**EJERCICIOS DE APRENDIZAJE**

**Para cada uno de los siguientes ejercicios realizar el análisis del problema e indicar cuáles son los datos de entrada y cuáles son los datos de salida. Escribir luego el programa en PSeInt.**

1. **Escriba un programa en donde se pida la edad del usuario. Si el usuario es mayor de edad se debe mostrar un mensaje por pantalla indicándolo.**

Proceso Ejercicio01

// Lo primero que haremos será definir una variable, que contenga

// la edad ingresada por el usuario

definir edadUsuario como entero;

// Luego pasaremos a imprimir por consola la pregunta

escribir "¿Cuantos años tiene?";

// Leemos la edad

leer edadUsuario;

// Ahora crearemos un condicional que valide laEeje edad ingresada

si edadUsuario<18 Entonces

// Diremos que si es mas chica, nos muestre en la consola

// que semos menores de edad y si no se cumple esa condicion

// que somos mayores

escribir "¡Sos menor de edad!";

SiNo

escribir "¡Sos mayor de edad!";

FinSi

FinProceso

1. **Un hombre desea saber si su sueldo es mayor al sueldo mínimo, el programa le pedirá al usuario su sueldo actual y el sueldo mínimo. Si el sueldo el mayor se debe mostrar un mensaje por pantalla indicándolo.**

proceso Ejercicio02

// Empezamos declarando dos variable de tipo entero que contendran el sueldo

// del usuario y el sueldo minimo.

definir sueldo, sueldoMin como entero;

// Ahora preguntaremos por consola cuales es el sueldo y de cuanto es el minimo.

escribir "¿De cuanto es su sueldo?";

// Leemos el valor

leer sueldo;

escribir "¿De cuanto es el sueldo minimo actual?";

leer sueldoMin;

// Terminaremos el ejercicio haciendo un condicional que compare esas dos variables

// y nos imprima por consola el resultado

si sueldo<sueldoMin entonces

escribir "Su sueldo es menor al sueldo minimo";

siNo

escribir "Usted ganas mas que el sueldo minimo";

finSi

finProceso

1. **Realizar un programa que pida un número al usuario. Si el número es mayor que 100 se deberá de imprimir en pantalla “Es Mayor”, y en caso contrario se deberá imprimir “Es Menor”.**

proceso Ejercicio03

// Declaramos la variable que guardara el numero a evaluar

definir num como entero;

// Preguntamos cual es el numero a comparar

escribir "Ingrese el numero a evaluar";

// Leemos el numero que ingreso el usuario

leer num;

// Ahora haremos un condicional que evalue si el numero es

// menor o mayor a 100 y nos diga el resultado

si num<100 entonces

escribir "El numero es menor a 100";

SiNo

escribir "El numero es mayor a 100";

finSi

finProceso

**4. Realiza un programa que sólo permita introducir los caracteres ‘S’ y ‘N’. Si el usuario ingresa alguno de esos dos caracteres se deberá de imprimir un mensaje por pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”.**

proceso Ejercicio04

// Para empezar declararemos 3 variables, dos seran nuestras constantes S y N y la otra

// la usaremos para guardar la letra ingresada por el usuario

definir letra, s, n como caracter;

// Le asignaremos a nuestra variable S y N sus respectivos valores

s="s";

n="n";

// Le diremos por consola al usuario que ingrese una letra, luego leeremos el dato

escribir "Ingrese una letra";

leer letra;

// Evaluaremos si la letra ingresada por el usuario es igual a S o N y imprimiremos el resultado

si letra = s o letra = n entonces

escribir "CORRECTO";

siNo

escribir "INCORRECTO";

finSi

finProceso

1. **Realizar un programa que pida un número y determine si ese número es par o impar. Mostrar en pantalla un mensaje que indique si el número es par o impar. Nota: investigar la función mod de Pseint.**

proceso Ejercicio05

// Empezaremos declarando una variable que contendra dentro el valor ingresado

definir num como entero;

// Luego por consola le diremos al usuario que ingrese un numero para evaluarlo

escribir "Ingrese un numero para evaluar si es par o no";

// Leeremos el numero ingresado

leer num;

// Y ahora con un condicional evaluaremos el numero ingresado e imprimiremos el resultado

si (num mod 2=0) entonces

escribir "El numero ingresado es par";

siNo

escribir "El numero es impar";

FinSi

finProceso

1. **Realizar un programa que pida tres notas y determine si un alumno aprueba o reprueba un curso, sabiendo que aprobará el curso si su promedio de tres calificaciones es mayor o igual a 70; y reprueba en caso contrario.**

proceso Ejercicio06

// Empezaremos declarando tres variables que seran correspondientes a las notas a promediar,

// luego declararemos otra variable que sera la constante que contendra el promedio minimo para aprobar

definir nota1, nota2, nota3, promedio como entero;

promedio=7;

// Ahora preguntaremos y leeremos las notas por consola

escribir "Ingrese la primera nota";

leer nota1;

escribir "Ingrese la segunda nota";

leer nota2;

escribir "Ingrese la tercera nota";

leer nota3;

// Guardamos en nuestra variable promedio el promedio de las tres notas

promedio = (nota1+nota2+nota3/3);

// Ya teniendo el promedio pasaremos a mostrar los resultados a travez de un condicional

si promedio<=7 entonces

escribir "Usted esta desaprobado";

siNo

escribir "Usted esta aprobado";

finSi

finProceso

1. **Realizar un programa que pida introducir solo frases o palabras de 6 de largo. Si el usuario ingresa una frase o palabra de 6 de largo se deberá de imprimir un mensaje por pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”. Nota: investigar la función Longitud() de Pseint.**

Algoritmo longitud\_palabra

//Definimos la variables palabra

Definir palabra Como Caracter;

//Pedimos al usuario que ingrese una palabra

Escribir "Ingrese una palabra que tenga una longitud de 6 carácteres: ";

Leer palabraIngresada;

//Asignamos la palabra ingresada por consola a nuestra variable

palabra = palabraIngresada;

//Comprobamos que la longitud de la palabra sea IGUAL a 6.

si(Longitud(palabra) == 6) Entonces

//Si es igual a 6 se mostrará un mensaje de "CORRECTO"

Escribir "CORRECTO"

SiNo

//En el caso de que no lo sea se imprimirá un mensaje de "INCORRECTO"

Escribir "INCORRECTO"

FinSi

FinAlgoritmo

1. **Realizar un programa que pida una frase o palabra y si la frase o palabra es de 4 caracteres de largo, el programa le sumará un signo de exclamación al final, y si no es de 4 caracteres el programa le sumará un signo de interrogación al final. El programa mostrará después la frase final. Nota: investigar la función Longitud() y Concatenar() de Pseint.**

Algoritmo concatenar\_palabra

//Definimos la variables palabra

Definir palabra Como Caracter;

//Pedimos al usuario que ingrese una palabra

Escribir "Dime una palabra!"

leer palabraIngresada

//Asignamos la palabra ingresada por consola a nuestra variable

palabra = palabraIngresada

//Comprobamos que la longitud de la palabra sea IGUAL a 4.

si(Longitud(palabra) == 4) Entonces

//si la frase o palabra es de 4 caracteres de largo, se le sumará un signo de exclamación al final

Escribir Concatenar(palabra,"!")

