**Clase 12 - 13 - 14**

Algoritmo ejercicioTeoValorVSreferencia

Definir num Como Entero

num = 2

Escribir num

Escribir "Ahora enviamos el número a la función por valor y el resultado es:"

elevarAlCuadradoPorValor(num)

Escribir num

Escribir "\*\*\*\*\*\*\*\*\*\*\*"

Escribir "Ahora enviamos el número a la función por referencia y el resultado es:"

elevarAlCuadradoPorReferencia(num)

Escribir num

FinAlgoritmo

SubProceso elevarAlCuadradoPorValor(num Por Valor)

num = num \* num

FinSubProceso

SubProceso elevarAlCuadradoPorReferencia(num Por Referencia)

num = num \* num

FinSubProceso

//Realiza una función llamada Cooperar que reciba dos variables de tipo carácter, una variable debe

//contener el mensaje "Cooperando" y la otra "trabajamos mejor". La función debe concatenar ambos textos.

Algoritmo ejercicioTeoCooperar

Definir p1, p2, resultado Como Caracter

p1 = "Cooperando "

p2 = "trabajamos mejor "

resultado = Cooperar (p1,p2)

Escribir resultado

FinAlgoritmo

Funcion retorno <- Cooperar ( p1 Por Referencia, p2 Por Referencia )

Definir retorno Como Caracter

retorno=Concatenar(p1,p2)

Fin Funcion

Algoritmo ejercicioTeoDeteccionError

FinAlgoritmo

Funcion retorno <- Paridad (num Por Referencia)

retorno = num MOD 2 == 0

Fin Funcion

Funcion retorno <- Comparar ( num1 Por Referencia, num2 Por Referencia )

Definir retorno Como Logico

retorno = (num1> num2)

FinFuncion

Algoritmo Prueba

Definir num1, num2 Como Entero

Definir resultado Como Logico

num1 = 3

num2 = 6

resultado = Comparar(num1,num2)

Escribir "El num1 es mayor a num2, esta afirmación es: " resultado

FinAlgoritmo

//1. Realizar una función que calcule la suma de dos números. En el algoritmo principal le pediremos

// al usuario los dos números para pasárselos a la función. Después la función calculará la suma

// y lo devolverá para imprimirlo en el algoritmo.

Algoritmo ejercicio1

Definir val1, val2, total Como Entero

Escribir "Ingrese dos valores enteros "

Leer val1

Leer val2

total=suma(val1,val2)

Escribir "El resultado es ", total

FinAlgoritmo

Funcion resultado <- suma ( val1 Por Referencia, val2 Por Referencia )

Definir resultado Como Entero

resultado=(val1+val2)

Fin Funcion

Funcion retorno<-sumar(x,n)

Definir retorno Como Real

retorno<-x+n

FinFuncion

Algoritmo ej1Lucas

Definir num1,num2 Como Real

Escribir "Ingrese dos numeros"

Leer num1,num2

Escribir sumar(num1,num2)

FinAlgoritmo

//2. Realizar una función que valide si un número es impar o no. Si es impar la función debe devol-

//ver un verdadero, si no es impar debe devolver falso. Nota: la función no debe tener mensajes

//que digan si es par o no, eso debe pasar en el Algoritmo.

Algoritmo ejercicio2

Definir num Como Entero

Definir re Como Logico

Escribir "Ingrese un numero "

Leer num

re=parImpar(num)

Escribir "El numero ingresado es ", re

FinAlgoritmo

Funcion validar <- parImpar ( n )

Definir validar Como Logico

validar = (n MOD 2) <> 0

Fin Funcion

//Crea una función EsMultiplo que reciba los dos números pasados por el usuario, validando

//que el primer número múltiplo del segundo y devuelva verdadero si el primer número es múlti-

//plo del segundo, sino es múltiplo que devuelva falso.

