Guía de Trabajos Prácticos de Pascal

# Parte 2

Cada ejercicio debe ser copiado/escrito en la versión de Pascal que posean y luego compilarlo, probarlo y “decorarlo”

1) Calcular el IMC (índice de masa corporal) para esto se debe tener en cuenta los siguientes datos: Altura, Edad y Peso.

El cálculo del IMC es: IMC = Peso / (altura ^2)

Con el Peso en Kg y la altura en metros.

La altura puede ser de 10 cm a 240 cm, la edad puede ser de 4 a 140 años y el Peso puede ser de 10 kg a 300 kg.

Según el valor que da el calculo podemos tener:

* Para los mayores o iguales a 20 años, el IMC menor a 19 es bajo peso, entre 19 y 24 es normal, entre 25 y 29 es sobrepeso, y más de 30 es obeso.
* Para menores de 20 años el IMC es: menos de 14 es bajo peso, entre entre 14 y 19 es normal, entre 20 y 24 es sobrepeso y más de 25 es obeso.

program ej01guia2;

var

edad, peso, altura: integer;

alturam: real; //Altura en metros

imc : integer;

resultado: string;

begin

writeln('Ingrese la edad que este entre 4 y 140 años');

readln(edad);

writeln('Ingrese el peso que este entre 10 y 300 kg');

readln(peso);

writeln('Ingrese el altura que este entre 10 y 240 cm');

readln(altura);

// Paso de cm a metros

alturam := altura / 100;

imc := Round(peso / (alturam \* alturam));

if edad >= 20 then

begin

case imc of

1..18 : resultado := 'Bajo Peso';

19..24: resultado := 'Normal';

25..29: resultado := 'Sobrepeso';

else

resultado := 'Obeso';

end;

end

else

begin

case imc of

1..13 : resultado := 'Bajo Peso';

14..19: resultado := 'Normal';

20..24: resultado := 'Sobrepeso';

else

resultado := 'Obeso';

end;

end;

writeln('Su IMC es: ', IMC, ' usted tiene: ', resultado);

end.

Siguiendo la forma propuesta de resolver el ejercicio, codificar una subacción que valide que los datos a la entrada sean los correctos (que estén dentro de los rangos propuestos), si el usuario ingresa un valor no válido, volver a pedirlo.

2) Siguiendo el ejercicio anterior, realizar una subacción que tome los valores máximos de los rangos para determinar el estado de una persona y cargue el valor correcto en la variable resultado.

3) Escribir un algoritmo donde se ingresan 3 números distintos y se deben ordenar de menor a mayor, e imprimir por pantalla.

program ej03guia2;

var

x1, x2, x3 : integer;

procedure intercambio(var y,z: integer);

var

aux: integer;

begin

aux := y;

y := z;

z := aux;

end;

function min(a, b, c: integer): integer;

begin

if (a < b) and (a < c) then

min := a

else

begin

if b < c then

min := b

else

min := c;

end;

end;

begin

write('ingrese el primer valor: ');

readln(x1);

write('ingrese el segundo valor: ');

readln(x2);

write('ingrese el tercer valor: ');

readln(x3);

if min(x1, x2, x3) = x2 then

intercambio(x1, x2)

else

begin

if min(x1, x2, x3) = x3 then

intercambio(x2, x3);

end;

if min(high(integer), x2, x3) = x3 then

intercambio(x3, x2);

writeln('el orden es el siguiente');

writeln(x1, ' , ', x2, ' , ', x3);

end.

4) Desarrolle el algoritmo de validación de número y clave de usuario, teniendo en cuenta las siguientes reglas:

* Todos los usuarios tienen un número de usuario fijo de 5 números, y una contraseña numérica de longitud variable.
* Para validar la contraseña se toma el 1er dígito del número de usuario y, si es par se multiplica cada uno de los dígitos de la contraseña por 2, si es impar, se multiplica cada uno de los dígitos por 3.
* Los resultados de las multiplicaciones se deben sumar y, si el total es mayor a los primeros 2 dígitos menos significativos del usuario, la contraseña es válida.

Un ejemplo de esto es:

usuario: 43021

contraseña: 5362

entonces: 4 es el primer digito del usuario, es par así que tenemos que multiplicar todos los número de la contraseña por 2 y sumarlos.

5 x 2 + 3 x 2 + 6 x 2 + 2 x 2 = 32

Como 32 es mayor a 21 (los dos últimos dígitos del usuario) la contraseña es válida.

Se debe hacer las validaciones necesarias y mostrar por pantalla : “contraseña válida” o “contraseña inválida”.

program ej04guia2;

var

user: longint;

pass: longint;

function sumaDigitos(num: longint): integer;

var

resultado : integer;

begin

resultado := 0;

while (num > 10) do

begin

resultado := resultado + (num mod 10);

num := num div 10;

end;

resultado := resultado + num;

sumaDigitos := resultado;

end;

function valido(user, pass: longint): boolean;

var

mayorValor, multiplicador: integer;

begin

if ((user > 100000) or (user < 9999)) then

exit(false);

valido := false;

mayorValor:= user div 10000;

if mayorValor mod 2 = 0 then

multiplicador := 2

else

multiplicador := 3;

if (sumaDigitos(pass) \* multiplicador) > (user mod 100) then

valido:= true;

end;

begin

write('Escriba su usuario: ');

readln(user);

write('Escriba su password: ');

readln(pass);

if valido(user, pass) then

writeln('Usuario y contraseña válido')

else

writeln('Usuario y contraseña inválido');

end.

5) idem al ejercicio anterior pero ahora la multiplicación de cada dígito de la contraseña debe ser por la posición en la que se encuentra en el número. El primer dígito se multiplica por 1, el segundo por 2, el tercero por 3 y así.

Ej: contraseña igual a 5362, va a ser 5 x 1 + 3 x 2 + 6 x 3 + 2 x 4

La contraseña “válida”, al igual que el ejercicio 4, es la que el resultado es mayor a los 2 números menos significativos del usuario.