SiNo

//Si no se le sumará un signo de interrogación al final

Escribir Concatenar(palabra,"?")

FinSi

FinAlgoritmo

**9. Una tienda ofrece para los meses de septiembre, octubre y noviembre un descuento de 500 pesos sobre el total de la compra que realiza un cliente. Solicitar al usuario que ingrese un mes y el importe de la compra. El programa debe calcular cuál es el monto total que se debe cobrar al cliente e imprimirlo por pantalla.**

Algoritmo importe\_mes

//Definimos las variables a utilizar

definir mes Como Caracter

definir importe Como Real

definir importeFinal Como Real

//Pedimos al usuario que ingrese un mes y un importe

Escribir "Ingresar Mes: "

Leer mesIngresado

Escribir "Ingresar importe de la compra: "

Leer importeIngresado

//Asignamos los valores ingresados por consola a nuestras variables

mes = Minusculas(mesIngresado)

importe = importeIngresado

//Si el mes es igual a 'septiembre', 'octubre' o 'noviembre' se aplicará un descuento

si(mes == "septiembre" O mes == "octubre" O mes == "noviembre")Entonces

importeFinal = importe - 500

Escribir "Se aplicó $500 de descuento"

SiNo

importeFinal = importe

Escribir "No se aplicaron descuentos"

FinSi

Escribir "El importe final es de: ", importeFinal

FinAlgoritmo

1. **Solicitar al usuario que ingrese dos números enteros y determinar si ambos son pares o impares. Mostrar en pantalla un mensaje que indique “Ambos números son pares” siempre y cuando cumplan con la condición. En caso contrario se deberá imprimir el siguiente mensaje “Los números no son pares, o uno de ellos no es par”. Nota: investigar la función mod de Pseint.**

Algoritmo numeros\_pares

//Definimos las variables

definir numUno Como Entero

definir numDos Como Entero

//Le pedimos al usuario que ingrese ambos números por consola

escribir "Escribir el primer número: "

leer numUnoIngresado

escribir "Escribir el segundo número: "

leer numDosIngresado

//Asignamos los valores ingresados por consola a nuestras variables

numUno = numUnoIngresado;

numDos = numDosIngresado;

//Un número es par si el módulo (resto) de un número divido entre 2 da 0.

si(numUno MOD 2 == 0 Y numDos MOD 2 == 0)Entonces

//Si ambos números son par devolvemos "Ambos números son pares"

Escribir "Ambos números son pares"

SiNo

//Si no devolvemos "Los números no son pares, o uno de ellos no es par"

Escribir "Los números no son pares, o uno de ellos no es par"

FinSi

FinAlgoritmo

1. **Escriba un programa que pida 3 notas y valide si esas notas están entre 1 y 10. Si están entre esos parámetros se debe poner en verdadero una variable de tipo lógico y si no ponerla en falso. Al final el programa se debe decir si las 3 notas son correctas usando la variable de tipo lógico. 10**

Algoritmo notas\_correctas

//Definimos las variables

Definir notas Como Entero

//La variable notas la definimos como un ARREGLO y en la línea siguiente le asignamos una longitud

Dimension notas[3]

Definir esNotaCorrecta Como Logico

//Mediante una iteración PARA le pedimos al usuario que ingrese una nota que luego será asignada al ARREGLO notas

Para i<-1 Hasta 3 Con Paso 1 Hacer

Escribir "Dime la nota n° ", i

leer notaIngresada

notas[i] = notaIngresada

Fin Para

//Comprobamos que todas las notas sean mayor o igual a 1 y menor o igual a 10

si((notas[1] >= 1 Y notas[1] <= 10) Y (notas[2] >= 1 Y notas[2] <= 10) Y (notas[3] >= 1 Y notas[3] <= 10))entonces

//Si es así la variable esNotaCorrecta será verdadera

esNotaCorrecta = Verdadero

FinSi

si(esNotaCorrecta == Verdadero)Entonces

Escribir "Las 3 notas son correctas"

SiNo

Escribir "Una o más de las 3 notas es incorrecta"

FinSi

FinAlgoritmo

1. **Escriba un programa que pida una frase o palabra y valide si la primera letra de esa frase es una ‘A’. Si la primera letra es una ‘A’, se deberá de imprimir un mensaje por pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”. Nota: investigar la función Subcadena de Pseint.**

Algoritmo frase\_a

//Definimos la variable

definir frase Como Caracter

//Pedimos al usuario que ingrese por consola una frase

Escribir "Escribe una frase o una palabra: "

leer fraseIntroducida

//Asignamos el valor introducido por consola en nuestra variable 'frase' (el valor lo convertimos en minúscula para que sea más fácil comprobarlo)

frase = Minusculas(fraseIntroducida)

//Comprobamos si la frase que introdujo el usuario comienza con una letra A

si(SubCadena(frase,1,1) == "a")Entonces

Escribir "Correcto, la frase comienza con A"

SiNo

Escribir "INCORRECTO, la frase no comienza con A"

FinSi

FinAlgoritmo

1. **Continuando el ejercicio anterior, ahora se pedirá una frase o palabra y se validara si la primera letra de la frase es igual a la última letra de la frase. Se deberá de imprimir un mensaje por pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”.**

Algoritmo Ejercicio13

//Se pedirá una frase o palabra y se validara si la primera letra

//de la frase es igual a la última letra de la frase. Se deberá

//de imprimir un mensaje por pantalla que diga

//"CORRECTO", en caso contrario, se deberá imprimir "INCORRECTO".

//Declaración de variables

Definir frase, primer\_letra, ultima\_letra Como Caracter;

Escribir "Ingrese una palabra o frase: ";

Leer frase; //Asignamos el ingreso por teclado a la variable 'frase'

//Convertimos toda la frase a minuscula para una correcta comparacion

frase = Minusculas(frase);

//Tomamos la primera letra de 'frase' y la asiganmos a la variable 'primer\_letra'

primer\_letra = Subcadena(frase, 1, 1);

//Tomamos la última letra de 'frase' y la asignamos a la variable 'ultima\_letra'

ultima\_letra = Subcadena(frase, Longitud(frase), Longitud(frase));

//Comparamos las variables 'primer\_letra' y 'ultima\_letra'

Si primer\_letra == ultima\_letra Entonces

Escribir "CORRECTO"; //Acción si son iguales

SiNo

Escribir "INCORRECTO"; //Acción si no son iguales

FinSi

FinAlgoritmo

**14. La empresa “Te llevo a todos lados” está destinada al alquiler de autos y tiene un sistema de tarifa que consiste en cobrar el alquiler por hora. Si el cliente devuelve el auto dentro de las 2 horas de uso el valor que corresponde pagar es de $400 pesos y la nafta va de regalo. Cuando el cliente regresa a la empresa pasadas las 2 horas, se ingresan la cantidad de litros de nafta gastados y el tiempo transcurrido en horas. Luego, se le cobra 40 pesos por litro de nafta gastado, y la hora se fracciona en minutos, cobrando un total de $5,20 el minuto de uso. Realice un programa que permita registrar esa información y el total a pagar por el cliente.**