Algoritmo ejercicio3

Definir num1,num2 Como Entero

definir resultado como logico

Escribir "Escribir un número:"

Leer num1

Escribir "Escribir otro número:"

Leer num2

resultado=EsMultiplo(num1,num2)

Escribir resultado

FinAlgoritmo

Funcion retorno <- EsMultiplo(num1,num2)

Definir retorno Como Logico

retorno = num1 MOD num2 = 0

FinFuncion

Algoritmo ejercicio4

Definir frase, letra Como Cadena

Escribir "ingrese una frase"

Leer frase

Escribir "ingrese una letra a buscar en la frase"

Leer letra

Escribir "Se encontró " buscar(frase,letra) " vez/veces la letra " letra

FinAlgoritmo

Funcion contar<-buscar(x,c)

Definir i, contar, suma Como Entero

contar<-0

Para i<-0 Hasta Longitud(x) Hacer

Si Mayusculas(SubCadena(x,i,i)) = Mayusculas(c) Entonces

contar<-contar+1

FinSi

Fin Para

FinFuncion

Algoritmo ejercicio4Lucas

definir frase, letra como cadena

escribir "ingrese una frase"

leer frase

escribir "ingrese una letra a buscar en la frase"

leer letra

escribir "Se encontró " buscar(frase,letra) " vez/veces la letra " letra

FinAlgoritmo

Funcion contar<-buscar(x,c)

Definir i, cont como entero

cont<-0

Para i<-0 Hasta Longitud(x) Hacer

si Mayusculas(SubCadena(x,i,i)) = Mayusculas(c) entonces

cont<-cont+1

FinSi

Fin Para

FinFuncion

//5. Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número es

// primo o no. Un número es primo cuando es divisible sólo por 1 y por sí mismo, por ejemplo: 2,

// 3, 5, 7, 11, 13, 17, etc. Nota: recordar el uso del MOD.

Algoritmo ejercicio5

Definir num Como Entero

Definir re Como Logico

Escribir "Ingrese un numero entero "

Leer num

re=EsPrimo(num)

Escribir "El numero ingresado es primo. Esto es ", re

FinAlgoritmo

Funcion pnop <- EsPrimo ( n )

Definir pnop Como Logico

Definir i Como Entero

Para i = 2 Hasta n-1 Con Paso 1 Hacer

Si (n MOD i <> 0 ) Entonces

pnop = Verdadero

Sino //alguna de las divisiones dio resto cero ya no es primo

pnop = Falso

i = n-1

FinSi

Fin Para

Fin Funcion

////6. Realizar una función que calcule y retorne la suma de todos los divisores del número n distintos

//// de n. El valor de n debe ser ingresado por el usuario.

Algoritmo ejercicio6

Definir num,x Como Entero

Escribir "Ingrese numero "

Leer num

x=divisor(num)

Escribir "La suma de los divisores ", x

FinAlgoritmo

Funcion z<-divisor(num)

Definir z, i Como Entero

z=0

Para i<-1 Hasta (num -1) Hacer

Si num MOD i=0

z=z+i

FinSi

FinPara

FinFuncion

//7. Diseñar una función que reciba un numero en forma de cadena y lo devuelva como numero

// entero. El programa podrá recibir números de hasta 3 dígitos. Nota: no poner números con

// decimales ni letras. Ejemplo: ingresando "100"(carácter) debe convertirse en 100(entero).

Algoritmo ejercicio7

Definir num Como Cadena

Definir long, r Como Entero

Escribir "Ingrese un número de hasta 3 cifras"

Leer num

Si Longitud(num)>=1 y Longitud(num)<=3 Entonces

Escribir Convertir(num)

FinSi

r=Convertir(num)

Escribir r \* 2

FinAlgoritmo

Funcion resultado = Convertir(num)

Definir resultado Como Entero

resultado = ConvertirANumero(num)

FinFuncion

//8.Crear una función llamada "Login", que recibe un nombre de usuario y una contraseña y que

// devuelve Verdadero si el nombre de usuario es "usuario1" y si la contraseña es "asdasd".

// Además, la función calculara el número de intentos que se ha usado para loguearse, tenemos solo

// 3 intentos, si nos quedamos sin intentos la función devolverá Falso.

