTA N HACER**
 IMPRIMIR X(J)
 FIN_PARA
Paso 9 **IMPRIMIR "LA DESVIACION MEDIA ES: ", DM**
Paso 10 **ESCRIBIR "PRESIONE LA TECLA ENTER PARA CONTINUAR"**
Paso 11 **LEER RESPUESTA**
Paso 12 **FIN DESMED**

PROCEDIMIENTO SUMDES(CAPTURA, MED, SUM)

{ CALCULO DE LA SUMATORIA DE LAS DESVIACIONES }

Paso 1 **INICIO**
Paso 2 **LLAMAR CAPTURA(N, X)**
Paso 3 **LLAMAR MED(N, X, XMED)**
Paso 4 **SUM ← 0**
Paso 5 **PARA J DE 1 HASTA N HACER**
 SUM ← SUM + (X(J) - XMED)²
 FIN_PARA
Paso 6 **FIN SUMDES**

PROCEDIMIENTO ESTANDAR(SUMDES, DESV, RESPUESTA)

{ CALCULO DE LA DESVIACION ESTANDAR }

- Paso 1 **INICIO**
- Paso 2 **LLAMAR SUMDES(CAPTURA, MED, SUM)**
- Paso 3 **DESV \leftarrow (SUM/N)^{1/2}**
- Paso 4 **IMPRIMIR "LOS DATOS ANALIZADOS SON: "**
- Paso 5 **PARA J DE 1 HASTA N HACER**
 IMPRIMIR X(J)
 FIN_PARA
- Paso 6 **IMPRIMIR "LA DESVIACION ESTANDAR ES", DESV**
- Paso 7 **ESCRIBIR "PRESIONE LA TECLA ENTER PARA CONTINUAR"**
- Paso 8 **LEER RESPUESTA**
- Paso 9 **FIN ESTANDAR**

PROCEDIMIENTO VARIANZA(SUMDES, S, RESPUESTA)

{ CALCULO DE LA VARIANZA }

- Paso 1 **INICIO**
- Paso 2 **LLAMAR SUMDES(CAPTURA, MED, SUM)**
- Paso 3 **S \leftarrow SUM/N**
- Paso 4 **IMPRIMIR "LOS DATOS ANALIZADOS SON: "**
- Paso 5 **PARA J DE 1 HASTA N HACER**
 IMPRIMIR X(J)
 FIN_PARA
- Paso 6 **IMPRIMIR "LA VARIANZA ES: ", S**
- Paso 7 **ESCRIBIR "PRESIONE LA TECLA ENTER PARA CONTINUAR"**
- Paso 8 **LEER RESPUESTA**
- Paso 9 **FIN VARIANZA**

4.5 PROBLEMAS PROPUESTOS

1- Determinar que valor se imprime de acuerdo al siguiente algoritmo:

Algoritmo Principal

Paso 1 **INICIO**
 Paso 2 $A \leftarrow 2$
 Paso 3 $B \leftarrow 5$
 Paso 4 Llamar $X(A, B, Z)$
 Paso 5 **IMPRIMIR Z**
 Paso 6 **FIN**

PROCEDIMIENTO X (A, B, Z)

Paso 1 **INICIO**
 Paso 2 $Y \leftarrow A - 2 * B \uparrow 2$
 Paso 3 $D \leftarrow A \uparrow 2 - B \uparrow 2$
 Paso 4 $E \leftarrow -Y / D \uparrow (1/5)$
 Paso 5 $F \leftarrow 1/(Y * D \uparrow (1/3))$
 Paso 6 $Z \leftarrow E - F$
 Paso 7 **FIN X**

- 2- Elaborar un procedimiento para reducir un ángulo expresado en radianes a grados, minutos y segundos.
- 3- Elaborar un procedimiento para convertir el peso de un artículo dado en arrobas a Kilogramos, libras y gramos.
- 4- Para un grupo de N estudiantes se dispone de 3 notas de

evaluaciones por registro cada una con un valor del 20% y una nota del proyecto final con un valor del 40%. Calcular la nota definitiva mediante la aplicación de la filosofía del Diseño Descendente.

- 5- Calcular mediante la aplicación de un procedimiento el valor del servicio de energía, si el valor del Kilovatio es de \$20 =, además se cobra un cargo fijo según la siguiente tabla.

ESTRATO	CARGO FIJO
1	50
2	100
3	300
4	500
5	1.000
6	2.000

Por cada registro se lee el número del contador, consumo y estrato social. El proceso termina cuando el estrato sea menor o igual a cero.

- 6- Realizar una prueba de escritorio al siguiente algoritmo y determinar que calcula.