Algoritmo Ejercicio14

//Declaración de variables

Definir tiempo\_alquiler, litros, minutos, total Como Real;

Escribir "Ingrese el tiempo de alquiler del auto: ";

Leer tiempo\_alquiler; //Asignamos el tiempo ingresado a la variable 'tiempo\_alquiler'

Si tiempo\_alquiler <= 2 Entonces

total = 400; //Accion si el tiempo no supera las 2 horas

SiNo //Si el tiempo supera las 2 horas pedimos el ingreso de los litros de nafta consumidos

Escribir "Ingrese la cantidad de litros consumidos:";

Leer litros; //Asignamos la cantida ingrasada a la variable 'litros'

minutos = tiempo\_alquiler\*60; //Convertimos la cantidad de horas ingresadas a minutos

total = (litros\*40) + (minutos\*5.2); //Calculamos el total a pagar por el cliente

FinSi

//Mostramos el total a pagar por pantalla

Escribir "El total a pagar por el cliente es de $",total;

FinAlgoritmo

1. **Solicitar al usuario que ingrese un valor entre 1 y 7. El programa debe mostrar por pantalla un mensaje que indique a qué día de la semana corresponde. Considere que el número 1 corresponde al día “Lunes”, y así sucesivamente.**

Algoritmo Ejercicio15

//Declaración de variables

Definir num Como Entero;

Definir dia Como Caracter;

//Pedimos al usuario que ingrese un numero

Escribir "Ingrese un número entre 1 y 7:";

Leer num; //Asignamos el número ingresado a la variable 'num'

Escribir "El número ingresado corresponde al día:";

Segun num Hacer //Según el número ingresado se mostrará el día correspondiente por pantalla

1:

Escribir "Lunes";

2:

Escribir "Martes";

3:

Escribir "Miércoles";

4:

Escribir "Jueves";

5:

Escribir "Viernes";

6:

Escribir "Sábado";

7:

Escribir "Domingo";

De Otro Modo:

Escribir "El número ingresado está fuera del rango requerido."; //Mensaje que se mostrará en caso que el número no corresponda a ningún día

Fin Segun

FinAlgoritmo

1. **Construir un programa que simule un menú de opciones para realizar las cuatro operaciones aritméticas básicas (suma, resta, multiplicación y división) con dos valores numéricos enteros. El usuario, además, debe especificar la operación con el primer carácter de la operación que desea realizar: ‘S' o ‘s’ para la suma, ‘R’ o ‘r’ para la resta, ‘M’ o ‘m’ para la multiplicación y ‘D’ o ‘d’ para la división. REHACER CON SEGUN**

Algoritmo Ejercicio16

//Declaracion de variables

Definir num1, num2 Como Entero;

Definir operacion Como Caracter;

Definir total Como Real;

//Pedimos el ingreso de los números y asignamos a las variables 'num1' y 'num2'

Escribir "Ingrese el primer número a operar: ";

Leer num1;

Escribir "Ingrese el segundo número a operar: ";

Leer num2;

//Pedimos el ingreso de la inicial de la operacion

Escribir "Ingrese la inicial de la operación a realizar:";

Escribir "s para suma,";

Escribir "r para resta,";

Escribir "m para multiplicaciom y";

Escribir "d para division.";

Leer operacion; //Asignamos la letra ingresada a la variable 'operacion'

operacion = Minusculas(operacion); //Convertimos la variable 'operacion' a minusculas para trabajar con ella

//Segun la letra ingresada se realizará la operacion indicada, de no corresponder a ninguna operacion, se muestra un mensaje por pantalla

Si operacion == "s" Entonces

total = num1 + num2;

SiNo

Si operacion == "r" Entonces

total = num1 - num2;

Sino

Si operacion == "m" Entonces

total = num1 \* num2;

SiNo

Si operacion == "d" Entonces

total = num1 / num2;

Sino

Escribir "La letra ingresada no corresponde a ninguna operación."

FinSi

FinSi

FinSi

FinSi

//Mostramos el resultado obtenido por pantalla

Escribir "El resultado de la operación elegida entre ",num1," y ",num2," es ",total;

FinAlgoritmo

1. **Leer tres números que denoten una fecha (día, mes, año) y comprobar que sea una fecha válida. Si la fecha no es válida escribir un mensaje de error por pantalla. Si la fecha es válida se debe imprimir la fecha cambiando el número que representa el mes por su nombre. Por ejemplo: si se introduce 1 2 2006, se deberá imprimir “1 de febrero de 2006”.**

Algoritmo Ejercicio17

//Declaracion de variables

Definir dia, mes, anio Como Entero;

Definir mes\_letra Como Caracter;

Definir aux Como Logico;

aux = Falso;

//Pedimos al usuario que ingrese un numero para el dia, uno para el mes y uno para el año y los asignamos a las variables correspondientes

Escribir "Ingrese un número para el día:";

Leer dia;

Escribir "Ingrese un número para el mes:";

Leer mes;

Escribir "Ingrese un número para el año:";

Leer anio;

//Analizamos la validez de la fecha segun la cantidad de dias del mes

Si (mes==1 O mes ==3 O mes==5 O mes==7 O mes==10 O mes==12) Y (dia>0 Y dia<32) Entonces //Condicion meses de 31 dias

aux = Verdadero;

Sino Si mes==2 Y ((((anio % 4 == 0) Y (anio % 100 <> 0) O (anio % 400 == 0)) Y dia<30) O dia<29) Entonces //Condicion febrero, incluyendo si el año es bisiesto

aux = Verdadero;

Sino Si (mes==4 O mes==6 O mes==8 O mes==9 O mes==11) Y (dia>0 Y dia<31) Entonces //Condición meses de 30 días

aux = Verdadero;

FinSi

FinSi

FinSi

//Escribimos el mes en letras

Segun mes Hacer

1:

mes\_letra = "Enero";

2:

mes\_letra = "Febrero";

3:

mes\_letra = "Marzo";

4:

mes\_letra = "Abril";

5:

mes\_letra = "Mayo";

6:

mes\_letra = "Junio";

7:

mes\_letra = "Julio";

8:

mes\_letra = "Agosto";

9:

mes\_letra = "Septiembre";

10:

mes\_letra = "Octubre";

11:

mes\_letra = "Noviembre";

12:

mes\_letra = "Diciembre";

Fin Segun

//Si la fecha es válida la mostramos por pantalla, de lo contrario mostramos mensaje de fecha invalida

Si aux Entonces

Escribir dia, " de ",mes\_letra," de ",anio;

SiNo

Escribir "Fecha inexistente";

FinSi

FinAlgoritmo

1. **Realizar un programa que, dado un número entero, visualice en pantalla si es par o impar. En caso de que el valor ingresado sea 0, se debe mostrar “el número no es par ni impar” (para que un número sea par, se debe dividir entre dos y su resto debe ser igual a 0). Nota: investigar la función mod de PSeInt**

Algoritmo Ejercicio18

//Definicion de variables

Definir num Como Entero;

//Pedimos al usuario que ingrese un numero y lo asignamos a la variable 'num'

Escribir "Ingrese un número:";

Leer num;

//Analizamos el numero ingresado

Si num == 0 Entonces

Escribir "El número no es par ni impar"; //Accion si el número ingresado es 0

Sino

Si (num MOD 2) == 0 Entonces //Comprobamos si el numero es par, analizando el valor del resto de la division entre el numero y 2

Escribir "El número es par";

SiNo

Escribir "El número es impar";

FinSi

FinSi

FinAlgoritmo

1. **Realice un programa que, dado un año, nos diga si es bisiesto o no. Un año es bisiesto bajo las siguientes condiciones: Un año divisible por 4 es bisiesto y no debe ser divisible por 100. Si un año es divisible por 100 y además es divisible por 400, también resulta bisiesto. Nota: recuerde la función mod de PSeInt**

Algoritmo anio\_bisiesto

// Declaración de variables

Definir anio Como Entero

Definir esAnioBisiesto Como Logico

// Pedido de datos al "usuatio"

Escribir "Ingrese un año, sin comas o puntos"

Leer anioIngresado

// Asignamos el valor optenido por la consola a nuestra variable declarada

anio = anioIngresado

// Es año bisiesto, si es divisible por 4 y el resto da cero ademas no es divisible por 100.