Funcion res <- login ( us, ct, i )

Definir res Como Logico

Si (us="usuario1" Y ct="asdasd" Y i=4) Entonces

res=Verdadero

Sino

Si (us="usuario1" Y ct="asdasd") O i>=3 Entonces

res=Falso

FinSi

Finsi

Fin Funcion

Algoritmo ejercicio8

Definir us, ct Como Caracter

Definir res Como Logico

Definir i, x Como Entero

Escribir "Ingrese usuario"

Leer us

Escribir "Ingrese contraseña"

Leer ct

i=0

res=login(us, ct, i)

Para i<-0 Hasta 3 Con Paso 1 Hacer

Escribir "Resultado incorrecto"

Escribir "Ingrese su usuario"

Leer us

Escribir "Ingrese su clave"

Leer ct

i=i+1

res=login(us, ct, i)

Si res=Verdadero

Escribir " Verdadero Ingreso Exitoso "

i=4

Sino

res=Falso

Escribir " Falso "

i=i+1

Finsi

FinPara

FinAlgoritmo

//9. Los empleados de una fábrica trabajan en dos turnos: Diurno y Nocturno. Se desea calcular el

// jornal diario de acuerdo con las siguientes reglas:

// a) La tarifa de las horas diurnas es de $ 90

// b) La tarifa de las horas nocturnas es de $ 125

// c) En caso de ser feriado, la tarifa se incrementa en un 10% si el turno es diurno y en

// un 15% si el turno es nocturno.

// El programa debe solicitar la siguiente información al usuario: el nombre del trabajador, el día

// de la semana, el turno (diurno o nocturno) y la cantidad de horas trabajadas. Además, debemos

// preguntarle al usuario si el día de la semana (lunes, martes, miércoles, etc.) era festivo o

// no, para poder calcular el jornal diario. Utilice una función para realizar el cálculo.

Funcion calculo <- JornalDiario (turno,feriado,canhs)

Definir calculo Como Real

Si turno="diurno" Entonces

Si feriado=="s" Entonces

calculo=(canhs\*90)\*1.1

Sino

Si feriado=="n" Entonces

calculo=canhs\*90

FinSi

FinSi

FinSi

Si turno="nocturno" Entonces

Si feriado=="s" Entonces

calculo=(canhs\*125)\*1.15

SiNo

Si feriado=="n" Entonces

calculo=canhs\*125

FinSi

FinSi

FinSi

Fin Funcion

Algoritmo ejercicio9

Definir canhs, pago Como Real

Definir nom, dia, turno, feriado Como Caracter

Escribir "Ingrese el nombre del empleado "

Leer nom

Escribir "Ingrese el turno. diurno/nocturno "

Leer turno

Escribir "Ingrese el dia de la semana "

Leer dia

Escribir "Indique si fue festino. s/n "

Leer feriado

Escribir "Ingrese la cantidad de hs trabajadas "

Leer canhs

pago=JornalDiario(turno,feriado,canhs)

Escribir "El empleado ", nom " recibira como pago ", pago

FinAlgoritmo

//10. Realizar una función que calcule la suma de los dígitos de un número.

// Ejemplo: 25 = 2 + 5 = 7

// Nota: Para obtener el último número de un digito de 2 cifras o más debemos pensar en el

// resto de una división entre 10. Recordar el uso de la función Mod y Trunc.

Funcion resultado <- SumaDigitos (n)

Definir resultado, i, z Como Entero

z=0

Mientras n > 0 Hacer

z = z + n MOD 10

n<-trunc(n/10)

FinMientras

resultado=z

Fin Funcion

Algoritmo ejercicio10

Definir num, s Como Entero

Escribir "Ingrese un numero entero "

Leer num

s=SumaDigitos(num)

Escribir "La suma de los digitos de ", num " es ", s

FinAlgoritmo

//11. Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número

// tiene todos sus dígitos impares (ejemplo: 333, 55, etc.). Para esto vamos a tener que separar el

// numero en partes (si es un numero de más de un digito) y ver si cada número es par o impar.

// Nota: recordar el uso de la función Mod y Trunc(). No podemos pasar el numero a cadena para

// realizar el ejercicio.