ALGORITMO PRINCIPAL

Paso 1 INICIO

Paso 2 Leer N

Paso 3 $M \leftarrow N + 1$
Paso 4 **PARA I DE 1 HASTA M HACER**
 $A(I) \leftarrow \text{FACT}(N) / (\text{FACT}(I-1) * \text{FACT}(N-I+1))$
 FIN_PARA
Paso 5 **PARA I DE 1 HASTA M HACER**
 IMPRIMIR A(I)
 FIN_PARA
Paso 6 **FIN**

FUNCION FAC(N)

Paso 1 **INICIO**
Paso 2 $\text{FACT} \leftarrow 1$
Paso 3 $\text{SI } N > 1$
 ENTONCES PARA I DE 2 HASTA N HACER
 $\text{FACT} \leftarrow \text{FACT} * I$
 FIN_PARA
 FINSI
Paso 4 $\text{FAC} \leftarrow \text{FACT}$
Paso 5 **FIN**

BIBLIOGRAFIA

Serie SCHAUM. Introducción a las ciencias de las computadoras. Mc. Graw Hill. 1984.

LEVINE Gutierrez Guillermo. Introducción a la computación. Mc. Graw Hill. 1985.

VERZELLO J. Robert. Procesamiento de datos. Mc. Graw Hill. 1985.

GOLDSCHLAGER Andrew Lister. Introducción moderna a la ciencia de la computación. Un enfoque algorítmico. Prentice Hall. 1986.

TREMBLAY Jean Paul. Introducción a las ciencias de los computadores. Un enfoque algorítmico. Mc. Graw Hill. 1987

SINGELMANN Jay. Programación lógica aplicada a la administración. Un enfoque estructurado. Mc. Graw Hill. 1987

AHO V. Alfred. Estructura de datos y algoritmos. Addison Wesley. 1988.

WIRTH Niklaus. Algoritmos + Estructura de datos = Programas. Ediciones del Castillo.

INDICE ALFABETICO

Acumulador, 5, 60, 61, 62, 63, 67, 74, 119, 120, 121, 132,
224, 253, 259
Ada, 33, 232
Addison, 272
Adyacentes, 179, 180
AHO, 272
Ahorrador, 30
AL-KHOWARIZMI, 6
ALGORITMUS, 6
Anchura, 29, 30
AND, 18
Anidados, 5, 25, 26, 47
ARCHIVO, 107
ARGUMENTOS, 216, 217, 221, 224, 225, 227, 230, 232
ARITMETICA, 18, 239, 258, 259, 263, 265, 266
ARMONICA, 223, 224, 225
Arreglo, 144, 145, 146, 148, 149, 150, 151, 157, 158, 171,
176, 186, 187
Ascendente, 235, 236, 237
ASIGNACION, 16

BIBLIOGRAFIA, 272
Bidimensional, 186
Bifurcacio, 32

Ciclicos, 5
Calificaciones, 108, 109, 137, 196
Cambia, 10, 15, 157
CASCADA, 42
CASE, 5, 33, 47, 48
CATEGORIA, 49, 50, 51
CENTINELA, 6, 107, 108, 10, 111, 131
CESANTIAS, 231, 233, 234
CICLICAS, 28, 55
CLASIFICACION, 5, 6, 25, 216
Columna, 186, 187, 191, 192, 205, 207, 209, 210, 213
Combinatoria, 88, 89, 219, 220, 221
COMISION, 123
Compiladores, 15, 21, 227
Computador, 6, 66, 107, 145, 157, 226
CONJUNCION, 18

Constante, 4, 15, 17, 19, 85, 110, 145, 146, 147, 244
Contador, 5, 67, 68, 69, 70, 71, 72, 73, 74, 77, 99, 108,
109, 134, 137, 138, 180, 194, 223, 224, 270
Contadora, 66, 68, 69
CONTENIDO, 4, 17, 27, 34, 113, 148, 150, 157, 161, 187
Contribuyente, 45, 46
Coseno, 217, 218, 219
Cosenos, 217
CUALITATIVOS, 26
CUANTITATIVOS, 26

DE-HACER, 47
DECISION, 9
DEPRECIACION, 244, 246, 247, 251, 252
DESCENDENTEMENTE, 236
DESVIACION, 239, 241, 242, 243, 258, 259, 263, 266, 267,
268
Diagonal, 198, 199, 209, 210, 213
Diagonales, 198
Diagramas, 33
Dimensio, 151, 186
Dimensiones, 2, 6, 29, 186
DISEÑO, 214, 260
DISYUNCION, 18
DOWN, 215
DUMMY, 6, 107, 108

ELSE, 5, 33, 39
END, 48, 56, 83
ENDIF, 33, 35, 39
Error, 69, 80, 89, 259, 260, 261, 261, 263, 264
ESTADISTICAS, 259, 262
ESTRUCTURAS, 5, 6, 24, 28, 32, 33, 42, 55, 92, 95, 103,
119, 157, 161, 188, 215, 249
Estudiante, 2, 6, 84, 85, 136, 171, 172, 196
Etiqueta, 122, 123, 124
EXIT, 35
EXPRESION, 18, 48