Si ( anio mod 4 == Par Y anio mod 100 <> Par) Entonces

esAnioBisiesto = Verdadero

FinSi

// Es año bisiesto, si es divisible por 100 y ademas es divisible por 400

Si ( anio mod 100 == Par Y anio mod 400 == par)

esAnioBisiesto = Verdadero

FinSi

// Mostramos por consola si el numero ingresado es un año bisiesto

si ( esAnioBisiesto ) Entonces

Escribir "El año ingresado es bisiesto"

SiNo

Escribir "El año ingresado no es bisiesto"

FinSi

FinAlgoritmo

**20. Escriba un programa para obtener el grado de eficiencia de un operario de una fábrica de tornillos, de acuerdo a las siguientes dos condiciones que se le imponen para un período de prueba: ▪ Producir menos de 200 tornillos defectuosos. ▪ Producir más de 10000 tornillos sin defectos. El grado de eficiencia se determina de la siguiente manera: ▪ Si no cumple ninguna de las condiciones, grado 5. ▪ Si sólo cumple la primera condición, grado 6. ▪ Si sólo cumple la segunda condición, grado 7. 11 ▪ Si cumple las dos condiciones, grado 8 Nota: para trabajar este ejercicio de manera prolija, ir probando cada inciso que pide el ejercicio. No hacer todos al mismo tiempo y después probar.**

Algoritmo grado\_eficiencia

// Declaro variables

Definir tornillosDefectuosos Como Entero

Definir tornillosSinDefectos Como Entero

Definir eficiencia Como Entero

//Solicitud de datos

Escribir "Ingrese la cantidad de tornillo defectuosos"

Leer tornillosDefectuososConsola

Escribir "Ingrese la cantidad de tornillo sin defectos"

Leer tornillosSinDefectosConsola

//Asignar valores a la varibles declaradas

tornillosDefectuosos = tornillosDefectuososConsola

tornillosSinDefectos = tornillosSinDefectosConsola

// Verificar las condiciones para asignar un grado de eficiencia

// Primera comprobación : la cantidad de tornillos defectuoso es "menor" a 200 Y la cantidad

// de tornillos sin defectos es "mayor" a 10000

Si ( tornillosDefectuosos < 200 Y tornillosSinDefectos > 10000 ) Entonces

eficiencia = 8

FinSi

// Segunda comprobación : la cantidad de tornillos defectuoso es "menor" a 200 Y la cantidad

// de tornillos sin defectos es "menor o igual" a 10000

Si ( tornillosDefectuosos < 200 Y tornillosSinDefectos <= 10000 ) Entonces

eficiencia = 7

FinSi

// Tercera comprobación : la cantidad de tornillos defectuoso es "mayor o igual" a 200 Y la cantidad

// de tornillos sin defectos es "mayor" a 10000

Si ( tornillosDefectuosos >= 200 Y tornillosSinDefectos > 10000 ) Entonces

eficiencia = 6

FinSi

// Primera comprobación : la cantidad de tornillos defectuoso es "mayor o iual" a 200 Y la cantidad

// de tornillos sin defectos es "menor o igual" a 10000

Si (tornillosDefectuosos >= 200 Y tornillosSinDefectosConsola <= 10000 ) Entonces

eficiencia = 5

FinSi

// Mostramos por consola el grado de eficiencia del operario

Escribir "El grado de eficien del operario es : ", eficiencia

FinAlgoritmo

**21. Una empresa tiene personal de distintas áreas con distintas condiciones de contratación y formas de pago. El departamento de contabilidad necesita calcular los sueldos semanales (lunes a viernes) en base a las 3 modalidades de sueldo: a) comisión b) salario fijo + comisión, y c) salario fijo. a) Para la modalidad salario por comisión se debe ingresar el monto total de las ventas realizadas en la semana, y el 40% de ese monto total corresponde al salario del empleado. b) Para la condición de salario fijo + comisión, se debe ingresar el valor que se paga por hora, la cantidad de horas trabajadas semanalmente y el monto total de las ventas en esa semana. En este tipo de contrato las horas extras no están contempladas y se fija como máximo 40 horas por semana. La comisión por las ventas se calcula como 25% del valor de venta total. c) Finalmente, para la modalidad de salario fijo se debe ingresar el valor que se paga por hora y la cantidad de horas trabajadas en la semana. En el caso de exceder las 40 horas semanales, las horas extras se deben pagar con un extra del 50% del valor de la hora. Realizar un menú de opciones para poder elegir el tipo de contrato que tiene un empleado.**

Algoritmo empresa\_contratacion

//Declaro las variables a utilizar

Definir salario Como Real

Definir horasTrabajadas Como Entero

Definir montoPorHora Como Real

Definir horasExtra Como Entero

Definir montoPorHoraExtra Como Real

Definir totalVentas Como Real

Definir comision Como Real

//Obtener respuesta del usuario para calcular el salario

Escribir "Ingrese [ A ] para calcular el Salario por Comisión de venta"

Escribir "Ingrese [ B ] para calcular el Salario + Comisión"

Escribir "Ingrese [ C ] para calcular el Salario"

Leer respuesta

// Usamos estructura de control "Switch" o "Segun el caso" como alternativa a multipes "Si"

Segun respuesta Hacer

"A":

// Solicitamos nuevamente los datos al usuario

Escribir "Ingrese el monto total de ventas "

Leer montoTotalConsola

// Asignamos los valores dados por consola a nuestras variables

totalVentas = montoTotalConsola

// Calculamos la comision que es el cuarenta por ciento ( 40% o 0.4 ) del total de ventas

comision = totalVentas\*0.4

// Como la opcion elegida fue "A" el salario equivale al 40% de las ventas

salario = comision

// Mostramos por consola lo calculado

Escribir "El salario por comisiones fue del : $", salario

"B":

// Solicitamos nuevamente los datos al usuario

Escribir "Ingresa la cantidad de horas trabajadas"

Leer horasTrabajoConsola

Escribir "Ingrese el monto por hora trabajada"

Leer montoPorHoraConsola

Escribir "Ingrese el monto total de ventas "

