

# Programación I

## Ejercicios –

### Paso de argumentos a función por referencia

Iván Cantador

Escuela Politécnica Superior

Universidad Autónoma de Madrid

- Ejercicio 1: filtro de valores
- Ejercicio 2: cesta de la compra

# Ejercicio 1: filtro de valores

2

Implementar en C una función “filtrarValores” que recibe una lista de números enteros y su longitud, una clave entera, y una lista de enteros vacía, y que copia a esta segunda lista aquellos números de la primera lista cuyos valores están por debajo de la clave. La función ha de devolver la longitud de la segunda lista, es decir, la cantidad de números copiados.

Implementar un main que pida al usuario una lista de números y una clave y muestre por pantalla la lista obtenida al llamar a filtrarValores.

```
#include <stdio.h>

#define MAX_DIM 256
#define MAX_CAD 32

int filtrarValores(int *listaOrigen, int longitud, int clave, int *listaDestino) {
    int copiados, i;

    if ( listaOrigen == NULL || longitud < 0 || listaDestino == NULL ) {
        return -1;
    }

    copiados = 0;
    for ( i=0; i<longitud; i++ ) {
        if ( listaOrigen[i] >= clave ) {
            listaDestino[copiados] = listaOrigen[i];
            copiados++;
        }
    }

    return copiados;
}
```

```

int main() {
    int lista1[MAX_DIM], lista2[MAX_DIM], longitud1, longitud2, k, i;

    printf("Introduzca la longitud de la lista: ");
    scanf("%d", &longitud1);

    for ( i=0; i<longitud1; i++ ) {
        printf("Introduzca un número de la lista: ");
        scanf("%d", &lista1[i]);
    }

    printf("Introduzca la clave de filtrado: ");
    scanf("%d", &k);

    longitud2 = filtrarValores(lista1, longitud1, k, lista2);

    if ( longitud2 >= 0 ) {
        printf("La lista filtrada es:");
        for ( i=0; i<longitud2; i++ ) {
            printf(" %d", lista2[i]);
        }
        printf("\n");
    }
    else {
        printf("Error: algún argumento es incorrecto.\n");
    }

    return 1;
}

```

# Ejercicio 2: cesta de la compra

5

En el contexto de una aplicación de gestión de la cesta de la compra, definir en C las siguientes estructuras de datos:

- Producto: con un nombre y un tipo
- ElementoCesta: con un producto y una cantidad del producto
- Cesta: con una lista de elementos de cesta y el número de elementos

Implementar una función “cestaAniadirElemento” que recibe una cesta, un producto y una cantidad de ese producto, y añade tal producto a la cesta.

Implementar un main que pida al usuario que indique la cantidad de dos productos:

- Leche (de tipo bebidas)
- Pan (de tipo cereales)

Con los datos leídos, el main insertará los elementos en una cesta y posteriormente los escribirá por pantalla.

```
#include <stdio.h>
#include <string.h>

#define MAX_CAD 256
#define MAX_CESTA 100

typedef struct {
    char nombre[MAX_CAD];
    char tipo[MAX_CAD];
} Producto;

typedef struct {
    Producto producto;
    int cantidad;
} ElementoCesta;

typedef struct {
    ElementoCesta elementos[MAX_CESTA];
    int numElementos;
} Cesta;
```

```
int cestaAniadirElemento(Cesta *c, Producto *p, int n) {
    int posicion;

    if ( c == NULL || p == NULL || n <= 0 ) {
        return -1;
    }

    if ( c->numElementos < 0 || c->numElementos >= MAX_CESTA ) {
        return -1;
    }

    posicion = c->numElementos;

    c->elementos[posicion].cantidad = n;
    strcpy(c->elementos[posicion].producto.nombre, p->nombre);
    strcpy(c->elementos[posicion].producto.tipo, p->tipo);

    c->numElementos++;

    return 0;
}
```



```

int main() {
    Cesta cesta;
    Producto leche = {"leche", "bebidas"}, pan = {"pan", "cereales"};
    int cantidadLeche, cantidadPan, ret, i;

    cesta.numElementos = 0;

    printf("Cuántos briks de leche quieres? ");
    scanf("%d", &cantidadLeche);

    ret = cestaAniadirElemento(&cesta, &leche, cantidadLeche);
    if ( ret == -1 ) {
        printf("Error: argumento incorrecto.\n");
    }

    printf("Cuántas barras de pan quieres?");
    scanf("%d", &cantidadPan);

    ret = cestaAniadirElemento(&cesta, &pan, cantidadPan);
    if ( ret == -1 ) {
        printf("Error: argumento incorrecto.\n");
    }

    printf("Hecho!\n");

    for ( i=0; i<cesta.numElementos; i++ ) {
        printf("%s: %d de %s.\n",
            cesta.elementos[i].producto.tipo,
            cesta.elementos[i].cantidad,
            cesta.elementos[i].producto.nombre);
    }

    return 1;
}

```