Funcion digimp <- DigitosImpar (num)

Definir digimp Como Logico

Definir i, sum, sp, simp Como Entero

i=0

sum=0

sp=0

simp=0

digimp=Falso

Mientras num>0 Hacer

sum=sum+num MOD 2

i=i+1

Si num MOD 2 <> 0 Entonces

simp=simp+1

SiNo

sp=sp+1

FinSi

num<-trunc(num/10)

FinMientras

Escribir "el numero tiene ", sp " pares y ", simp " impares. Total digitos ", i

Si i=simp Entonces

digimp<-Verdadero

SiNo

digimp<-Falso

FinSi

Fin Funcion

Algoritmo ejercicio11

Definir num Como Entero

Definir re Como Logico

Escribir "Ingrese un numero "

Leer num

re=DigitosImpar(num)

Escribir re

FinAlgoritmo

//Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número

//tiene todos sus dígitos impares (ejemplo: 333, 55, etc.). Para esto vamos a tener que separar el

// numero en partes (si es un numero de más de un digito) y ver si cada número es par o impar.

// Nota: recordar el uso de la función Mod y Trunc(). No podemos pasar el numero a cadena para

// realizar el ejercicio

Algoritmo ejercicio11Lucas

Definir num Como Entero

Definir var Como Logico

Escribir "ingrese un num"

Leer num

var<-impares(num)

Si !var

Escribir "El número tiene todos sus dígitos impares"

SiNo

Escribir "El número tiene algún dígito par"

FinSi

FinAlgoritmo

Funcion ret<-impares(x)

Definir ret Como Logico

Definir i,j Como Entero

i<-0

j<-0

Hacer

i<-(x MOD 2)

x<-trunc(x/10)

si i=0

ret<-Verdadero

j<-j+1

SiNo

ret<-Falso

FinSi

Mientras Que ret=Falso y x<>0

Si j>0

ret<-Verdadero

FinSi

FinFuncion

//12. Realizar una función que permita obtener el término n de la sucesión de Fibonacci. La sucesión

// de Fibonacci es la sucesión de los siguientes números:

// 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

// Donde cada uno de los números se calcula sumando los dos anteriores a él. Por ejemplo:

// La sucesión del número 2 se calcula sumando (1+1)

// Análogamente, la sucesión del número 3 es (1+2), Y la del 5 es (2+3) Y así sucesivamente?

// La sucesión de Fibonacci se puede formalizar de acuerdo a la siguiente fórmula:

// Fibonacci (n) = Fibonacci (n-1) + Fibonacci (n-2) para todo n > 1

// Fibonacci (n) = 1 para todo n <= 1

// Por lo tanto, si queremos calcular el término "n" debemos escribir una función que reciba

// como argumento el valor de "n" y que calcule la serie hasta llegar a ese valor.

Funcion variable\_de\_retorno <- Nombre ( Argumentos )

Fin Funcion

Algoritmo ejercicio12

FinAlgoritmo

Algoritmo ej12Lucas

Definir num Como Entero

Leer num

fibo(num)

FinAlgoritmo

SubProceso fibo(x)

Definir ret,a,b,c,i Como Entero

a<-0

b<-1

para i<-1 Hasta x

Escribir Sin Saltar b " "

c<-a+b

a<-b

b<-c

FinPara

Escribir " "

FinSubProceso

//13. Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número es

// capicúa o no (Por ejemplo: 12321). Nota: recordar el uso del MOD y el Trunc. No podemos

// transformar el numero a cadena para realizar el ejercicio.

//pista para el siguiente ejercicio, yo lo hice con un hacer mientras que y tuve que buscar por todos l

//ados la siguiente fórmula y acá les dejo mi interpretación de la misma, lo que hace es ir dando vuelta

//los dígitos poniéndolos luego de la coma de un 0 (digito\*(10^(-contador)))

Algoritmo ejercicio13

Definir num Como Entero

Definir resp Como Logico

Escribir " Ingrese un numero entero "

Leer num

Escribir "¿El número ", num, " es capicúa? ", CapicuaOno(num)

FinAlgoritmo

Funcion resultado <- CapicuaOno (num)

Definir resultado Como Logico

Definir i, numinv, aux Como Entero

aux=num

i=0

numinv=0

Mientras aux<>0 Hacer

i=aux MOD 10

numinv=(numinv\*10)+i

aux<-trunc(aux/10)

FinMientras

Si num==numinv Entonces

resultado<-Verdadero

SiNo

resultado<-Falso

FinSi

FinFuncion