Leer montoTotalConsola

// Asignamos los valores dados por consola a nuestras variables

totalVentas = montoTotalConsola

horasTrabajadas = horasTrabajoConsola

montoPorHora = montoPorHoraConsola

// Verificamos que el total de horas trabajadas sea "menor o igual" a cuarenta ( 40 hs )

Si ( horasTrabajadas <= 40) Entonces

// No hizo más de '40 hs'

//la comision por ventas es del 25%

comision = totalVentas\*0.25

// Como la opción elegida fue "B" el salario se calcula como :

// "horas trabajadas" por "monto a pagar por hora" más la comisión

salario = horasTrabajadas \* montoPorHora

salario = salario + comision

SiNo

// Supera las '40 hs' de trabajo, segun el enunciado del ejercicio para la opcion "B", las horas extras

// no se contemplan y se fija un máximo de 40 como referencia a '40 hs'

comision = totalVentas - ( totalVentas\*0.25 )

salario = 40 \* montoPorHora

salario = salrio + comision

FinSi

// Mostramos por consola lo calculado

Si ( horasTrabajadas <= 40) Entonces

Escribir "El salario + comisiones sin horas extras fue del : $", salario

SiNo

Escribir "El salario + comisiones con horas extras fue del : $", salario

FinSi

Caso "C":

// Solicitamos nuevamente los datos al usuario

Escribir "Ingresa la cantidad de horas trabajadas"

Leer horasTrabajoConsola

Escribir "Ingrese el monto por hora trabajada"

Leer montoPorHoraConsola

// Asignamos los valores dados por consola a nuestras variables

horasTrabajadas = horasTrabajoConsola

montoPorHora = montoPorHoraConsola

// Verificamos que el total de horas trabajadas sea "menor o igual" a cuarenta ( 40 hs )

Si ( horasTrabajadas <= 40) Entonces

// No hizo más de '40 hs'

// Como la opción elegida fue "C" el salario se calcula como :

// "horas trabajadas" por "monto a pagar por hora"

salario = horasTrabajadas \* montoPorHora

SiNo

// Supera las '40 hs' de trabajo, segun el enunciado del ejercicio para la opcion "C", las horas extras

// Si se contemplan

// Para calcular las horas extras a las horas trabajadas les restamos el valor de 40, ya que como dice el

// enunciado del ejercio el máximo permito de horas es 40, el resto se considera "horas extra"

horasExtra = horasTrabajadas - 40

// las horas extras se pagan con una bonificación del 50% sobre el monto por hora

montoPorHoraExtra = montoPorHora + montoPorHora\*0.5

// Finalmente calculamos el salario como las "horas trabajadas" por el "monto a cobra por hora"

salario = ( 40 \* montoPorHora )

// más las "horas extras" por el "monto de bonificación" del 50%

salario = salario + ( horasExtra \* montoPorHoraExtra )

FinSi

// mostramos por consola lo calculado

Si ( horasTrabajadas <= 40) Entonces

Escribir "El salario sin horas extras fue del : $", salario

SiNo

Escribir "El salario con horas extras fue del : $", salario

FinSi

FinSegun

FinAlgoritmo

**22. Escriba un programa en el cual se ingrese un número y mientras ese número sea mayor de 10, se pedirá el número de nuevo.**

Algoritmo nuevoNumeroEnBaseADiez

//Declaramos las variables a utilizar

Definir numeroAComparar Como Entero

// El algoritmo utiliza la estructura "Do While" o "Hacer Hasta Que", primero ejecuta el código y luego valida la

// la condición, de ser 'Falso' ejecutará de nuevo el bucle. De ser una condición "Verdadera" no ejecutará el bucle

// nuevamente

// Nos mostrará el mensaje "Ingrese un número" repetidas veces hasta que el mismo sea mayor a 10

Hacer

// Pedimos el numero al usuario

Escribir "Ingrese un número"

Leer numeroConsola

//asignamos el valor la variable previamente declarada

numeroAComparar = numeroConsola

Hasta Que ( numeroAComparar < 10 )

FinAlgoritmo

**23. Escriba un programa que valide si una nota está entre 0 y 10, sino está entre 0 y 10 la nota se pedirá de nuevo hasta que la nota sea correcta.**

Algoritmo notaCorrectaIntervalo

// Declaramos las variables a utilizar

Definir nota Como Entero

// El algoritmo utiliza la estructura "Do While" o "Hacer Hasta Que"

// Nos mostrará el mensaje "Ingrese una nota" repetidas veces hasta que el mismo sea "mayor" a 0 "Y" "menor" a 10

Hacer

// Pedimos al usuario una nota

Escribir "Ingrese una nota"

Leer notaConsola

// Asignamos los ingresado por consola a nuestras variables

nota = notaConsola

Hasta Que ( nota > 0 Y nota < 10 )

FinAlgoritmo

**24. Escriba un programa que solicite dos números enteros (mínimo y máximo). A continuación, se debe pedir al usuario que ingrese números enteros situados entre el máximo y mínimo. Cada vez que un número se encuentre entre ese intervalo, se sumara uno a una variable. El programa terminará cuando se escriba un número que no pertenezca a ese intervalo, y al finalizar se debe mostrar por pantalla la cantidad de números ingresados dentro del intervalo**.

Algoritmo maximoMinimo

// Declaramos las variables a utilizar

Definir maximo Como Entero

Definir minimo Como Entero

Definir numeroAComparar Como Entero

Definir contador Como Entero

//Pedimos al usuario el maximo y el minimo

Escribir "Ingrese un número como valor máximo"

Leer maximoConsola

//Pedimos al usuario el maximo y el minimo

Escribir "Ingrese un número como valor mínimo"

Leer minimoConsola

// Asignamos los valores dados

maximo = maximoConsola

minimo = minimoConsola

// Inicializamos nuestro contador

contador = 0

// El algoritmo utiliza la estructura "Do While" o "Hacer Hasta Que" primero ejecuta el codigo y despues verifica

// la condición o expresión booleana, de ser "Falso" (el resultado de la expresión) se ejecuta el codigo nuevamente

// de ser "Verdadero" ejecutara el codigo dentro del bucle y lo terminara

Hacer

// Pedimos al usuario un numero

Escribir "Ingrese un numero"

Leer numeroConsola

// Asignamos los ingresado por consola a nuestras variables

numeroAComparar = numeroConsola

// cada vez que ingresemos un numero que cumpla con el intervalo aumentará el contador uno en uno

contador = contador + 1

// con el operaso " ! " invertimos un valor lógico

Hasta Que ( !( numeroAComparar > minimo Y numeroAComparar < maximo ) )

// Mostramos por consola el contador

Escribir "La cantidad de números ingresados fue de : ", contador

FinAlgoritmo

**25. Escriba un programa en el cual se ingrese un valor límite positivo, y a continuación solicite números al usuario hasta que la suma de los números introducidos supere el límite inicial.**

Algoritmo limitePositivo

//Declaramos las variables

Definir acumulador Como Entero

Definir limiteMaximo Como Entero

Definir numeroSolicitado Como Entero

// Solicitamos el valor máximo al usuario

Escribir "Ingrese el valor máximo"

Leer limiteMaximoConsola

//Asignamos el limite a la variable

limiteMaximo = limiteMaximoConsola

//Inicializamos el acumulador

acumulador = 0

// Utilizamos la estructura "While" o "Miestras que" primero evalua la condición "buleana" y de arrojar un valor

