



## Guion de prácticas 2

### *Paso de Parámetros y Uso de estructuras*



## Metodología de la Programación

Grado en Ingeniería Informática

Prof. David A. Pelta



## Introducción al guion

En este guion se pondrán en práctica los conceptos asociados al uso de structs y el paso de parámetros a funciones. La programación se realizará utilizando herramientas provistas en una instalación Linux estándar.

## Ejercicio 1: Lista de la Compra

En este ejercicio se pretende representar una compra realizada en una frutería. Para ello, se define una estructura *Producto* que permite representar UN producto que se haya comprado. De cada producto se almacena su nombre, peso (en gramos) y precio por Kg.

Posteriormente, podemos representar la información asociada a la compra de varios productos utilizando un struct *Compra* que contiene un array de elementos de tipo *Producto*. Ambas estructuras se muestran a continuación.

---

```
struct Producto{
    string nombre;
    int peso; // en gramos
    float precio_kg;
};

struct Compra{
    static const int MAX = 10;
    Producto lista[MAX];
    int util;
};
```

---

Estas definiciones están incluidas en el fichero *fruteria.cpp* que tiene disponible en PRADO. Además, el código incluye la definición de una función auxiliar para mostrar el contenido de una variable de tipo *Producto*, y un conjunto de instrucciones en la función *main* para probar las funciones a implementar.

```

Bitwise xterm - david@modo.ugr.es:22 - david@modo: ~/MP
david@modo:~/MP$ g++ fruteria.cpp -std=c++0x -o compra
david@modo:~/MP$ ./compra < datos.txt

***** Prueba de funcion listarCompra *****
cereza      345      2.550000
naranja     1380     1.100000
kiwi        876      1.800000
pera       1150     2.000000
platano     890      1.190000
melon      3500     1.500000
uva         530      2.100000
mango       456      2.500000
manzana     750      1.690000
limon       275      1.190000

***** Prueba de funcion obtenerImporteYPeso *****
El importe de la compra es: 16.4314, su compra pesa:10 Kg.

***** Prueba de la funcion mostrarTicketCompra *****
cereza      345      2.550000
naranja     1380     1.100000
kiwi        876      1.800000
pera       1150     2.000000
platano     890      1.190000
melon      3500     1.500000
uva         530      2.100000
mango       456      2.500000
manzana     750      1.690000
limon       275      1.190000
Subtotal:   16.4314
IVA (21%):   3.45059
Total de la compra: 19.882

***** Prueba de la funcion incrementarPrecio *****
***** y listarCompra de nuevo *****
cereza      345      2.805000
naranja     1380     1.210000
kiwi        876      1.980000
pera       1150     2.200000
platano     890      1.309000
melon      3500     1.650000
uva         530      2.310000
mango       456      2.750000
manzana     750      1.859000
limon       275      1.309000
david@modo:~/MP$

```

Figura 1: Compilación, ejecución y salida a mostrar por el programa. El parámetro `-std=c++0x` que aparece en la orden de compilación permite la utilización de la función `to_string()` que solo está disponible a partir de C++11. Dependiendo de la versión del compilador, puede que necesite usar `-std=c++11` o `-std=gnu++11`

## Tareas a Realizar

Cree un directorio `fruteria` que contenga a su vez los directorios `src`, `bin`, `include`, `obj`. Complete la implementación de las funciones indicadas en el fichero `fruteria.cpp` (donde se incluye una breve descripción de su funcionalidad). Posteriormente, construya un módulo `ListaCompra`, un fichero de pruebas `compra.cpp` (con el código de la función `main` y el `makefile` correspondiente).

Si el directorio actual es `fruteria`, la ejecución de los comandos

```
make
```

```
./bin/compra <datos.txt
```

debe producir una salida similar a la mostrada en la Figura 1. El fichero `datos.txt` (disponible en PRADO) contiene valores que se pueden asignar al vector `compra` utilizando la redirección de entrada para la lectura (esto es lo que hace el segundo comando).

## Ejercicios adicionales

Se sugiere la implementación de funciones para resolver las siguientes tareas.

- obtener un array de *Productos*, con aquellos elementos de una *Compra* cuyo peso esté comprendido entre  $[min, max]$ .
- suponga que el total de una compra  $C$  es  $T€$ , pero al intentar pagar se da cuenta que dispone de  $K €$  (con  $K < T$ ). Implemente una función que dada una compra  $C$  y una cantidad de euros  $K$ , devuelva una nueva compra cuyo importe no sea superior a  $K$ . Decida que productos eliminaría.
- Suponga que al final del día, la tienda ha recogido la información de todas las compras que se han realizado en un array *Compras* *listado*  $[MAX]$ . Implemente una función que calcule y muestre, para cada producto, los Kg. vendidos y la cantidad de ventas. Para la lectura de datos para el array, puede utilizar el fichero *listadoCompras.txt* disponible en PRADO.

**Nota:** las instrucciones aquí detalladas deben complementarse con las indicadas en la clase de prácticas.