Factorial, 78, 79, 81, 88, 89, 98, 102, 103, 119, 120, 121,
220, 221, 222
FIBONACCI, 112
FIN-MIENTRAS, 56
FINCASO, 48, 52
FOR, 5, 55, 82, 83, 85, 86, 87, 89, 96, 99, 102, 106, 119
FUNCION, 216, 217, 222, 225, 271

GENERAL, 1, 33, 39, 48, 56, 75, 76, 83, 215, 216, 220, 221,
227
GENERALIDADES, 4, 6
GOLDSCHLAGER, 272
Grados, 217, 269

Graw, 272
Guillermo, 272
Gutierrez, 272

Hall, 272
Herramientas, 1, 2, 144
Hijos, 252, 253, 254, 256, 258
Hill, 272
HORAS, 49, 50, 86, 87, 88, 104, 105, 106

IF, 5, 33, 35, 39, 42, 47, 48
IF-THEN-ELSE, 38
IMPRESION, 228, 231, 244, 266
Impresora, 230, 260
Impuesto, 45, 46
IN, 83, 119, 198
Inicializa, 57, 112, 150, 184
INICIAR, 35, 108, 148, 194
Instrucciones, 25, 33, 38, 54, 55, 61, 214, 216, 226, 230
INTERESES, 95, 96, 231, 232, 233, 235
Intervalo, 43, 77
INTRODUCCION, 4, 5, 6, 24, 144, 214
Inventario, 133, 168, 169, 170

JERARQUIA, 20
Jornal, 104

KHOWARIZMI, 6
Kilogramos, 269
Kilovatio, 270

LEVINE, 272
Libro, 100, 101
LIQUIDACION, 231
Longitud, 15, 29, 30, 110, 111

MATRICES, 186, 200, 201, 202, 203, 204
Mc, 272
MENU, 259, 260, 261, 262, 263
Mientras-Hacer, 56, 61
MODULOS, 6, 216
MOHAMMED, 6
Multiplicatorias, 88

NEGACION, 18
Niklaus, 272
NOMINA, 253
Nota, 36, 47, 54, 108, 109, 110, 136, 137, 144, 231, 270
Notas, 47, 107, 109, 136, 137, 138, 144, 145, 196, 197, 269
Observacio, 80, 99, 151, 152, 158, 190, 225, 230

OBSERVACION, 227
OPCION, 261, 262, 263
OPERACION, 18
Operador, 19
Operadores, 4, 17, 18, 19
Ordenamiento, 161, 235, 237
Ordenar, 12, 134

Patrimonio, 45, 46, 84, 85
PORTADA, 236, 237, 254, 255
Prentice, 272
Presentación, 231, 233, 3235, 239, 242, 244, 245, 246, 249,
250, 251, 253, 259, 262
Primera, 24, 76, 151, 153, 177, 201, 205, 210, 222
Prioridades, 15
Procedimiento, 98, 116, 205, 226, 227, 228, 229, 230, 233,
234, 237, 238, 242, 243, 246, 247, 250, 252, 255, 256, 257,
260, 262, 263, 264, 265, 266, 267, 268, 269, 270
Procedimientos, 6, 216, 217, 226, 227, 228, 230, 232, 237,
239, 244, 248, 251, 260, 265

PROLOGO, 1
Promedio, 72, 73, 108 109, 136, 144, 171, 172, 182, 183,
195, 196, 197, 207, 208

Radianes, 142, 269
Radio, 110, 111, 112
READ, 35
RECOMENDACIONES, 4, 14
Reglas, 4, 6, 19
REPEAT, 5, 55, 75, 76
REPEAT-UNTIL, 74
REPETICION, 10
RETENCION, 43, 253
Robert, 272

Salarios, 252, 253
Seno, 142, 217, 218, 219
SI-ENTONCES, 38
SI-ENTONCES-SINO, 38
SINGELMANN, 272
SINOPTICO, 26
Sistema, 14, 217
Sistemas, 218, 238
SUBALGORITMO, 220, 222, 223, 224, 225
Subalgoritmos, 235, 240, 253, 259
Subsidio, 252, 254, 258
Suscrito, 144, 145, 146

TREMBLAY, 272
Triangular, 198, 199
TRIANGULO, 228

Trigonome, 217

Unidad, 17, 57, 60, 84, 98, 104, 123, 223

Universidad, 26, 28, 84, 233, 262

UNTIL, 5, 55, 75

Validar, 260

Varianza, 166, 167, 258, 259, 260, 261, 263, 268

VECTORES, 151, 153, 158, 159, 167, 182, 184, 196, 205, 207,
213, 235, 236

VERZELLO, 272

Wesley, 272

WHILE, 5, 55, 56, 75, 76

WHILE-DO 56, 76

WIRTH, 272

WRITE, 35