// "Verdadero" ejecuta el codigo, en caso de ser una condición con valor 'Falso' termina el bucle

// sin ejecutar el código

Mientras ( acumulador <= limiteMaximo) Hacer

// Pedimos al usuario un numero

Escribir "Ingrese un numero"

Leer numeroConsola

// asignamos el valor a la variable declarada

numeroSolicitado = numeroConsola

// sumamos al acumular el valor actual más el nuevo valor

acumulador = acumulador + numeroSolicitado

FinMientras

// Mostramos por consola los resultados obtenidos

Escribir "Se ha superado el limite"

Escribir "Limite: ", limiteMaximo, " Acumulador: ",acumulador

FinAlgoritmo

**26. Dada una secuencia de números ingresados por teclado que finaliza con un –1, por ejemplo: 5,3,0,2,4,4,0,0,2,3,6,0,……,-1; realizar un programa que calcule el promedio de los números ingresados. Suponemos que el usuario no insertará número negativos.**

Algoritmo Ejercicio\_26

//Definimos las variables a utilizar.

// Num por numero ingresado

Definir num Como Entero;

//Sum por la sumatoria de los numeros ingresados

Definir sum Como Entero;

//Cont por el contador de numeros ingresados.

Definir cont Como Entero;

//Prom por la division de la sumatoria por la cantidad de numeros ingresados. Este sera un numero real es decir que puede contener decimales

Definir prom Como Real;

//Inicializamos en 0 las variables que lo necesitan.

sum = 0;

cont = 0;

//Bucle para ir agregando y sumando los numeros ingresados.

Hacer

Escribir "Ingrese el numero a agregar. Escriba -1 para terminar la secuencia"

Leer numConsola;

num = numConsola;

Si num <> -1 Entonces

sum = sum + num

cont = cont + 1;

FinSi

Hasta Que(num == -1)

//Calculamos el promedio

prom = sum / cont;

//Retornamos el valor

Escribir "El resultado del promedio de los numeros ingresados es: ", prom;

FinAlgoritmo

**27. Calcular las calificaciones de un grupo de alumnos. La nota final de cada alumno se calcula según el siguiente criterio: la parte práctica vale el 10%; la parte de problemas vale el 50% y la parte teórica el 40%. El programa leerá el nombre del alumno, las tres notas obtenidas, mostrará el resultado por pantalla, y a continuación volverá a pedir los datos del siguiente alumno hasta que el nombre sea una cadena vacía. Las notas deben estar comprendidas entre 0 y 10, y si no están dentro de ese rango no se imprimirá el promedio y se mostrará un mensaje de error. 12**

Algoritmo Ejercicio\_27

//Definimos las variables que se ingresaran.

Definir nombre Como Caracter;

Definir notaPractica Como Entero;

Definir notaProblemas Como Entero;

Definir notaTeoria Como Entero;

Definir notaFinal Como Real;

//Este bucle se repetira hasta que el nombre ingresado sea "", lo que indica que el usuario ya termino de usar el programa

Hacer

//Alamacenmos cada dato en su variable

Escribir "Ingrese el nombre del alumno. Si quiere terminar de usar el programa no escriba nada y pulse la tecla Enter";

Leer nombreIngresado;

nombre = nombreIngresado;

Escribir "Ingrese la nota practica.";

Leer notaPracticaIngresada;

notaPractica = notaPracticaIngresada;

Escribir "Ingrese la nota de resolucion de problemas.";

Leer notaProblemasIngresada;

notaProblemas = notaProblemasIngresada;

Escribir "Ingrese la nota teoria.";

Leer notaTeoriaIngresada;

notaTeoria = notaTeoriaIngresada;

//Aqui validamos si una de las notas ingresadas no es valida, es decir menor a 1 y mayor a 10

//Si una de las notas ingresadas es menor a 1 o mayor a 10 va a escribir el mensaje de Error correspondiente

Si notaTeoria <= 10 y notaTeoria >= 1

Si notaPractica <= 10 y notaPractica >= 1

Si notaProblemas <= 10 y notaProblemas >= 1

notaFinal = (notaPractica \* 0.10) + (notaProblemas \* 0.50) + (notaTeoria \* 0.40);

Escribir "El promedio del alumno ", nombre , " es de ", notaFinal;

SiNo

Escribir "ERROR: La nota de Resolucion de Problemas no es valida. Recuerde que deben ser valores del 1 al 10";

FinSi

SiNo

Escribir "ERROR: La nota de Practicas no es valida. Recuerde que deben ser valores del 1 al 10";

FinSi

SiNo

Escribir "ERROR: La nota de Teoria no es valida. Recuerde que deben ser valores del 1 al 10";

FinSi

//Este ultimo Escribir es para dejar una linea en blanco entre cada alumno

Escribir " ";

Hasta Que nombre == "";

**FinAlgoritmo**

**28. Escribir un programa que calcule cuántos dígitos tiene un número entero positivo (pista: se puede hacer dividiendo varias veces entre 10). Nota: investigar la función trunc().**

Algoritmo Ejercicio\_28

//Definimos las variables

Definir contador Como Entero;

Definir num Como Real;

//Utilizamos mostrador para almacenar el numero por segunda vez, ya que num va a ser editado en el bucle Mientras

Definir mostrador Como Real;

//Inicializamos el contador en 0, para que pueda ir aumentado su valor en cada iteracion del bucle Mientras

contador = 0;

//Le pedimos al usuario el numero con el que vamos a trabajar

//Mediante este bucle Hacer nos aseguramos que el numero ingresado sera positivo.

Hacer

Escribir "Ingrese el numero para que el programa cuente la cantidad de digitos: "

Leer numeroIngresado;

num = numeroIngresado

Si num <= 0

Escribir "Este programa solo funciona con numeros positivos, pruebe de nuevo"

FinSi

Hasta Que num >= 1;

//Una vez que el numero es valido, asignamos su valor a mostrador para cuando devolvamos el resultado

mostrador = num;

//Mientras que el numero ingresado sea mayor o igual a 1, este sera dividido por 10, haciendo que le quite un digito por iteracion

//En cada iteracion el contador aumenta en 1, dandonos como resultado el total de digitos que tenia el numero.

