

Introducción a la Programación. Examen práctico. (21/enero/2020)

Apellidos:			Nombre:		
Dni :	Grupo :	Subgrupo :			

Instrucciones:

- Entregar esta hoja como prueba de asistencia.
- La duración del examen es de dos horas. No está permitido salir del laboratorio durante el examen.
- **Desconecta el cable de red** del PC. Antes de empezar el examen **descarga en el PC** toda la información que lleves en el PenDrive. Una vez comenzado el examen ya no podrás usar el PenDrive.
- Debes entregar un único archivo con el programa que se solicita. El archivo deberá llamarse "**Apellidos_Nombre.pas**", donde Apellidos_Nombre son los apellidos y nombre del alumno.
- Dentro del archivo se debe introducir un **comentario** al principio con la siguiente información: apellidos y nombre del alumno, DNI, Grupo y subgrupo de prácticas, y nombre del profesor de prácticas.
- 20 minutos antes del fin del examen se abrirá una **Tarea en Aula Virtual** para entregar el examen. Cuando vayas a entregar **avisa al Profesor Vigilante**, conecta la red, entra a Aula Virtual, sube el .pas a la Tarea, cierra el PC y sal del Laboratorio.

Ejercicio

Un establecimiento vende **N** productos distintos, y en él trabajan **M** dependientes. Se dispone de la información sobre las existencias de cada producto, y también de las ventas de cada producto por cada dependiente. Por ejemplo si hubiese 6 productos y 5 dependientes la información podría ser la del ejemplo de las siguientes figuras:

Existencias: Existencias de cada producto

Producto1	Producto2	Producto3	Producto4	Producto5	Producto6
40	60	35	80	50	25

Ventas: Ventas de cada producto por cada dependiente

Dependiente	Producto1	Producto2	Producto3	Producto4	Producto5	Producto6
1	12	4	0	0	4	5
2	5	12	0	0	14	1
3	6	4	35	0	24	2
4	7	3	0	36	1	7
5	2	0	0	0	6	2

Construye un programa Pascal que contenga lo que se solicita en los apartados 1, 2 y 3, incluyendo un programa principal con unos datos de ejemplo que compruebe el correcto funcionamiento de la función y la acción que se solicita.

1. Crea las estructuras de datos adecuadas para representar esta información, suponiendo **N** y **M** dos constantes, que se denominarán **Tipo_Existencias** y **Tipo_Ventas**. (1 punto)
2. Crea una función que a partir de la siguiente información (parámetros de entrada): **ventas**, **existencias iniciales**, y **dependiente** (un entero X, tal que $1 \leq X \leq M$) devuelve **True** si dicho vendedor ha liquidado **él solo** todas las existencias de algún producto y **False** en caso contrario. Se supone que la información siempre es consistente, es decir, de un artículo no se pueden vender más unidades que las disponibles inicialmente. Para el ejemplo, el dependiente 3 es True. (4 puntos)
3. Supón que se dispone de la información sobre el precio de cada artículo (5 puntos). Por ejemplo:

Producto1	Producto2	Producto3	Producto4	Producto5	Producto6
3.75€	1.15€	2.00€	7.13€	4.00€	0.75€

- a) Crea un tipo de datos adecuado para representar esta información, denominado **Tipo_Precio**.
- b) Define una **acción**, que a partir de dos parámetros de entrada: **ventas** y el **precio** de cada artículo, devuelva **en un parámetro** el identificador (un entero X, tal que $1 \leq X \leq M$) del dependiente que mayor cantidad de dinero haya conseguido. Si hay varios con igual cantidad, se devuelve el que tenga mayor identificador. Para el ejemplo, el mayor es el dependiente 4, que facturó 265.93€.

Nota: Recuerda que una tabla 2D se puede inicializar en la declaración de variables de la siguiente forma. Puedes utilizar cualquier otra forma que consideres. Para la tabla de ventas ejemplo, la inicialización sería:

```
Var
  t : TipoTabla = ( ( 12, 4, 0, 0, 4, 5 ),
                    ( 5, 12, 0, 0, 14, 1 ),
                    ( 6, 4, 35, 0, 24, 2 ),
                    ( 7, 3, 0, 36, 1, 7 ),
                    ( 2, 0, 0, 0, 6, 2 ) );
```