Mientras num >= 1 Hacer

num = num / 10;

contador = contador + 1;

Fin Mientras

//Devolvemos el resultado

Escribir "La cantidad de digitos de ", mostrador," es ",contador;

FinAlgoritmo

**29. Teniendo en cuenta que la clave es “eureka”, escribir un programa que nos pida ingresar una clave. Sólo se cuenta con 3 intentos para acertar, si fallamos los 3 intentos se deberá mostrar un mensaje indicándonos que hemos agotado esos 3 intentos. Si acertamos la clave se deberá mostrar un mensaje que indique que se ha ingresado al sistema correctamente.**

Algoritmo Ejercicio\_29

Definir clave Como Caracter;

clave = "eureka";

Escribir "Ingrese la contraseña: ";

//PREGUNTAR POR BREAK. ME GUSTARIA USAR EL BREAK PARA CORTAR EL BUCLE UNA VEZ QUE LA PW ES CORRECTA

Para iterador<-3 Hasta 1 Con Paso iterador-1 Hacer

Escribir "Tiene ", iterador, " intentos";

Leer claveIngresada

Si claveIngresada == clave

Escribir "Clave correcta. Bienvenido Arquímedes de Siracusa";

SiNo

Escribir "Clave Incorrecta, pruebe otra vez"

FinSi

Fin Para

FinAlgoritmo

**30. Se debe realizar un programa que: 1º) Pida por teclado un número (entero positivo). 2º) Pregunte al usuario si desea introducir o no otro número. 3º) Repita los pasos 1º y 2º mientras que el usuario no responda n/N (no). 4º) Muestre por pantalla la suma de los números introducidos por el usuario.**

Algoritmo Ejercicio\_30

Definir continuar Como Logico;

Definir acumulador Como Entero;

//Inicializamos como Verdadero a continuar, para que de por lo menos una vuelta por el bucle.

continuar = Verdadero;

//Inicializamos el acumulador a 0 para poder ir sumando los numeros ingresados

acumulador = 0;

Mientras continuar == Verdadero Hacer

Escribir "Ingrese un numero. En caso de querer terminar de agregar numeros, escriba N";

Leer ingreso

//Si el usuario ingresa N o n, siempre vamos a almacenar N ya que con la funcion Mayusculas transformamos la cadena a mayuscula

ingreso = Mayusculas(ingreso);

//Si ingresa N continuar pasa a ser falso y el bucle deja de ejecutarse

Si ingreso == "N"

continuar = Falso;

//Si no ingresa N, se espera que el usuario este ingresando un numero para sumar al acumulador

SiNo

//Aqui determinamos que el numero ingresado es POSITIVO y ENTERO

//Convertimos la cadena ingreso a Numero gracias a la funcion ConvertirANumero

//Determinamos que el numero sea entero gracias a la funcion trunc

Si ConvertirANumero(ingreso) > 0 y ConvertirANumero(ingreso) - trunc(ConvertirANumero(ingreso))== 0

acumulador = acumulador + ConvertirANumero(ingreso);

//El bucle sigue, pero el numero al ser invalido no se suma a la acumulacion

SiNo

Escribir "El numero debe ser entero y positivo";

FinSi

FinSi

Fin Mientras

//Devolvemos la suma de todos los numeros ingresados validos.

Escribir "La suma de todos los numeros ingresados es: ", acumulador;

FinAlgoritmo

**31. Se pide escribir un programa que calcule la suma de los N primeros números pares. Es decir, si ingresamos el número 5 como valor de N, el algoritmo nos debe realizar la suma de los siguientes valores: 2+4+6+8+10.**

Algoritmo Ejercicio\_31

Definir num Como Entero;

Definir acum Como Entero;

acum = 0;

Escribir "Ingrese la cantidad de numeros pares que se va a sumar"

Leer ingreso;

num = ingreso;

//Inicializamos el iterador del bucle en 1, para omitir el valor 0 al mostrar datos.

//num es el limite de veces que se va a ejecutar el bucle

//Al terminar cada iteracion el iterador aumenta en 1

Para iterador <- 1 Hasta num Con Paso iterador + 1 Hacer

//Mostramos los numeros pares obtenidos

Escribir iterador \* 2;

//Vamos acumulando cada uno de estos numeros

acum = acum + iterador \* 2;

Fin Para

//Retornamos el valor solicitado

Escribir "La suma de los valores pares obtenidos es: ", acum;

FinAlgoritmo

**32. Escribir un programa que lea números enteros hasta teclear 0 (cero). Al finalizar el programa se debe mostrar el máximo número ingresado, el mínimo, y el promedio de todos ellos.**

Algoritmo Ejercicio32

//Acumulador de la sumatoria de números ingreasados

acum = 0

contador = 0

//Contador de la cantidad de veces que ingresamos números distintos de 0

//Numero para ingresar al bucle

Escribir "Escribir un número (Para terminar la ejecucíon escribir 0)"

Leer numOriginal

Si numOriginal = 0 Entonces

contador = 1

FinSi

//En una primera instancia se asignan al numero ingresado

//El numero mas grande ingresado

max = numOriginal

//El numero mas pequeñ ingresado

min = numOriginal

//Accede al bucle mientras el numero es distinto de 0

Mientras numOriginal <> 0 Hacer

//Sumatoria de numeros

acum = acum+numOriginal

Escribir "Escribir un número (Para terminar la ejecucíon escribir 0)"

Leer numBucle

//Si el num ingresado en el bucle es mayor al maximo se asigna un nuevo maximo

Si numBucle > max Entonces

max = numBucle

FinSi

//Si el num ingresado en el bucle es distinto de 0 y es menor al mínimo se asigna un nuevo minimo

Si numBucle <> 0 & numBucle < min Entonces

min = numBucle

FinSi

//Aumentar contador para obtener la cantidad de números ingresados

contador = contador+1

//Asignar el nùmero ingresado en el bucle al numeroOriginal

numOriginal = numBucle

FinMientras

//Salida: Muestra el promedio, el máximo y el mínimo

Escribir "Promedio: ", acum/contador

Escribir "Número máximo: ", max

Escribir "Número mínimo: ",min

FinAlgoritmo

**33. Programar un juego donde la computadora elige un número al azar entre 1 y 10, y a continuación el jugador tiene que adivinarlo. La estructura del programa es la siguiente: 1º) El programa elige al azar un número n entre 1 y 10. 2º) El usuario ingresa un número x. 3º) Si x no es el número exacto, el programa indica si n es más grande o más pequeño que el número ingresado. 4º) Repetimos desde 2) hasta que x sea igual a n. El programa tiene que imprimir los mensajes adecuados para informarle al usuario qué hacer y qué pasó hasta que adivine el número. NOTA: Para generar un número aleatorio entre 1 y 10 se puede utilizar la función Aleatorio(límite inferior, límite superior) de PSeInt.**

Algoritmo Ejercicio33

Escribir "Encontrar el nùmero aleatorio entre 1 y 10"

//Asignar un número aleatorio entre 1 y 10 con la función Aleatorio(límite inferior, límite superior)

numAleatorio = Aleatorio(1,10)

//Repetir hasta que número ingresado sea igual al número aleatorio

Repetir

Escribir "Escribir un número"

//Entrada: Número que ingresa el usuario

Leer num

//Si el número ingresado es mayor al número aleatorio

Si num > numAleatorio Entonces

Escribir "El número aleatorio es mas pequeño"

SiNo

//Si el número ingresado es menor al número aleatorio

Si num < numAleatorio Entonces

Escribir "El número aleatorio es mas grande"

FinSi

FinSi

Hasta Que num = numAleatorio

//Salida: Muestra que el usuario adivinó el número

Escribir "Felicidades el número es: ",numAleatorio

FinAlgoritmo

**34. Escribir un programa que calcule el cuadrado de los 9 primeros números naturales e imprima por pantalla el número seguido de su cuadrado. Ejemplo: “2 elevado al cuadrado es igual a 4”, y así sucesivamente.**

Algoritmo Ejercicio34

//Hacer mientras el iterador sea menor a 9

Mientras iterador< 9 Hacer

Escribir 'Ingresar número natural nro: ',iterador+1

//Entrada: Nùmero natural

Leer numeroIngresado

//Si el número ingresado es menor a 0 no es natural

Si numeroIngresado < 0 Entonces

Escribir 'Por favor ingrese numeros naturales"

Sino

//Aumentar contador para seguir iterando

iterador = iterador+1

//Salida: Muestra el número elevado al cuadrado

Escribir numeroIngresado, ' elevado al cuadrado es igual a ', numeroIngresado^2

FinSi

FinMientras

FinAlgoritmo

**35. Realizar un programa que muestre la cantidad de números que son múltiplos de 2 o de 3 comprendidos entre 1 y 100.**

Algoritmo Ejercicio35

//Asignar variable para recorrer el bucle

iterador = 0

//Contador de multiplos de 3

multiplos3 = 0

//Contador de multiplos de 2

multiplos2 = 0

//Hacer mientras que iterador sea igual Y menor a 1000

Mientras iterador <= 100 Hacer

//Si el resto de dividir un número en 2 da 0 entonces es múltiplo de 2

//Para obtener el resto de una división usar dividendo%divisor (ej: 12%4 = 0)

Si iterador%2 = 0 Entonces

//Aumentar contador de múltiplos de 2

multiplos2 = multiplos2+1

FinSi

//Si el resto de dividir un número en 3 da 0 entonces es múltiplo de 3

Si iterador%3 = 0 Entonces

//Aumentar contador de múltiplos de 3

multiplos3 = multiplos3+1

FinSi

//Aumentar contador para seguir iterando

iterador = iterador+1

FinMientras

//Salida: Muestra la cantidad de números multiplos de 2 y 3 entre 1 y 100

Escribir "Cantidad de números que son múltiplos de 2 entre 1 y 100: ", multiplos2

Escribir "Cantidad de números que son múltiplos de 3 entre 1 y 100: ", multiplos3

FinAlgoritmo

**36. Realizar un programa que pida una frase y el programa deberá mostrar la frase con un espacio entre cada letra. La frase se mostrará así: H o l a. Nota: recordar el funcionamiento de la función Subcadena(). NOTA:. En PSeInt, si queremos escribir sin que haya saltos de línea, al final de la operación “escribir” escribimos “sin saltar”. Por ejemplo: 13 Escribir sin saltar “Hola, ” Escribir sin saltar “cómo estás?” Imprimirá por pantalla: Hola, cómo estás?**

Algoritmo Ejercicio36

Escribir "Ingrese una frase"

//Entrada: La frase a separar

Leer fraseIngresada

//Recorrer la frase desde el primer caracter hasta el ultimo con un bucle, para obtener el total de caracteres de la frase

//Se usa Longitud(cadena)

Mientras iterador <= Longitud(fraseIngresada) Hacer

//Subcadena nos permite obtener caracteres desde una cadena con las posiciones de inicio y fin

//Recibe tres parametros (cadena, posicionInicial, posicionFinal). Mas info: https://pseint.site/funciones/funcion-subcadena.php

//Salida: La frase ingresada pero separada por un espacio en blanco " "

Escribir Subcadena(fraseIngresada,iterador,iterador)," " Sin Saltar

//Aumentar contador para seguir iterando

iterador = iterador+1

FinMientras

FinAlgoritmo

**37. Siguiendo el ejercicio anterior, ahora deberemos hacer lo mismo pero que la cadena se muestre al revés. Por ejemplo, si tenemos la cadena: Hola, deberemos mostrar a l o H.**

Algoritmo Ejercicio37

Escribir "Ingrese una frase"

//Entrada: La frase a separar y a revertir

Leer fraseIngresada

//Asignar a variable iterador el total de caracteres

iterador<-Longitud(fraseIngresada)

//Recorrer la frase desde para atras con un bucle, para obtener el total de caracteres de la frase

//Se usa Longitud(cadena) para obtener el total de caracteres en la frase

//Hacer mientras iterador es mayor a 0 ya que la posicion inicial del primer caracter de una cadena es 0

Mientras iterador > 0 Hacer

//Subcadena nos permite obtener caracteres desde una cadena con las posiciones de inicio y fin

//Recibe tres parametros (cadena, posicionInicial, posicionFinal). Mas info: https://pseint.site/funciones/funcion-subcadena.php

//Salida: La frase ingresada y al reves pero separada por un espacio en blanco " "

Escribir Subcadena(fraseIngresada,iterador,iterador)," " Sin Saltar

//Disminuir en uno el contador para iterar la cadena al reves

iterador = iterador-1

FinMientras

FinAlgoritmo

**38. Un docente de Programación tiene un listado de 3 notas registradas por cada uno de sus N estudiantes. La nota final se compone de un trabajo práctico Integrador (35%), una Exposición (25%) y un Parcial (40%). El docente requiere los siguientes informes claves de sus estudiantes: ▪ Nota promedio final de los estudiantes que reprobaron el curso. Un estudiante reprueba el curso si tiene una nota final inferior a 6.5 ▪ Porcentaje de alumnos que tienen una nota de integrador mayor a 7.5. ▪ La mayor nota obtenida en las exposiciones. ▪ Total de estudiantes que obtuvieron en el Parcial entre 4.0 y 7.5. El programa pedirá la cantidad de alumnos que tiene el docente y en cada alumno pedirá las 3 notas y calculará todos informes claves que requiere el docente. Nota: para trabajar este ejercicio de manera prolija, ir probando cada inciso que pide el ejercicio. No hacer todos al mismo tiempo y después probar.**

**39. Realizar un programa que lea un número entero (tamaño del lado) y a partir de él cree un cuadrado de asteriscos de ese tamaño. Los asteriscos sólo se verán en el borde del cuadrado, no en el interior. Por ejemplo, si se ingresa el número 4 se debe mostrar: \* \* \* \* \* \* \* \* \* \* \* \* Nota: Recordar el uso del escribir sin saltar en Pseint.**

**40. Escriba un programa que lea un número entero (altura) y a partir de él cree una escalera invertida de asteriscos con esa altura. Por ejemplo, si ingresamos una altura de 5 se deberá mostrar: \*\*\*\*\*** \*\*\*\* \*\*\* \*\* \*

**41. Una compañía de seguros tiene contratados a n vendedores. Cada vendedor realiza múltiples ventas a la semana. La política de pagos de la compañía es que cada vendedor recibe un sueldo base más un 14 10% extra por comisiones de sus ventas. El gerente de la compañía desea saber por un lado, cuánto dinero deberá pagar en la semana a cada vendedor por concepto de comisiones de las ventas realizadas, y por otro lado, cuánto deberá pagar a cada vendedor como sueldo total (sueldo base + comisiones).**

**42. La función factorial se aplica a números enteros positivos. El factorial de un número entero positivo (!n) es igual al producto de los enteros positivos desde 1 hasta n: n! = 1 \* 2 \* 3 \* 4 \* 5 \* (n-1) \* n Escriba un programa que calcule los factoriales de todos los números enteros desde el 1 hasta el 5. El programa deberá mostrar la siguiente salida: !1 = 1 !2 = 1\*2 = 2 ... !5 = 1\*2\*3\*4\*5 = 120**